ENVIRON - Visualization of CAD Models In a Virtual Reality Environment

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, Catholic University of Rio de Janeiro, Brazil.

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.

Categories and Subject Descriptors (according to ACM CCS): I.3.7: Virtual reality; J.6: Computer-aided design

1. Introduction

One of the main objectives of the engineering departments of large industries, such as automobile and oil & gas, are the construction of integrated information systems to control their projects. They look for computational systems that, besides accessing 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. 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, serving only as schematic representations of the characteristics to be analyzed. In other situations, the models are too large and complex to be visualized in real time. This is the case of CAD models, 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 large models, facilitating the use of CAD models for VR applications. 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.

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. 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 models, such as oil platforms and refineries, are loaded, especially in regions with large object concentrations.

- **Lack of realism.** CAD models generally don't have texture information associated to objects.

- **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.

Actually, the abovementioned problems are due to the diversity 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 execution process, and does not consider the effects of this in real time visualization. Therefore, it is necessary a "chain of adaptation steps" to convert CAD models to VR ones [PMT03].

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.

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.
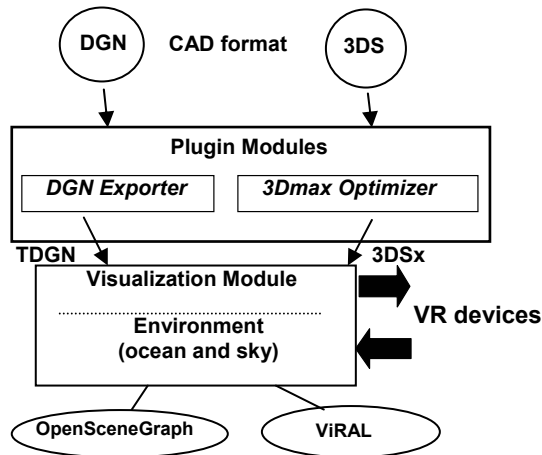

Figure 1: *CAD to VR pipeline.*

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

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. DGN was chosen as the first to be exported to ENVIRON because it is the standard in important Brazilian industries.

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 [Ben99]. Using MDL, we could access the DGN internal structure and export the models to another format, adequate to visualization. We defined a format called TDGN [Tec04] 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 (see section 3.3.)

Some important information of DGN is included in TDGN. An example is the layer identification of each object. This 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.

The links between the information in the DGN file (graphic objects) and the information coming from the project data base system 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.

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) [OB03]. 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. 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 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.

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. Different from the TDGN, the 3DSx exported file is strictly composed by meshes.

3.3.2. NURBS simplification

As stated in section 2, complex surfaces in the CAD model, especially those in the NURBS forms, 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.

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.

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:

    a. If a vertex of another surface is found, use it on this surface's border.

    b. If the current position along the border is too far from the last used vertex, create a new vertex.

3. Tessellate the surfaces in two distinct steps. The inner portion of the surface is formed by a uniform NxM grid. The border is formed by a single triangle strip containing all the vertices selected on the border.

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.

*   **Data base access.** ENVIRON is capable of accessing the information stored in the project data base associated to the CAD model. This capacity offers better parameters for the user to evaluate the model and to analyze possible alterations. For example, it is possible to identify the composition of a process plant subprocess based on the identification of their objects that are defined in the data base.

*   **Sky and ocean models.** In addition to a good representation of the CAD model, another important aspect is its insertion in a realistic environment. ENVIRON integrates dynamic representations of the sky and the ocean. The sky was constructed by the mapping of one or more textures in a semi-sphere, adding dynamic movements of clouds and stars. The ocean simulation was based on the wave technique of Gerstner [Tes00]. Terrain representations are planned in future versions.

*   **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 2). The same applies to the configuration of the stereo visualization. Other real time configurations available are related to the dynamic representations of the ocean and the sky.

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.

In essence, ViRAL has the same objectives of any other abstraction device tool, such as VRJuggler [Bie00]. However, ViRAL has at least two major differences with other VR toolkits. The first one is that it can be run as the VR application. The developer creates her plugin and uses ViRAL to load and run it. The second important difference is that ViRAL can also be used embedded in a host application, as is the case of ENVIRON. The configuration panels of ViRAL can be shown from within a third party application. This way ViRAL is not the running
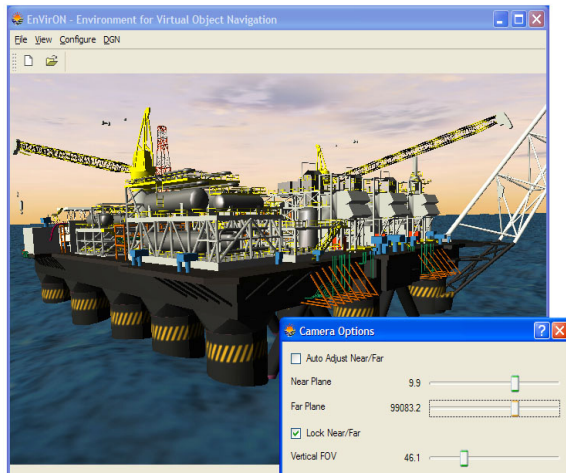
Figure 2: *ENVIRON screenshot.*

application, but all its VR abstraction features can be called from the host application.

3.5. Experimental results

ENVIRON was developed based on CAD models of the oil & gas industry. The example used in this section is the CAD model for the project of P43, a FPSO (Floating, Production, Storage and Offloading) of Petrobras (Brazilian oil & gas governmental industry).

Tests were realized to compare performance in different hardware and with different parts of the 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. 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 demand more processing than others with the same number of polygons. It depends on the kinds of the objects; the larger quantity of parametric objects, the better performance we achieve.

Table 1: *Performance of ENVIRON in a desktop PC.*

Polygons	Fps	Polygons	Fps
300,000	85	3,000,000	13
600,000	42	6,000,000	8
900,000	28	15,000,000	4
1,200,000	24		

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.

It is important to reinforce that we avoid making performance comparisons with similar commercial tools. This is due to the fact that there are few applications that have capability to run large models like that, with an order of millions of polygons and also 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.

4. Conclusion

ENVIRON is part of an initiative to create an infrastructure for the "immediate" generation of VR environments from CAD models, a task that currently requires a significant effort from the VR teams in industry. It has been designed to be an extensible tool, with flexibility to receive new functionalities and to incorporate plugins. This conception is the opposite of that of available commercial solutions, normally offered as "black boxes", with enhancements implemented by the developer on demand.

Among the main problems of the CAD to VR conversion mentioned in section 2, ENVIRON has been particularly well succeeded 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 manually interventions, which are specific for the working model.

References

[Ben99]   Bentley Systems Incorporated: *MDL Function Reference Manual*, 1999.

[Bie00]   Bierbaum, A.:*VR Juggler: A Virtual Platform for Virtual Reality Application Development*. Master Thesis, Iowa State University, 2000.

[OB03]   Osfield, R., Burns, D.: *Open Scene Graph* (2003), http://www.openscenegraph.org

[PMT03]   Paillot, D., Merienne, F., Thivent, S.: CAD/CAE visualization in virtual environment for automotive industry. *9th Eurographics Workshop on Virtual Environments* (2003), pp. 315-316.

[Tec04]   Tecgraf: *TDGN Format Specification* (2004), http://www.tecgraf.puc-rio.br/~mhgpinto

[Tes00]   Tessendorf, J.: Simulating Ocean Water. In *SIGGRAPH´00: Course Notes 25: Simulating Nature: From Theory to Practice*. 2000.