An easy-to-use visualization tool for three-dimensional field computation and CAD purposes

Peter Berghmans, R. Belmans and Kay Hameyer
Katholieke Universiteit Leuven, E.E. Dept. Div. ESAT/ELEN
Kardinaal Mercierlaan 94, B-3001 Leuven, BELGIUM
Internet: http:\\www. esat.kuleuven.ac.be\elen\elen.html

Abstract

Nowadays, numerical field computation is a common tool to develop and design technical devices such as electrical machines, electrostatic actuators and other devices. Program packages are commercially available at the market. They meet the needs of almost 85% of the industrial engineers and are updated frequently. In those programs, all the graphical environment is integrated within the field computation environment.

To consider special physical phenomena or to investigate new algorithms and methods, several international research groups at universities are working in the area of numerical field computation. Those groups are developing own computer code for various kinds of three-dimensional numerical methods.

Such groups are interested to see the results obtained by their new algorithms graphically. Here, a visualization tool is introduced that is platform independent, inexpensive and of general application range. With a minimum of programming, the software is adapted to own needs and recommendations.

1 Introduction

Commercially available graphic programs are often slow, expensive and not optimized to the needs of this particular user group such as:

- graphical control of 2D/3D CAD models
- color graphics of 2D/3D field quantities and solutions
- platform independence of the software, because program development is performed under different operating systems
- interactive viewing and processing
- easy to use handling (mouse operated instead of command line operated)
- applicable and adaptable to arbitrary 2D/3D problem formulations

To meet those required specifications, a viewer for three dimensional colored objects is presented in this paper. The viewing tool is used as part of the existing software environment. The viewer contains several options such as rendering objects in real-time and light sources can easily be modified.

The main part of the tool is a viewer for VRML-models. VRML stands for Virtual Reality Modeling Language. This language is based on the Open Inventor file format from Silicon Graphics Inc. VRML becomes more and more a standard language within the World Wide Web for interactive simulation applications.

Its widespread use and acceptance as a language to show 3D objects on web pages can also be concluded from the effort several industrial software companies have undertaken to incorporate VRML interfacing routines in their packages. On the Internet, several file format translators to and from VRML are available[3]. Various CAD-packages have included VRML export possibilities (Pro/ENGINEER [4]) or external programs that do the conversion (AutoCAD [5]).

This viewer can be used as a stand alone application or can be installed as an Internet browser plug-in. With this property, the platform-dependency problem is now inherently solved, because Internet browsers are operating under UNIX and PC systems and many other platforms as well.

These viewers draw the VRML-model by reading and interpreting the appropriate file (.wrl extension). VRML-scenes are described in ASCII-text files. The viewer represents an open architecture that can be used as a graphical system to be adapted to various needs by employing simple input/output software routines. The data exchange is performed by ASCII files, generated by an interface routine in an arbitrary program and afterwards read by the viewer itself.

A minimum of software has to be written to develop a strong overall visualization tool. In the following section those parts of the VRML language will be discussed to complete a three-dimensional finite element pre- and post-processor visualization tool.

2 The visualization tool

2.1 The VRML Language

VRML is a language specification for the construction of a three-dimensional scene [1]. Though the language was designed for the construction of virtual reality environments, it seemed to be interesting for mechanical CAD and general field computation software, as described in this paper.

The language is based on different types of objects. Most obvious are the geometric primitives like cubes, spheres, lines,...Other objects are rotations, translations, light sources, color and reflection properties.

VRML is a general graphics language which contains more possibilities than are usually desired in a finite element environment. The language can be used to view both finite element models, meshes and computed results. To view models, some elementary rendering techniques are included which allow a more realistic appearance of the model. Different light sources (directional light, spot light) can be defined. In some VRML viewers, the ambient light can be modified during viewing. The color of the objects can be specified with separate values for the red, green and blue intensity. Objects can either emit light or reflect light in a diffuse manner.

For finite element models, it is important that polygons can be defined in a straightforward way. All data concerning one polygon can be grouped by means of the 'Separator{}' keyword. A list of 3D node coordinates is summed up followed by a list of node number sequences which form the polygon. To avoid rendering calculations by the viewer, it is good practice to define the color of the polygon as emissive.

Nodal coordinates can be referred to in the VRML file as a number. Because the coordinates do not have to be summed up more than once, the format of VRML can be qualified as compact.

For the graphical representation of computed field distributions, care has to be taken with shading effects. This shading effect could influence the interpretation of the color shaded plots. For this purpose, the color of the polygons has to be emissive. No light sources and no reflection properties are defined. However, sometimes the outline of the model may not be clear all light is declared as emissive. In this case, the color has to be diffuse, but comparison with a scaled color bar may be difficult. Different intensities of every color occur in that case. If the color green represents the average value of the scale, as is traditional in color bar scales, then different intensities of green occur in the picture ranging from full intensity green to almost black. Some shading is necessary to view the outlines, but on the other hand, too much shading makes interpretation of the colors more difficult. Fortunately, both effects can be combined at the same time allowing a reasonable trade-off.

To obtain color shaded plots for the field computation results, it is possible to either select one color per face of the polygon or select a color per node of the polygon. In the latter case, a linear interpolation of the color is then performed within the face of the polygon. This property is useful when viewing continuously varying properties of 3D finite element models.

Meshes can be visualized easily by means of the 'IndexedLineSet{}' keyword. A list of node number pairs forming the edges of the mesh is then given (Fig. 1).

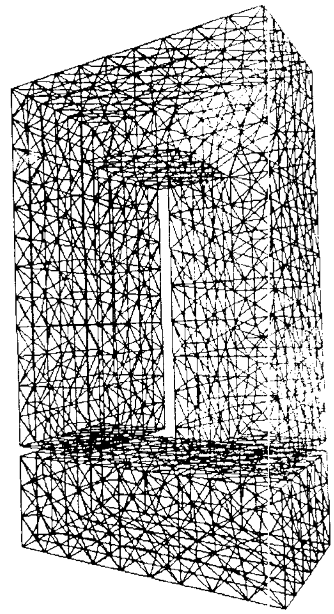

Fig. 1: View of the mesh of an iron core of an inductor.

In a more recent version of the language specification, VRML 2.0, it has become possible to include user activity and predefined behavior [2]. In this way, it is possible to visualize movements of geometrical parts in the model which can illustrate the working principle of the model. Mechanical vibrations and oscillations can be shown in this way as well.

2.2 Incorporation in field computation software

In order to view a model, a mesh or a color shaded plot of computed field values, a VRML file has to be created. To view the results, the file has to be loaded into the VRML-viewer.

A routine has been programmed which creates the VRML-file as a part of a 3D-FEM postprocessor. To use this routine, the user has to set the visibility of region labels he wants to see or not. The name of the VRML file is then entered followed by the desired background color. The user can then select to export either a model, a color plot for scalar potential solution values or a color plot for computed field strength values.

It is possible to join different models into one VRML-file. This can also be done in a Text Editor by means of a simple Copy and Paste operation because the file format is ASCII and because of the hierarchical structure of the file format. In some finite element packages, model meshes and coil meshes are stored separately. All these parts can be joined into one VRML-file from within the interface.

A non-trivial example of this joining is given in figure 2. In this figure, a 3D model of a double layer end-winding of an electrical machine excited by permanent magnets has been joined with a flux line distribution of an associated two-dimensional calculation. To achieve this, an interface has been programmed in a 2D-FEM postprocesor to achieve the flux line image in a VRML file. The main part of the latter file was copied into the VRML file for the 3D model. Finally, the 2D flux plot was grouped together by means of 'Separator{}' and an appopriate 'Transformation{}'.

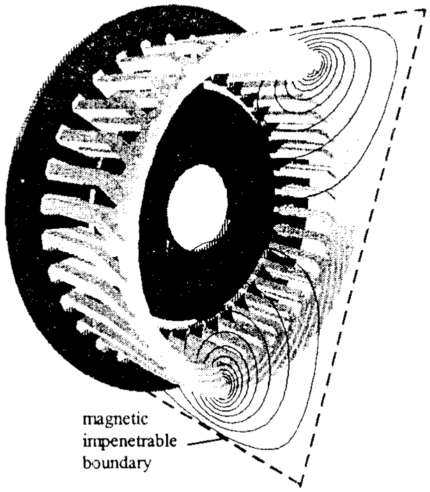

Fig. 2: Combination of 3D model with equipotential lines from 2D calculation

Another example of this simple joining possibility is shown in figure 3. In a research project on high-voltage lines, various models for the towers and lines were built. A post-processing interface allows to add planar slices of the 3D solution into the tower model (Fig 3).

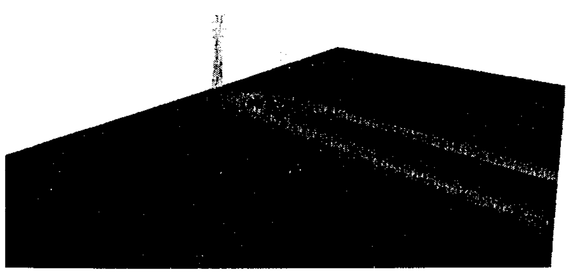

Fig. 3: 3D model of a high voltage tower with planar slice.

One obvious shortcoming is the fact that for every new 3D image, a new file has to be created. For example, changing the background color can not be done online. The ASCII-file has to be edited by hand. Both the program with the VRML-interface and the viewer have to run at the same time. This is no problem since all popular operating systems nowadays support multi-tasking.

An example of a model which has been edited by hand is given in figure 4. It displays a quarter of an iron core with a double layer coil. Here, rectangles have been added to explain the process of extrusion based mesh generation for educational purposes.

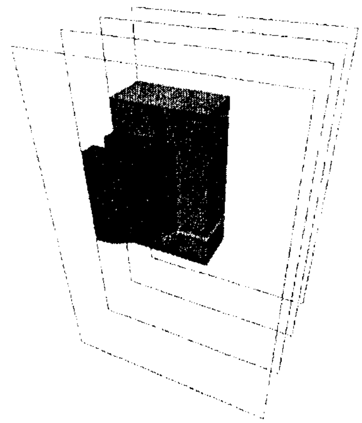

Fig. 4: Iron core with double layer coil. Rectangles are added to visualize the process of the extrusion based mesh generation

Since the VRML file format is compact and very flexible when it comes to blank spaces and carriage returns, it takes very little time to create the file. Once the file has been loaded into the viewer, the graphic can be rotated translated and zoomed in and out by means of simple mouse clicks. No user command line input is necessary.

To summarize, the following advantages and disadvantages can be listed:

Advantages

- cost-free
- platform-independent
- standard language
- compact file format in ASCII
- fast visualization and transformation of the model
- easy to use

Disadvantages

- new file for each graphic (change) is necessary

3 Example viewed by the visualization tool

The viewer used here is a viewer delivered with the HP-UNIX operating systems as a stand alone application. The object itself is located in a 3-dimensional Cartesian co-ordinate system.The user of the viewing program looks through a lens of a virtual camera onto the object.

Considering these aspects, the viewer can be used during the preprocessing stage for geometrically complex models. When the mesh has to be made by extrusion techniques in one direction, it is sometimes useful to have a 3D view of the model to see for errors in the model.

Figure 5 shows an example of a double layer end-winding of an electrical machine excited by permanent magnets. The buttons on the right of the window are used for the transformation of the viewing point. The ruler at the bottom of the window is used to change the ambient light. The model can be rotated by dragging the mouse within the viewing window.

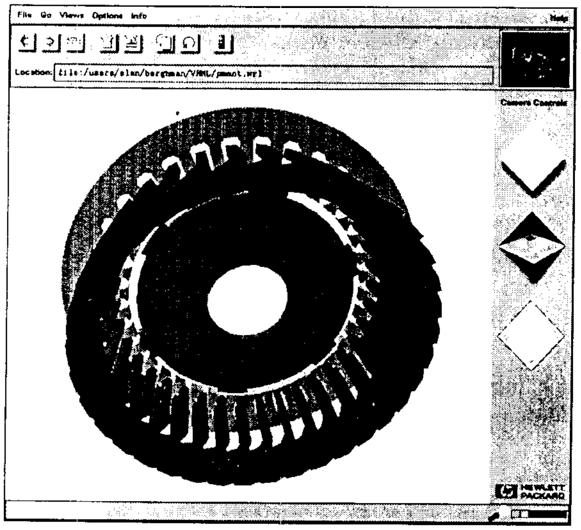

Fig. 5: View of a three dimensional finite element model.

4 Incorporation into WWW-page

The VRML-models can be incorporated in an Internet home page. The introduction into a web page can be achieved in several ways:

- load directly into a Web browser, filling the page
- embed into a page, filling a page rectangle
- load into a page frame

Within a page frame, the viewer can either fill the entire frame or it can be embedded in a rectangle inside the frame. In this way it is possible to create an frequently needed rectangular color bar with a scale. In this way, the colors on the model which represent field quantities can be interpreted to their numerical value. Another possibility is to use different frames for different views of the model, i.e. a 3D view accompanied by a top view, a frontal view and a side-view

Links to standard HTML-pages can be assigned to faces of the model in order to add comments or explanations to the shown object.

5 VRML and Java3D

To obtain a visualization and interactive processing tool in a field computation package, one can apply the above described VRML viewer with interfacing routines or one can employ Java3D. Java3D is an API which means that it can be part of a complete field computation package written in Java.

Java3D shares a lot of advantages with VRML, e.g. cost-free, platform independent, etc. VRML file loaders will be included in Java3D. However, 3D scenes created with Java3D are not necessarily stored into a file. VRML is certainly easier to learn. Java3D offers more possibilities concerning interactivity, i.e. it is more powerful. The use of Java3D requires better programming skills.

At the time of writing, the specification for Java3D is available via the Internet and at bookshops. An alpha release of the Java3D API is available for downloading and it is expected that the final release will be in late 1998. It is expected that Java3D will include features presently not included in VRML[7].

Conclusion

The VRML language, which was originally meant to operate general 3D environments on the Internet, can be easily incorporated within an existing finite element package for all visualization needs. In other words, it can replace third-party graphics programs and furthermore it is no longer necessary to program graphic code.

The only requirement is that interface routines, generating ASCII files, have to be written adapted for the different types of visualization. The platform independence ensures the portability of the field computation program as a whole.

As can be noticed in the reference list, the most recent documentation can be found on the Internet.

Acknowledgments

The authors are grateful to the Belgian "Fonds voor Wetenschappelijk Onderzoek Vlaanderen" for its financial support of this work and the Belgian Ministry of Scientific Research for granting the IUAP No. P4/20 on Coupled Problems in Electromagnetic Systems. The research Council of the K.U.Leuven supports the basic numerical research.

References

[1] VRML 1.0 language specification, http://www.vrml.org/VRML1.0/vrml10c.html, 1996

[2] VRML97 language specification, http://www.vrml.org/Specifications/VRML97/

[3] The VRML Repository, VRML File Translators, www.embl-heidelberg.de/vrml/gmtranslators.html

[4] VRMLout for AutoCAD (R13 and R14), www.cadastudio.cz/vrmlout.htm

[5] Pro/ENGINEER VRML Design Gallery, http://www.ptc.com/products/vrml/index.htm

[6] Introduction to VRML97 and Java3D, Course slides, http://www.sdsc.edu/~nadeau/Courses/Visualization97/slides

[7] Java3D FAQ, http://www.javasoft.com/products/java-media/3D/forDevelopers/java3dfaq.html