Distributed visualization of engineering models

VIZUALIZAÇÃO DISTRIBUÍDA DE MODELOS EM ENGENHARIA

Raúl Ademar Valdivia Pacheco
Marcelo de Andrade Dreux
Mechanical Engineering Department, Pontifical University of Rio de Janeiro

ABSTRACT

The growing increase in computational resources for numerical simulation allows researchers and engineers to generate huge amount of data. The understanding of those data through the use of computer graphics techniques is known as Scientific Visualization. Often, it is necessary to have more than one computer/user accessing the same data set, known as Distributed Visualization This work proposes Scientific Visualization of engineering problems through the use of a Distributed Architecture via Internet (using a browser with VRML Plug-in). The simulated data are obtained directly from a data base and the appropriate information is generated to their visualization. Pre-processing data (such as nodes, elements, lines, areas etc) and post-processing data (such as deformation, stress and strain, from a finite element analysis) are generated. A Distributed Architecture is considered which allows the numerical simulation to be done in a main computer (server) and the visualization to take place in another computer (client), by using a simple, but robust, interface, as is a Internet browser, disposing the access from wherever have Internet. Results obtained with the ANSYS software have proven to be satisfactory when compared with the results visualized through the mentioned software. The proposed architecture can be extended to cope with other solid mechanics software.

KEYWORDS

Scientific Visualization. Distributed Architecture. Finite Elements Analysis. VRML. ANSYS.

RESUMO

O crescente aumento da disponibilidade de recursos computacionais para a simulação numérica permite que cientistas e engenheiros produzam enormes quantidades de dados. A melhor compreensão destes dados mediante o uso de técnicas de computação gráfica é conhecido como visualização cientifica. Este trabalho propõe a visualização científica de problemas de engenharia usando uma arquitetura distribuída via WEB. Os dados simulados são lidos diretamente de um banco de dados e são gerados arquivos com as informações necessárias para sua visualização. Geram-se arquivos com dados de pré-processamento (como nós, elementos, linhas, áreas e elementos diferenciados por índices) e pós-processamento (como deformação, deslocamento e tensões, resultados mais importantes na análise utilizando o método de elementos finitos). Considerando uma arquitetura distribuída, a simulação numérica pode ser feita em um computador (servidor) e a visualização pode ser feita em um outro computador (cliente), utilizando uma interface simples, porém robusta para a visualização, como é o caso da WEB. A utilização do formato VRML facilita a distribuição e compartilhamento nesta visualização, tornando-se independente da plataforma do servidor que contém o software de simulação numérica e da plataforma do cliente. Usando como caso de estudo o software de Análise de Elementos Finitos ANSYS, os resultados obtidos mostraram-se satisfatórios e melhor manipuláveis ao se comparar com resultados visualizados por aquele software. O estudo de caso pode ser estendido para outros softwares de simulação da área de mecânica dos sólidos.

PALAVRAS CHAVE

Visualização Cientifica. Arquitetura WEB. Análise de Elementos Finitos. VRML. ANSYS.

INTRODUCTION

Finite Element Analysis (FEA) is a largely used technique to evaluate stress, strain, vibration and other properties of engineering structures, with the use of computers. The use of computer is necessary since there are many operations involved in a large engineering structure (Budgell, 1998). Due to its flexibility and wide applicability FEA is used in both academic and industrial environments (Huebner et al., 2001). There are many available FEA softwares and ANSYS is the most widely used.

Data storage increase, better data transmission, new interaction and visualization techniques and more computer power are bringing a new paradigm to the engineering design process. With those characteristics it is possible to envisage a work environment more reliable, portable, remote, configurable, dynamic and cooperative. Workers in different geographic locations and with mobile computers can share information through the Internet and are able to present real time visualizations of their engineering projects (Rosenblum et al., 1994).

Usually, large engineering projects are developed in different locations and there is a lack of integrated updated information. There are some possible computer architectures to integrate geographically dispersed data. A collaborative environment could be adopted in order to deal with the integration of engineering projects (Felicíssimo; Salles, 2002). It is also necessary to enhance 3D data transmission through the Internet. VRML (Virtual Reality Modeling Language), a standard language designed to represent 3D geometric models, could appropriately suit this need. Basically, VRML is a text file, which is able to define a virtual 3D world, with animation and interaction, controlled by a reasonably simple script language (Marudur, 1998).

The purpose of this work is to present the VRMLGer, which consists in a pre and post-processor of a FEA software. VRMLGer works in a distributed WEB architecture. As a case study, ANSYS is the used FEA software. There will be provided macros responsible for extracting, from the proprietary ANSYS files, the necessary data for the generation of VRML files.

2. A DISTRIBUTED ARCHITECTURE VIA WEB

It is shown in Fig. 1 a distributed environment, divided in three layers: user, mediator and server. Each layer is independent of the others and has a specific function.

Figure 1 - Distributed environment with three layers.

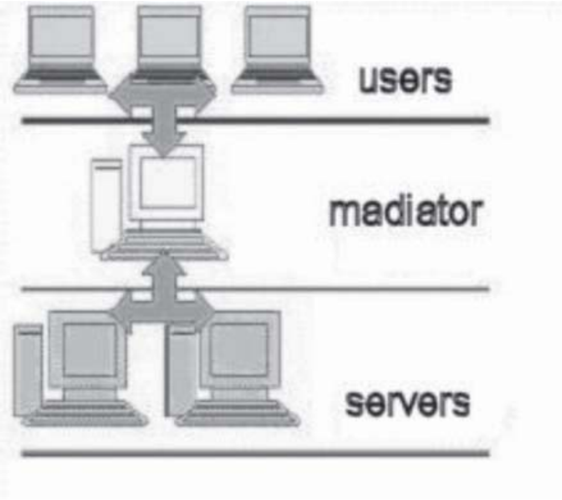
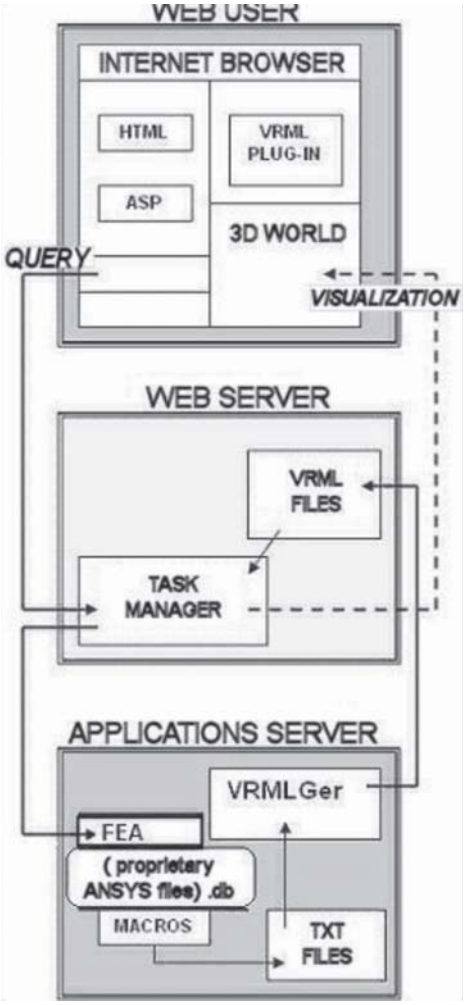


Figure 2 - Data flow in a three-layer distributed environment.



The first layer, called user layer, allows data access, visualization and query. In this layer the user (client) interacts with the tri-dimensional generated model. The second layer, called mediator layer, is in charge of receiving the requested tasks from the user layer and distributing them to the following layer. It is also responsible to keep track of the owner (client) of each request task. The third layer, the server layer, is where the requested tasks are performed.

The data flow between the layers must be transparent to the user. The user, located at the first layer, must have the sensation that he is only interacting with a WEB server, the second layer (mediator). The existence of an application server is transparent to him.

Figure 2 shows the data flow in more detail, from the request of visualization, through a WEB browser, to the generation of VRML files. As can be seen in Fig. 2, there is also an application server where the FEA software is located, usually storing proprietary formatted data. In this server, there are macros responsible for obtaining the text files, which are input to the VRMLGer.

VRML LANGUAGE

VRML is a popular standard to model and transmit 3D objects through the Internet. It allows the definition of the most common graphic models characteristics, such as virtual cameras, geometric transformations, light sources, animation, fog, textures etc. It can be thought as an 3D analogy to the HTML protocol, i.e. it is a simple multi-platform language to publish 3D WEB pages. In applications such as games, scientific visualization, engineering, architecture, education etc it is necessary to have, besides the HTML characteristics, the possibility of 3D visualization and manipulation, present in VRML. VRML integrates text, 2D, 3D and multi-media and can also contain Java or JavaScript (Goodman, 2001) to define behaviors to objects of the scene (Ranga, 2000). VRML is able to be connected to external programs and thus perform sophisticated computation, and hence nicely suits scientific visualization tasks.

VRML came with the necessity to provide a WEB 3D graphics format, similar to HTML (Graham, 2000), i.e. a language, platform independent, to describe 3D scenes. Initially the reference language was SGI Open Inventor ((http://www.sgi.com). Later, VRML 1.0 emerged as a language to describe static 3D scenes. In 1997, VRML 2.0, added to the original version the possibility of animation and interaction (Ames et al, 1997).

VRML is an open source consortium of researchers from companies and universities. Until 1999, the consortium was named VRML Consortium and after that became Web 3D Consortium (http://www.web3d.org/). Its main purpose is to elaborate and maintain new standard to the transmission of 3D data through the WEB. It also includes a better integration among VRML, Java 3D, MPEG-4 and other related technologies (Raposo et al, 2000).

A VRML scene visualization is achieved by means of a VRML browser, usually connected to a standard WEB browser. The VRML browser reads and interprets a file with a virtual world description (extension .wrl), draws it in a window and provides a graphical interface to the user, such that he can interact with scene objects.

VRML worlds are based on nodes, which are objects and real world entities abstractions, such as geometric models, lights, sounds etc. Like in Java 3D (http://java.sun.com/products/java-media/3D), a VRML world is a hierarchical graph of nodes (Hartman & Wernecke, 1996)

IMPLEMENTATION OF VRML FILE GENERATOR - VRMLGER

VRMLGer is written in C++ (Stevens & Walnum, 2000) but could be developed in any language that supports dynamic data structures, and uses object-oriented concepts.

Input data are text files with the following information:

ALLNODES - the geometry of each point. These points are called nodes in FEA.

ALLELEM - the nodes that are part of each element. Elements may be quads or triangles, or the union of two points. Usually there are three identifiers as attributes of each element: material, element type and element thickness.

KPLIST - Key-Points defined by the user during mesh generation.

LINES - line information, defined by the user during mesh generation. Each line is composed of a series of nodes.

AREAS - surfaces defined by the user during mesh generation. Each surface is composed of a series of elements. Each element belongs to a single surface.

ALLDOF - degrees of freedom (DOF) of each node. It stores the x, y and z displacements of each node after the FEA processing.

RES - Other FEA results such as temperature, stress, etc. of each node.

In order to extract the necessary data from the used FEA software (ANSYS) it has been used the APDL programming language (ANSYS, 2000). These data are in a proprietary format (extension .db) and can only be accessed by scripts written in APDL. Those scripts generate the text files, presented above, and are the only part of the system dependent on the FEA used software. Any FEA software that allows the generation of such files could easily be used in the proposed environment.

It is possible to configure many aspects of the VRMLGer, through the use of various text files, as follows:

Palette - contains 128 lines with red, green and blue values, between 0 and 1. 0 means absence of color and 1 is the maximum value. It sets up the palette to be used.

Transparency - a value between 0 and 1 to indicate the transparency of the object, where 1 means 100% transparency.

DiffuseColor - 1 or 0 to indicate if the diffuse color contribution should or should not be evaluated, respectively. The diffuse color contribution provides more realistic scene shading (Watt, 2000).

Angle - a numeric value between 0 and 2? responsible for eventual smoothness of face junctions. 0 means a sharp contour and greater angle values mean smooth contours.

Function - a numeric value, related to the type of a desired effect. There are twenty available effects and the main ones are presented in the next section.

RESULTS

The results presented below are related to a bifurcation, which mesh is composed of 2041 nodes and 1934 quadrilateral elements.

Figure 3 - VRMLGer function 3, with filled elements and contour lines.

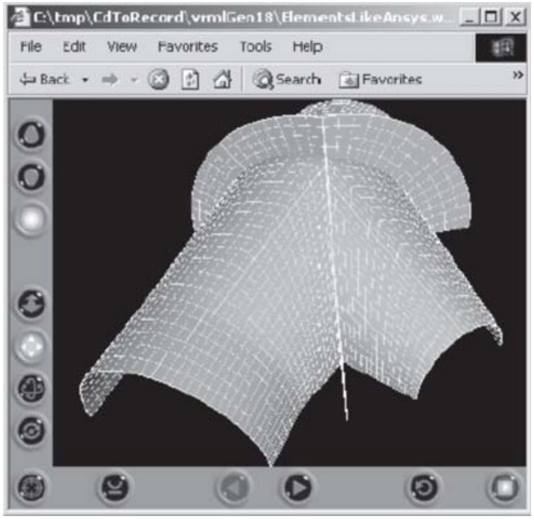


Figure 4 - VRMLGer function 6, where elements are classified by thickness.

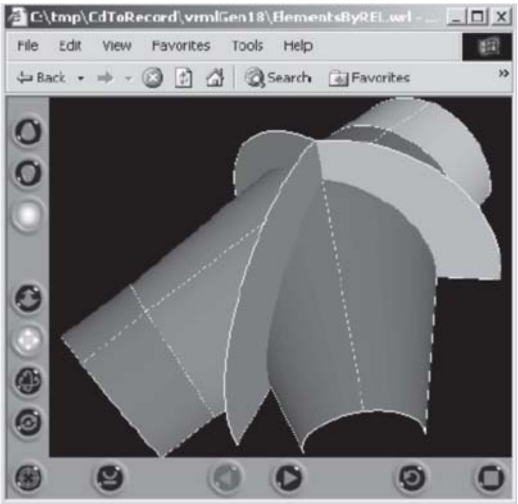
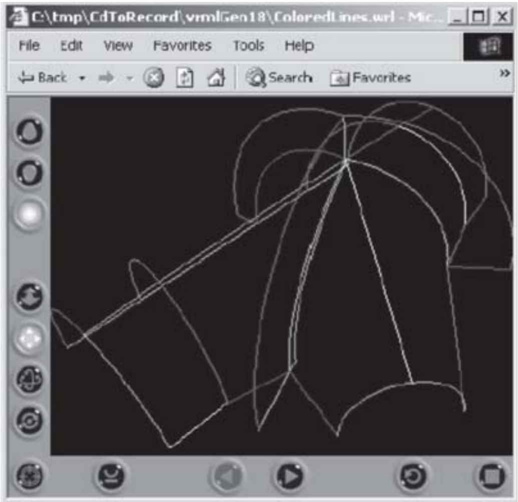


Figure 5 - VRMLGer function 7, with colored contour lines.



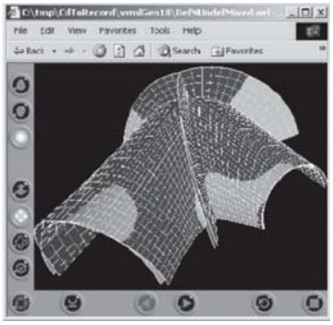
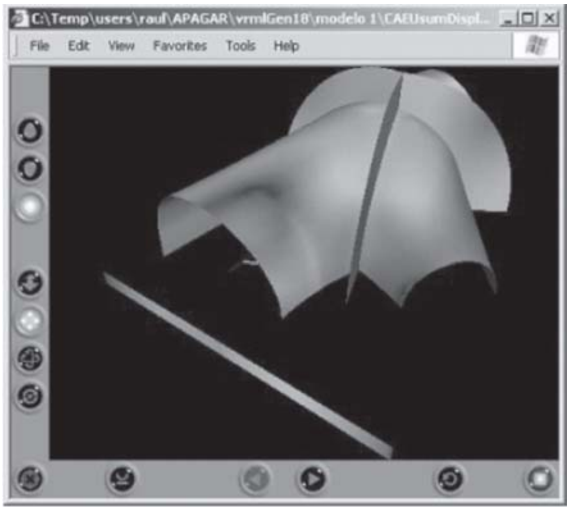


Figure 6 - VRMLGer function 13, where original and deformed meshes are overlapped.

Figure 7 - VRMLGer function 14.3, with colors proportional to displacements



The computational time required to generate the files through VRMLGer is negligible. The larger processing time, related to function 13, is less than one second, to the above-mentioned mesh. This time has been obtained in an 890MHz CPU with 192 MB RAM memory and Windows 2000.

VRML file sizes are insignificant if compared to the proprietary db files. The larger generated file, corresponding to function 13, is 482 Kbytes, while the db file occupies 18,560 Kbytes. It means 38 times larger than the VRML file. It has a huge impact if one needs to transmit the file through the Internet, in order to generate a remote visualization.

CONCLUSIONS

It has been presented a distributed architecture to manipulate and visualize large engineering projects. The idea to provide a VRML generator guarantees a platform independent approach and provides many visualization techniques not available in FEA software. Another important feature is that FEA results can be remotely visualized and the client does not need to have the FEA software installed. It needs only a WEB browser with a VRML plug-in.

The VRML generator requires some data from the FEA software and any FEA software could be coupled to the system, if functions to generate the necessary text files are developed. Our case study has used ANSYS, widely used in industries and universities. The proposed architecture is divided in three layers (user, mediator and server) and only the last layer is dependent on the FEA software.

At the moment, the implemented VRML functions are able to consider shading proportional to a certain property (stress, strain, displacement etc), to visualized transparent objects, to consider diffuse reflection, to select specific mesh elements based on certain properties etc. Zooming and rotation are naturally available through the use of a VRML plug-in.

ACKNOWLEDGEMENTS

The first author thanks CNPq and FAPERJ for the granted scholarship during the development of this work. This work was developed in ICAD/IGames/VisionLab of PUC-Rio to which the authors are grateful.

REFERENCES

AMES, A.L., NADEAU, D.R. & MORELAND, J.I. The VRML 2.0 sourcebook, 2ed. Canada: John Wiley & Sons, 1997. 688 p.

ANSYS, 2000, APDL Programmer's guide. Disponível em: <https://censs.chofu.jaxa.jp/jpn/manual/cevis/apl_guide/ANSYS/docu/japanese/ansyshelp/Hlp_P_APDLTOC.html>. Acesso em: 24 abril 2005.

BUDGELL, P.C. Finite Element Analysis and Optimization Introduction. Disponível em:< http:/ww3.sympatico.ca/peter_budgell/FEA_intro.html>. Acesso em: 29 jan. 2005.

FELICÍSSIMO, C.H. & SALLES, M.A.V. Um Ambiente Colaborativo para Desenvolvimento de Projetos de Engenharia de Grande Porte. Rio de Janeiro: EasyCAE Designer, 2002. 29p.

GOODMAN, D. *JavaScript Bible*. 4 ed. New York: John Wiley & Sons, 2001. 1200 p.

GRAHAM, I. Introduction do HTML. Disponível em:< http://www.utoronto.ca/webdocs/HTMLdocs/NewHTML/htmlindex.html>.Acesso em: 27 mar. 2007.

HARTMAN, J., WERNECKE, J. & SILICON GRAPHICS. The VRML 2.0 Handbook: building moving worlds on the web. [S.I.]: Addison-Wesley, 1996. 448 p.

HUEBNER K. H et al. *The Finite Element Method for Engineers.* 4 ed. [S.I.]: Wiley-Interscience. 2001. 744p.

MARUDUR, K.S. *Concurent-Interactive Design and Analysis using the Internet and VRML*. 108p. 1998. Thesis, (Master of science) - University of Oklahoma, Norman, 1998.

MAGALHÃES, L.P., RAPOSO, A.B. AND TAMIOSSO, F.S. VRML 2.0: an Introductory view by examples. Disponível em: <http://www.dca.fee.unicamp.br/~leopini/tut-vrml/vrml-tut.html>. Acesso em: 14 feb. 2005.

RANGA, K. 3-D: Finite Element Analysis over the Internet using Java and VRML. 151p. 2000. Thesis (Master of Science) - School of Aerospace and Mechanical Engineering, University of Oklahoma, Norman, 2000.

RAPOSO, A.B. et al, 2000, Software Livre para Computação Gráfica e Animação por Computador. In: BRAZILIAN SYMPOSIUM ON COMPUTER GRAPHICS AND IMAGE PROCESSING, 13., 2000. *Tutirial...* [S.I.]: SIBGRAPI, 2000.

ROSENBLUM, L. et al. *Scientific Visualization*: Advances and Challenges. San Diego: Academic Press, 1994. 570 p.

STEVENS, A., WALNUM, C. Standard C++ Bible. New York: Hungry Minds, 2000. 839 p.

WATT, A. H., 2000, 3D *Computer Graphics*. 3ed. [S.I.]: Addison-Wesley, 2000.