

## A VR Tool for the Visualization of CAD Models

**Eduardo T. L. Corseuil, Alberto B. Raposo, Romano J. M. da Silva,  
Marcio H. G. Pinto, Gustavo N. Wagner, Marcelo Gattass**

TeCGraf/PUC-Rio – Computer Graphics Group  
Dept. of Computer Science – Catholic University of Rio de Janeiro, Brazil

{thadeu, abraposo, romano, mhgpinto, gustavow, mgattass}@tecgraf.puc-rio.br

***Abstract:** This paper presents ENVIRON (ENvironment for VIRTUAL Objects Navigation), an application that was developed motivated by the necessity of using Virtual Reality in large industrial engineering models coming from CAD (Computer Aided Design) tools. This work analyzes the main problems related to the production of a VR model, derived from the CAD model, which must provide an immersive sensation to the user and allow him to interact with it in real-time. ENVIRON format conversion and visualization aspects are detailed, and performance considerations are discussed based on tests made with real oil platforms models.*

### 1. Introduction

One of the main objectives of the engineering departments of large industries, such as automobile, aerospace, and oil & gas, are the construction of integrated information systems to control their projects. They look for computational systems that, besides accessing the data bases with project information, offer resources for the 3D visualization of their models with enough realism to be used for design review, change management systems, and training, among other activities.

Virtual reality systems are being developed to accomplish these goals. In this kind of system, the efficiency is related to its immersion and interactivity capabilities, which requires a real time response of the system to the solicitations demanded by the users or by the model.

The difficulties encountered in this process are basically due to the fact that the engineering models are not constructed to be visualized in real time. In some cases, the models are visually simplified representations, serving only as schematic representations of the characteristics to be analyzed. In other cases, there are situations in which the models are too large and complex to be visualized in real time. This is the case of CAD models, the topic of this work, where many components, which are important to the engineering project, are unnecessary in the context of VR. Even the components necessary to the visualization, in their majority, need to be optimized for achieving a real time performance, since they may have an excessive detailing.

In this context, this paper introduces ENVIRON (ENvironment for VIRTUAL Objects Navigation), a tool developed to bridge the gap between CAD and VR models, facilitating the use of CAD models for VR applications. In this paper, we use the term "CAD model" referring to the engineering model, and "VR model" referring to the 3D model for real time visualization and interaction.

In the following section we point out the main difficulties in using CAD models for VR simulations, which motivated the development of ENVIRON. In section 3 we present the tool and some results using it. Then, we discuss the potential of ENVIRON in the light of some related tools. Finally, section 5 concludes the paper.

## 2. CAD to VR conversion

In the search for an efficient solution for the conversion of CAD models to VR models, we may detect several problems regarding the visualization of the models [Corseuil 2004]. Among these problems, which are common both in academic and commercial solutions, we may cite:

- **Low performance for complex models.** The achieved frame rates are unsatisfactory when very complex models, such as oil platforms and refineries, are loaded, especially in regions with large object concentrations. The available optimization algorithms and the accessible graphics hardware don't offer enough performance for VR models of that size.
- **Lack of realism.** CAD models generally don't have texture information associated with objects. To apply textures in individual objects in VR models is not feasible, due to the complexity of the models.
- **Inadequate treatment of complex surfaces.** Some surfaces may appear in CAD models in the form of NURBS (NonUniform Rational B-Splines). The polygonal meshes generated for this kind of surface in the VR model are generally inefficient for visualization, demanding a different treatment.
- **Difficult or inexistent link to the associated database.** CAD models include project information linked to each object. Some systems loose this link in the CAD to VR conversion. Other systems maintain the data links in VR models at the cost of performance.

Actually, the abovementioned problems are due to the focus divergence between CAD and VR models, which aim at different purposes. The generation of a CAD model demands that the objects forms be highly detailed, since it is aimed at the engineering execution process, and does not consider the effects of this in the real time 3D visualization. Therefore, it is necessary a "chain of adaptation steps" to convert CAD models to VR ones [Paillot et al. 2003].

To the best of our knowledge, there is not yet an integrated system where one can migrate from a CAD to a VR model (and possibly vice-versa), allowing easy interaction to make the necessary adjustments. The conversion processes currently available are direct and generally imprecise translations from CAD to VR, which are strongly dependent on the translated CAD format. In the absence of a standard for CAD to VR conversion, VR tools currently available use partial solutions, limited to specific CAD formats and quite dependent on user adjustments in the generated VR model.

This scenario motivated the development of ENVIRON, a tool to convert CAD models to VR models and to visualize and interact with them in virtual environments.

### 3. ENVIRON

ENVIRON is a system composed of exportation plugins and a 3D environment for real time visualization and interaction with the formats exported by the plugins. Currently there is a DGN and a 3DS exporter developed, respectively, as MicroStation and 3DSMax plugins. Figure 1 illustrates ENVIRON's visualization pipeline.

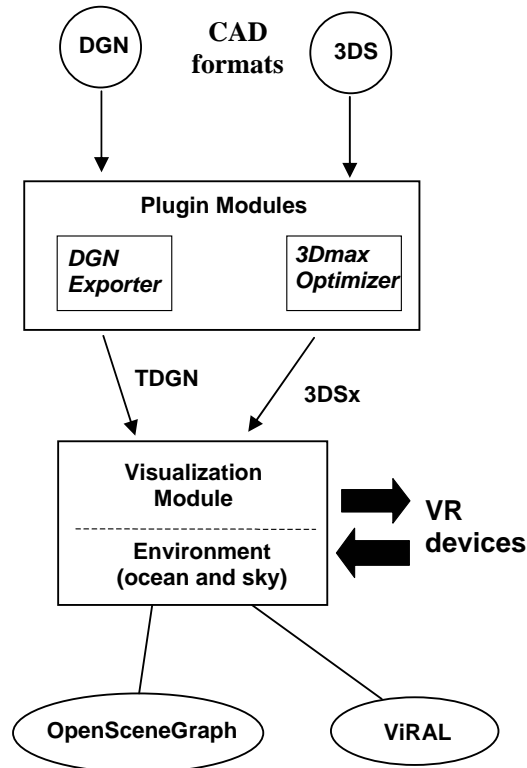


Figure 1: CAD to VR pipeline.

The goal of MicroStation plugin is not only to convert DGN files to a graphic format that enables the real time interaction and navigation in the CAD model, but also to recover and export the semantic information associated with the CAD's objects. In parallel, we also developed a 3DSMax plugin to enable the use of more photo-realistic models, not generated in a CAD tool.

In the following subsections we initially present the exportation plugins (subsections 3.1 and 3.2) and then detail some characteristics of the visualization tool (subsections 3.3 and 3.4). Finally, in subsection 3.5, we present some performance considerations.

#### 3.1. DGN exporter

The initial step of the CAD to RV conversion process is the understanding of the CAD format and its internal structure.

The DGN was chosen as the first format to be exported to ENVIRON because it is used as a standard by important Brazilian industries.

The Microstation offers some APIs to access the internal structure of DGN models and the data capture. We chose the MDL (MicroStation Development Language), written in C, as the plugin development tool [Bentley 1999]. Using the MDL, we could access the DGN internal structure and export the models to another format, adequate to visualization. We defined a format called TDGN [TeCGraf 2004] to make the link between the CAD tool and our visualization module, since DGN full internal structure is only accessible by MicroStation tools.

DGN objects are exported to TDGN format in three distinct types: parametric objects, NURBS and triangle meshes. The use of these types for visualization is discussed in section 3.3.

Some important information of DGN is included in TDGN. An example is the layer identification of each object. The layer information may be used to separate objects of distinct contexts in the CAD model. This information in TDGN associates the visualization to the project semantics and may be used to restrict the visualization to appropriate parts of the model.

Following the idea of bringing the design information to the VR visualization, it was also necessary to understand the relations between the information in the DGN file (graphic objects) and the information coming from the project database system. These links are also included in the TDGN format.

### **3.2. 3DS exporter**

Although ENVIRON is committed to the visualization of CAD models, it was also necessary to make it compatible with a tool capable of generating more realistic models. This necessity arises in non-technical and marketing presentations, where appearance is more important than accuracy. Therefore, we developed an exporter plugin for 3DSMax.

Natively, 3DSMax exports to several formats, including the old Autodesk 3DS format. This format was chosen because it is invariant and its source code is provided with 3DSMax software development kit. Moreover, it contains a lot of information useful for real-time rendering, such as vertex positions, normals, texture coordinates, material definition and animation. Besides that, all hierarchy information is stored in the file as a scene graph structure.

To complement these exported features, some enhancements to the native 3DS exporter plugin were made. The main add-ons were better normal calculation, since the original exporter had some normal flaws, and lightmap textures. The lightmap is rendered using 3DSMax functionalities and a special material is created. The native plugin didn't export this material.

The new exporter plugin was called 3DSx, since it was an extension to the 3DS plugin. With better normal calculation and the introduction of lightmaps there was a major improvement in visual quality.

### **3.3. Visualization module**

The visualization module of ENVIRON imports TDGN and 3DSx formats and executes the real time visualization of the models.

To enable a robust and efficient scene structuring and to enhance the application performance, we use the OpenSceneGraph (OSG) [Osfield and Burns 2003]. It implements important optimizations for the visualization of large models, such as frustum culling and techniques to reduce OpenGL state changes (lazy state change). Moreover, OSG is open source and has been showing extremely extensible.

Some implementation aspects of the visualization module are detailed in the following subsections.

#### **3.3.1. Handling objects for visualization**

Objects are loaded from TDGN as parametric ones, NURBS or triangle meshes, as previously stated.

Objects such as pipe and pipe knees that may appear in large number in process plants are stored in TDGN using the parametric form. This way we reduce the file size, the memory use during the visualization and enable the dynamic creation of meshes. The parametric objects support was integrated to OSG, being available for simple primitives, such as cylinders, spheres, lines, and curves, and their extrusion and revolutions.

For parametric objects, the information read is used to generate, when necessary, different visualization meshes, according to their distance to the observer. These objects enable the use of a LOD algorithm, implemented in the visualization module, that uses the parametric information of the form definition to generate, according to the object position, the triangle mesh needed for the visualization. In this visualization phase, there is a reduction in memory use because objects far away from the observer spend only the space necessary to store their few generation parameters, since no mesh is generated for these objects. This process offers the advantages of a conventional LOD algorithm, without the necessity of the pre-processing phase and the memory usage for storing different levels of detail, common in other techniques.

In some cases, due to the difficulty of obtaining the formation data of certain objects in the DGN file, they are stored in the TDGN file and loaded by the visualization module directly in the form of visualization meshes. Once these meshes are generated directly by internal functions of MicroStation, we have no control on their efficiency. In these cases, ENVIRON uses OSG resources directly, such as frustum culling and small features culling.

Different from the TDGN, the 3DSx exported file is strictly composed by meshes. There is no parametric data, although in 3DSMax much of the objects are described by parameters, such as a sphere or a cylinder. The 3DSx plugin only exports meshes because in 3DSMax one can create a modifier stack that changes substantially the characteristics of a parametric object, making its real time generation unfeasible. A main advantage of exporting mesh is that OpenGL and OSG work naturally with them, so no extra conversion needs to be made in real time, accelerating the rendering process.

Objects imported in NURBS form needed a special treatment, related in the following.

### 3.3.2. NURBS simplification

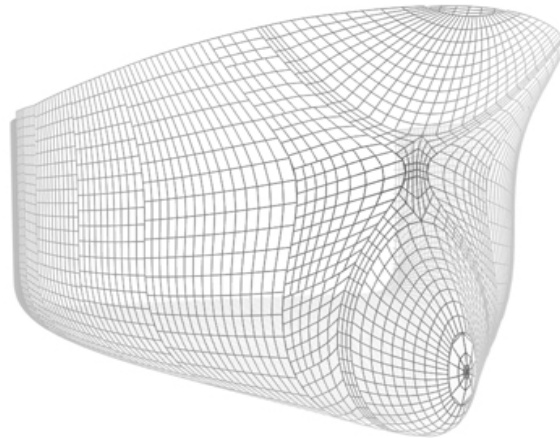
As stated in section 2, complex surfaces in the CAD model, especially those in the NURBS forms [Piegl and Tiller 1999], are a bottleneck in the generation of the VR model. CAD tools normally have limitations to generate models based on this kind of surface. Generally, surfaces generated by these tools are not optimized for visualization, since they have an excessive number of definition parameters.

Although CAD tools may offer functions for surface simplification and rebuilding, they generally cannot guarantee the desired result, since their process is based on the total reconstruction of the surface based on new parameters, and not on optimizations aimed at real time visualization.

To reduce the general problem of surface conversion, we created an algorithm to make the pre-processing of the imported surface parameters and to generate an adequate visualization mesh.

The goal is to obtain a mesh which geometric representation is analogous to the exported surface, but with a simplified discretization compared to the original model.

Nevertheless, there is a problem not solved simply with the mesh simplification. In every model tested, some objects were represented as a combination of many unconnected NURBS surfaces. When triangulated independently, the vertices of two adjacent surfaces would not match each other, creating an artifact which is very noticeable in real-time rendering, called T-Vertex (Figure 2).



**Figure 2: Shell of Petrobras P43 Platform, composed of 90 NURBS surfaces. Wireframe view of the simplified surface with T-Vertices.**

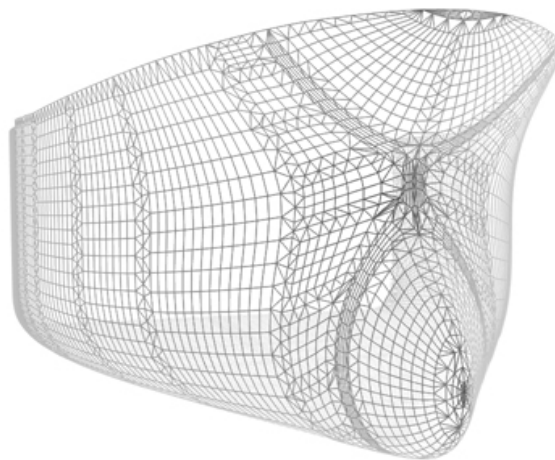
The T-Vertex problem was eliminated with a special NURBS tessellation algorithm, which uses information from adjacent surfaces to match the vertices on their borders.

The T-Vertices are an effect of the NURBS surfaces discretization. By working during the tessellation process rather than by trying to fix an already existing mesh, we

can take advantage of the information present on NURBS formulation, which is more accurate than the discrete mesh.

Our algorithm works in 3 steps:

1. Find the borders of each NURBS surface.
2. For each surface, walk along its border trying to define which vertices will be used on its tessellation:
  - If a vertex of another surface is found, use it on this surface's border.
  - If the current position along the border is too far from the last used vertex, create a new vertex on it.
3. Tessellate the surfaces in two distinct steps. The inner portion of the surface is formed by a uniform NxM grid (areas formed by squared polygons in Figure 3). The border of the surface is formed by a single strip of triangles, containing all the vertices selected on the border.



**Figure 3: P43 shell, after the use of the NURBS algorithm, with matching borders**

The resulting mesh has no T-Vertices, which would cause unrealistic visual artifacts in the VR model.

During this implementation, a library for NURBS visualization (NURBS++) based on OpenGL was adapted to the visualization module [Lavoie 1999]. This library helps in the reading of complex surfaces in the parametric space and in the generation of visualization meshes.

### **3.3.3. Other visualization features**

In order to offer something more than the real time visualization of the CAD model, additional features were implemented in ENVIRON. Some of them are described below.

- **Database access.** ENVIRON is capable of accessing the information stored in the project database associated with the CAD model.

- **Sky and ocean models.** In addition to a good representation of the CAD model, another important aspect is its insertion in a realistic environment. Regarding this aspect, ENVIRON integrates dynamic representations of the sky and the ocean.
- **Real time configurations.** In order to give the user a better visualization control, some parameters may be redefined in real time, during the interaction. For example, the near and far cutting planes and the FOV may be redefined at any time (Figure 4). The same applies to the configuration of the stereo visualization. The user may put the program in stereo state and define the eye separation distance and the parallax distance.

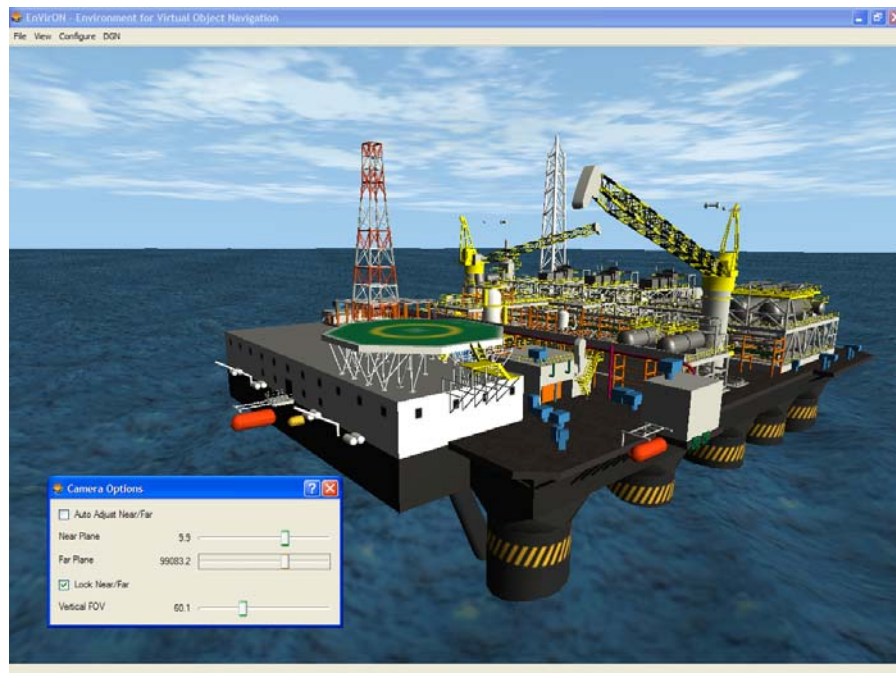


Figure 4: ENVIRON screenshot

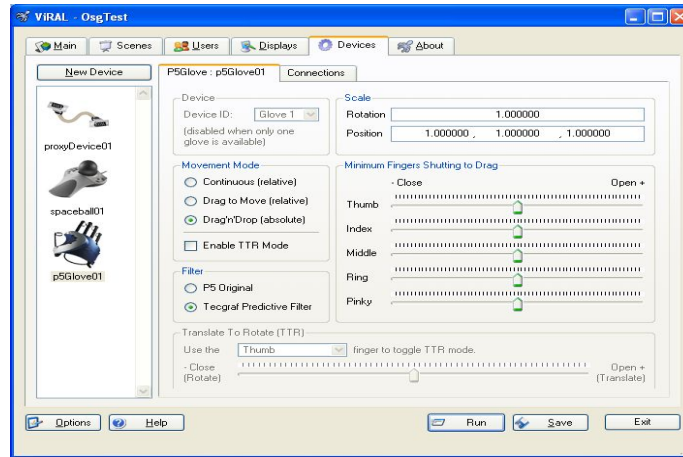
### 3.4. Immersive extension - ViRAL

To enable the use of ENVIRON in VR environments with different kinds of devices, we developed an independent tool, called ViRAL (Virtual Reality Abstraction Layer), which may also be used with other VR applications.

ViRAL is a tool designed to facilitate the development of VR applications. Applications developed with ViRAL are device independent, because ViRAL abstracts the context in which the application will be executed. The application doesn't have to know, for example, in how many windows, with how many users, or with which devices it is going to function. All these incognites are configurations defined by the application operator, making it possible to execute the same application in different VR systems without the necessity to modify the application.

ViRAL has a graphical user interface used to configure all devices, displays, projections and scenes. Figure 5 shows one of ViRAL screens, that of input devices configuration.





**Figure 5: ViRAL configuration window**

### 3.5. Experimental results

ENVIRON was developed based on CAD models of the oil & gas industry. More specifically, models of platforms and FPSOs (Floating, Production, Storage and Offloading), which are production and storage platforms adapted over oil ship shells. The example used in this section is the CAD model for the project of P43, a FPSO of Petrobras (Brazilian oil & gas governmental industry).

The data used here were furnished by Petrobras and were generated by PDS – Plant Design System [Intergraph 2003]. The data set consists of approximately 1800 vector graphic files representing design components and 1200 textual files with information about the components characterization coming from the database. Each textual file corresponds to an associated vector file. This data set has approximately 1GB and generates a 3D model of about 30 millions polygons.

Tests were realized to compare performance in different hardware and with different parts of the P43 model converted from DGN. It is important to reinforce that ENVIRON objective is to be easily accessible to engineers. Therefore, the most important results are those obtained in typical desktop PCs, and not in sophisticated graphics stations.

Table 1 illustrates some results varying the complexity of the model loaded in ENVIRON. The hardware configuration used was a Pentium IV, 2.5 GHz, 3Gb RAM, graphics board NVidia Quadro FX1000 (128 Mb). The fps rates were obtained using average values, with the observer located in a specific region of the platform, and not using stereo visualization.

**Table 1: Visualization performance of ENVIRON in a desktop PC.**

Number of polygons	Frames per second	Number of polygons	Frames per second
300,000	62	4,000,000	14
900,000	41	7,000,000	9
1,500,000	27	13,000,000	5

Note that the variation of the fps rate is not linear to the number of polygons loaded. This is explained by the fact that there are parts that demands more processing

than others with the same number of polygons. It depends on the kinds of the objects; the larger quantity of parametric objects we have the better performance we achieve.

The results shown in Table 1 cannot be considered a complete evaluation of the possibilities of ENVIRON. They just confirm the difficulties in obtaining a VR model with satisfactory performance from a large CAD model and, at the same time, show that the techniques used in ENVIRON are helpful in this process.

There is still room for several performance enhancements in ENVIRON. We are currently implementing occlusion tests in hardware and the hierarchical culling – i.e., the creation of an object hierarchy based on the objects localization to enable the culling of groups of objects at once. The hierarchical culling, in particular, is important because CAD objects are normally grouped by functionality – e.g., a kind of equipment. This functionality division, although essential for engineering purposes, is less important for real time visualization than a hierarchical grouping based on the objects localization.

Finally, it is important to reinforce that we avoid making performance comparisons with similar commercial tools. This is due to the fact that ENVIRON is a non-commercial tool, resultant from an academic research. Its goal is to be a flexible platform for building useful VR applications over real CAD models, as will be further explored in the following section.

#### **4. Flexibility issues**

ENVIRON is part of a research initiative to create an infrastructure for the “immediate” generation of VR environments from the CAD models, a task that currently requires a significant effort from the VR teams in industry.

In this context, ENVIRON has been designed to be an extensible tool, with flexibility to receive new functionalities and to incorporate plugins, according to the different necessities of the industry. This conception is the opposite of that of the available commercial solutions, normally offered as “black boxes”, with enhancements implemented by the developer on demand.

We haven’t ignored the available solutions for the VR visualization of CAD models, trying to continuously enhance ENVIRON with the advances in the field. In the following we present some available solutions (both academic and commercial) and then present some functions that are going to be incorporated into ENVIRON, illustrating its flexibility.

##### **4.1. Related systems**

Among the academic solutions for the real time visualization of large data volumes, GigaWalk [Baxter et al 2002] is one of the most important initiatives. This system uses scene graph reorganization techniques, such as hierarchical LOD and occlusion culling to achieve good visualization rates in CAD models with dozens of millions polygons. The limitation of GigaWalk is that, because of the necessary scene reorganization, it strongly depends on the model pre-processing, which may take several hours in large models.

Another system with similar objectives is REVIEW (Real-time Virtual Environment Walkthrough) [Shou et al 2001], which combines “conventional”

visualization techniques – such as frustum culling – with large databases I/O optimization techniques.

Commercial solutions have also been tested to confirm the bottlenecks in the CAD to VR conversion process. Among the available products, we may cite Division Reality [PTC 2003] and Walkinside [VRContext 2003]. Although they have different focuses, both aim at generating a VR experience from the CAD model.

The Walkinside has a simple interface, based on games, with few commands and intuitive use of the keyboard and interaction devices. However, it is restricted to the DGN CAD format, and the current version does not have resources for the use in highly immersive VR environments. The Division Reality, on the other hand, accepts more CAD formats and has advantages in the direct manipulation of objects. However, it is not accessible by desktop computers, being used only in high performance graphic stations. Both have problems with complex surfaces, such as NURBS.

#### 4.2. New targets

An important step toward expanding the use of VR in industry is to integrate other systems to the VR model converted from the CAD. The idea is to bring other classes of data to the VR model in order to build a more complex virtual environment.

An example, which is the long term goal of ENVIRON, is to integrate to the virtual environment not only the CAD model, but also the model coming from GIS (Geographic Information System) systems, representing the surface of a region, and seismic data sets. In the specific case of oil & gas industry, the integrated virtual environment would be an oil exploration area, including the CAD models of ships, platforms and undersea equipments, the GIS models that represent the oil exploration basin, and the representation of the subterranean reservoir. This environment would integrate several functions and analysis of oil exploration and production activities.

In the immediate future, there are more punctual steps already defined for ENVIRON enhancement:

- **Automatization of texture application in the VR model.** Since CAD models don't have texture information, it should be done manually in the VR model to enhance its realism.
- **Definition of a semantic model for objects coming from CAD to create a standard for visualization.** Due to limitations in the interpretation of DGN, objects are exported to the visualization module in three different forms, as detailed in section 3.3, what hinders the visualization process.
- **Development of a collaborative version.** Remote collaboration among groups of engineers geographically distributed is a current necessity in large industries. Our goal is to use ENVIRON in high performance nets, which would offer better conditions to the collaborative use of the tool.

#### 5. Conclusion

This paper presented ENVIRON, a tool that aims to “bridge the gap” between CAD and VR models and to be a platform for integrating other systems to the virtual environment.

Among the main problems of the CAD to VR conversion mentioned in section 2, ENVIRON has been particularly successful in the treatment of complex surfaces. Regarding the connection to the databases and the scene realism, ENVIRON also offers important resources. Although ENVIRON performance is good for a great part of available CAD models, this is perhaps the subject that still deserves more attention, especially because ENVIRON is designed to run in desktop computers and to visualize CAD models with increasing complexity.

The problems presented in the CAD to VR path have not been totally solved by any commercial or academic solution yet. This fact restricts, but doesn't impede the use of these systems in industry. The process is still executed in parts, some of them requiring manual interventions, which are specific for the working model. Nevertheless, there are already advantages in the use of VR models generated from CAD models in processes like construction simulation and equipment maintenance.

The problems presented will be minimized when standards for the CAD to VR conversion are established. Meanwhile, the effort presented in this work, as well as other efforts occurring in parallel are valid in the search for these solutions.

### **Acknowledgements**

Tecgraf/PUC-Rio is a group mainly funded by Petrobras, the Brazilian Oil & Gas Industry. We would like to thank all those involved with VR at Petrobras engineering area: Álvaro Maia, Heitor A. Filho, Ismael Santos and Enio Russo (CENPES), Paulo R. Oliveira (UN-Rio). Alberto Raposo has an individual grant from CNPq, n. 305015/02-8.

### **References**

- Bentley Systems Incorporated (1999): MDL Function Reference Manual.
- Baxter III, W. V., Sud, A., Govindaraju, N. K., Manocha, D. (2002): GigaWalk: Interactive WalkThrough of Complex Environments. Eurographics Workshop on Rendering.
- Corseuil, E. T. L., Raposo, A. B. et al. (2004): ENVIRON – Visualization of CAD Models In a Virtual Reality Environment. In Eurographics Symposium on Virtual Environments (EG-VE), pp. 79-82.
- Intergraph: Plant Design System – PDS (2003), <http://ppo.intergraph.com/pds>
- Lavoie, P. (1999): NURBS++: The Nurbs Package - User's Reference Manual - Version 3.0 <http://libnurbs.sourceforge.net> .
- Osfield, R., Burns, D. (2003): Open Scene Graph, <http://www.openscenegraph.org>
- Paillot, D., Merienne, F., Thivent, S. (2003): CAD/CAE Visualization in Virtual Environment for Automotive Industry. In Eurographics Symposium on Virtual Environments (EG-VE), pp. 315-316.
- Parametric Technology Corporation – PTC (2003): Division Reality, <http://www.ptc.com>
- Piegl, L., Tiller, W. (1999): The Nurbs Book, 2nd ed. Ottawa: Springer.
- Shou, L., Chionh, J., Huang, Z., Ruan, Y., Tan, K-L. (2001): Walking Through a Very Large Virtual Environment In Real-time. 27th Int. Conf. on Very Large Data Bases, pp. 401-410.
- Tecgraf (2004): TDGN Format Specification, <http://www.tecgraf.puc-rio.br/~mhgpinto>
- VRContext (2003): WalkInside, <http://www.walkinside.com>