TSINGHUA SCIENCE AND TECHNOLOGY
ISSN 1007-0214 05/67 pp26-33
Volume 13, Number S1, October 2008

CAD and CAE Integration Through Scientific Visualization Techniques for Illumination Design

Tiago Martinuzzi Buriol, Sergio Scheer**

Graduate Program in Numerical Methods for Engineering, Civil Engineering Research Center, Federal University of Parana, Curitiba 81531-980, Brazil

Abstract: Engineers tend to use different software to perform tasks such as geometry modeling, database management, numerical analysis, and visualization. This may cause decrease of productivity and loss of information during the conversion process between different data file formats. This paper presents a computer aided design (CAD) and computer aided engineering (CAE) system integration using scientific visualization tools and techniques. It deals with the development of a 3D CAD add-in for lighting analysis which uses the CAD model as 3D interface for creating a lighting scheme, processing, and visualizing 2D or 3D illuminance fields. Visualization features as color and contour mapping were developed using the visualization toolkit (VTK) toolkit. The application integrates all functionalities of the 3D CAD with tools for light sources database management, pre-processing, processing, and post-processing of illuminance fields in a single environment. This approach increases productivity and eliminates the need for different software.

Key words: illumination design; scalar fields post-processing; 3D computer aided design (CAD); visualization toolkit (VTK); systems integration

Introduction

For a same project, engineers tend to use different software to perform different tasks such as geometry modeling, database management, numerical analysis, and data visualization. In lighting plan, for example, usually a computer aided design (CAD) software is used for geometric modeling, and a lighting computer aided engineering (CAE) tool is used for the illuminance field processing. Additionally other tools and software may be needed for a more detailed study, such as scientific visualization toolkits. This may cause decrease of productivity and loss of information during the conversion process between different data file formats. Being aware of this fact, important industries such as electric power, oil and gas companies are looking for integrated systems, that allow users to edit, process, visualize and transfer geometric models, numerical analysis results, and other project data quickly and efficiently[1,2].

Visualization systems can be used as powerful communication tools and 3D digital models can be used as intuitive user interfaces. In architecture, engineering and construction (AEC), for example, virtual environments have been used as support tools for communication and planning[3]. Also, in science and engineering, 3D visualization techniques have been used as a powerful way to show large amount of information in a compact form. Moreover, these techniques help in suggesting behavior and affinity among different entities given their placement in the virtual world[4]. Thus, user interface paradigms such as virtual reality technology, can also be a good inspiration for an integrated system platform that allows different

Received: 2008-05-28
** To whom correspondence should be addressed.
E-mail: scheer@ufpr.br

software application capabilities. Unfortunately, many of the features provided by the powerful visualization toolkits available today (e.g., visualization toolkit (VTK) and OpenDX) are not yet fully explored in most CAD and CAE systems.

In many cases, the user wants to visualize geometric CAD models and other complex numerical results simultaneously and interactively, in order to increase the understanding of a complex phenomenon under study. But this kind of visualization model is not easily constructed. Many numerical analysis tools do not support CAD format files or not allow the interaction with numerical data simultaneously to display other geometric models. Scientific visualization techniques such as contours and color mapping, slice and cut plane, glyphs, stream lines and ribbons, and many others, provide ways to explore large and multidimensional data sets. Thus, virtual environments and visualization techniques can be used to communicate design intent, and to compare and evaluate design options.

This paper describes a lighting analysis application which consists of a SolidWorks 3D-CAD add-in that integrates management tools for fixture database, mesh generation algorithms, numerical processing, and post-processing of illuminance fields. Basically the application uses the CAD model as a 3D interface in order to plan the lighting scheme (luminaries choice and positioning), process the illumination in 2D or 3D meshes chosen by the user, and then, visualize and explore the results through scalar field scientific visualization techniques (contours, cut, slices, color mapping, etc.). Visualization features like color and contour mapping were developed using the VTK. The final visualization of the 3D model can be exported in a virtual reality modeling language (VRML) standard file format. This kind of model in VRML format can be easily visualized in a web browser.

The basic steps for the presented application use are: (1) Create or import your 3D CAD model; (2) Choose a light source, insert and target it into the model (repeat if necessary); (3) Choose a 2D or 3D mesh; (4) Process the illuminance field; (5) Visualize the result with visualization techniques.

The development of the application searches for an integration of the 3D CAD functionalities with tools for light source database management, pre-processing, processing and post-processing of illuminance fields in a single environment. This approach increases productivity and eliminates the need for different systems.

1 Visualization Techniques and Tools

1.1 Some visualization system basic concepts

Humans can recognize states, structures and objects as well as model behavior through different kinds of visualization. They are used to communicate complex content and support decision processes. Scientific visualization (SciVis) refers to the area dedicated to displaying physical or scientific data, usually obtained through numerical simulations or sensors. "…Visualization offers a method for seeing the unseen. It enriches the process of scientific discovery and fosters profound and unexpected insights …"[5].

The goal of SciVis is to enhance scientific productivity by utilizing human visual perception and computer graphics techniques. Therefore SciVis is a supporting technology that enables scientists and engineers to understand complex relationships typically represented by large amounts of data.

Many visualization techniques and algorithm have been proposed in the literature[6]. Some of them may be more appropriate for scalar data and others for vectors or tensors. On the other hand, some techniques are based in texture and others in geometric reconstruction such as line and surface contours. According to Collins[7], some SciVis techniques have their first records dating of the XII century, and are still used and implemented in many computational tools. Some well-known techniques for visualization of numerical data are mapping scalar values into colors (colormap) and contours (level curves and surfaces).

The process of displaying datasets usually passes through three steps, which are: data acquisition, transformation (filtering and mapping) and rendering[8]. Figure 1 illustrates the dataflow in a visualization pipeline.

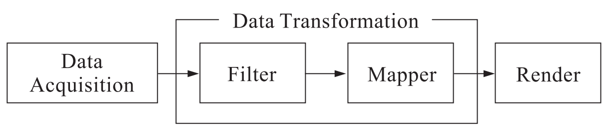


Fig. 1 Dataflow visualization model

This paradigm is called dataflow visualization model and is currently used in many visualization systems and tools[6-8]. Therefore, visualization techniques involve data processing algorithms, which extract interest data of a sample and convert into a suitable form for representation.

To enrich and improve simulation data exploration, more natural, three-dimensional, and highly interactive and visual techniques are needed. Interactive visualization algorithms and applications can be more easily developed using visualization toolkits or libraries. In this work the visualization toolkit (VTK) was used[8]. VTK is currently one of several visualization toolkits available.

1.2 Visualization toolkit

VTK is an open source and free software system for 3D computer graphics, image processing, and visualization (http://www.vtk.org). It consists of a C++ class library, and several interpreted interface layers including Tcl/Tk, Java, and Python. Some advantages of VTK are the support to a wide variety of visualization algorithms including scalar, vector, tensor, texture, and volumetric methods, and advanced modeling techniques such as implicit modeling, polygon reduction, mesh smoothing, cutting, contouring, and Delaunay triangulation. It provides a variety of data representations including unorganized point sets, polygonal data, images, volumes, structured, rectilinear, and unstructured grids. It comes with several reader/importers and writer/exporters to exchange data with other applications (including 3DS and VRML formats). Hundreds of data processing filters are available to operate on these data, ranging from image convolution to Delaunay triangulation. VTK's rendering model supports 2D, polygonal, volumetric, and texture-based approaches that can be used in any combination.

Using VTK is possible to develop visualization algorithms connecting filters and mappers through input/output process in such a way as creating a visualization pipeline or, alternatively, visualization network. For example, one may wish to read a set of unorganized points, create a polygonal mesh via Delaunay triangulation, then display the mesh using polygonal (surface) rendering. It is important to note that data connections may only be made when the input/output types match.

VTK consists of two major pieces: a compiled core (implemented in C++) and an automatically generated interpreted layer. Data structures, algorithms, and time-critical system functions are implemented in the C++ core. Common design patterns such as object factories and virtual functions ensure portability and extensibility. Since VTK is independent of any graphical user interface (GUI), it does not depend on the windowing system. It hooks into the window ID and event loop lets developers plug VTK into their own applications. An abstract graphics model achieves graphics portability. The graphics model is like an abstract layer above the OpenGL graphics language to ensure cross-platform portability. Figure 2 illustrates the graphics model as an abstract layer above the OpenGL graphics language to ensure cross-platform portability.

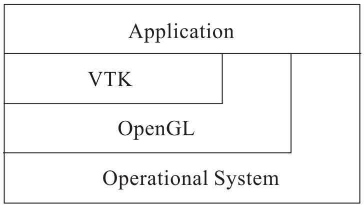


Fig. 2 VTK layer

In this work VTK was used to develop two main scalar field visualization algorithms. The first algorithm was used for 2D color mapping and image generation. The other one for 3D scalar mapping and isosurface generation.

1.3 Virtual reality modeling language

The virtual reality modeling language (VRML) is a non-proprietary ISO standard file format for describing interactive 3D objects and worlds. VRML is designed to be used on the internet, intranets, and local client systems. It is also intended to be an universal interchange format for integrated 3D graphics and multimedia. VRML may be used in a variety of application areas such as engineering and scientific visualization, multimedia presentations, entertainment and educational titles, web pages, and shared virtual worlds[9].

Three-dimensional information can be easily generated and transferred through the Internet by this technology. VRML models can be displayed in a number of freely available browsers and integrated with web browser plug-ins. Information in a VRML file is encoded in ASCII format. It is important to note that there is a VRML's latest incarnation, the X3D[9], which updates the former to more modern standards, including additional graphic features and a XML encoding.

VRML describes 3D models in the form of 'nodes'. Nodes generally define 3D physical descriptions that may be made up of 3D primitives, such as spheres, cube, cones and cylinders, or of complex polyhedra composed of polygon facets. In addition to these form descriptions, nodes can also define materials, colors, texture maps, lighting, shape transformations and viewing criteria. Another characteristic of VRML file is the ability to compose files together through inclusion and hyperlinking. The essential characteristic is that VRML is intended to be used in a distributed environment such as the World Wide Web.

2 Illuminance Field Processing and Visualization

An appropriate lighting plan for a built environment is important to provide comfort and good working conditions. Moreover, it generates economy in a number of ways, such as reducing electrical energy consumption. Architects and engineers are using lighting software programs to assist in the building design and facilities retrofit. In general, lighting programs perform analysis using the point-by-point method, which is very accurate and precise and can also be used for graphical renderings.

Point-by-point method is a design procedure for determining the illuminance at various locations (or at a mesh) in a lighting system using luminaire photometric data. For instance, these data can be described in an Illuminating Engineering Society (IES) file format[8]. This IES standard format can contain fixture product data from any manufacturer and can be put into a database. In addition to photometric data, the point-by-point method needs the location and focus direction of each light source, which can be obtained from a CAD model. In this context, a 3D virtual environment can provide a great 3D graphical interface for the illumination planning software application.

An example of geometric information needed in the illuminance field calculation and the photometric web for a simple lamp is shown in Fig. 3 as follows: L is the light source location, D is the focal point, P is the point of interest, n is a normal vector of plane A and γ is the angle between PL and n.

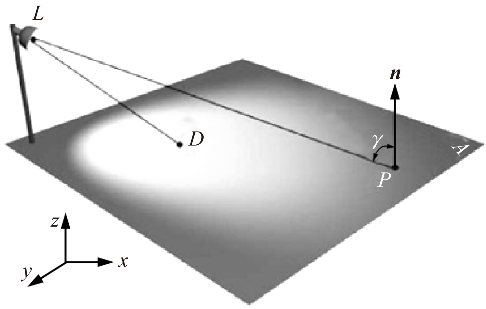


Fig. 3 Geometric information for illuminance field calculation

With the already built CAD model, the lighting design can be built using the geometry of the CAD model to define the points of installation of fixtures and point of focus. Thus, the coordinate system used in the point-by-point processing method will have the same coordinate system of CAD model.

3 Lighting Plan Application

The illumination field processing, as mentioned, needs some input information such as photometric data of light sources, design geometry (includes coordinates of each luminaire position and focus direction) and point coordinates where the illuminances will be calculated.

The add-in implementation was made of four modules. The first deals with luminaire register in a SQL database. The second is for the geometry illumination design definitions such as light localization and focus direction for each light source. The third module is related to the mesh choice and illuminance field processing. Finally, the last module is for result visualization and exportation of the complete visualization model in a VRML file for publish.

For the first module it extended the SolidWorks GUI (Graphics User Interface) using the API (Application Programming Interface) available as highlighted in Fig. 4. A graphical interface (called page in the SolidWorks environment) was created for adding luminaire models and few information related to light sources in the database. The SolidWorks workspace was used as a 3D interface to input the necessary data for illuminance calculation. This allowed an easy and interactive choice of light source, location and focus direction. With the 3D CAD model interface the user can select points in objects of the model by simple mouse pick and then insert light sources chosen from files in a database. This process can be repeated as often as desired.

The light point and focus coordinates are stored in an input file for the illuminance field processing. In SolidWorks, the customization (add-in) allowed also to create graphical interface features for access and edit the database containing IES files of many light source models as showed in Fig. 4.

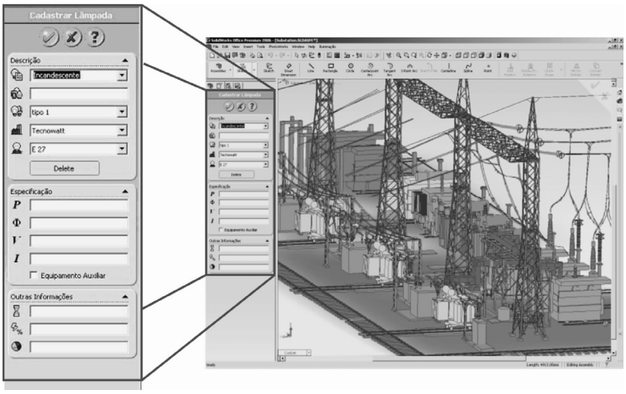


Fig. 4 Add-in interface for edit IES file database

The second module was developed using CAD features for the point coordinates capture and created pages for the luminaire choice and insertion in the digital model. Once defined an illumination scheme is possible to save it for subsequent restoration and/or edition. Many illumination designs can be created and compared for get the best result.

The third module uses the saved illumination schemes in order to process illuminances in a mesh defined by the user. The mesh is a regular set of points evenly distributed in any plane or box (volume) desired. The point-by-point algorithm will calculate the direct lighting at each point.

Finally, the last module post-processes the illuminance field through the visualization algorithm (contours, colormap, slice planes, and stream lines).

The choice for SolidWorks is only an alternative way to input data for the illuminance field processing algorithm. This data input can also be made through an ASCII file. Therefore, the use of this CAD system is not mandatory. The developed numerical algorithm can read the input data from an ASCII file and generate an output file in VTK legacy format. This format can be easily read by a visualization algorithm written using the VTK library.

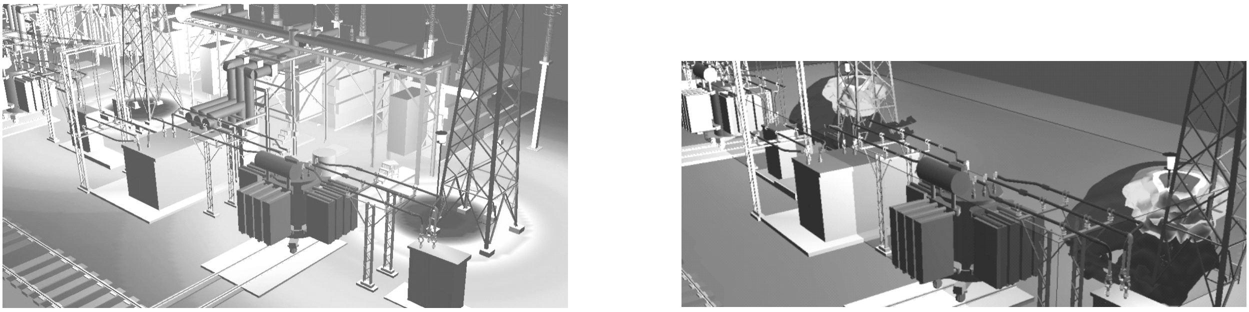
In this sense, the numerical algorithm read the input data, process the illuminance field and store the results in a VTK legacy file. Then, the visualization algorithm reads, generates the color mapping and contours, and also, exports the final model to the VRML format. Therefore using the SolidWorks interface is possible to perform several simulations for different lighting systems in different study cases and compare them. The steps for the use of the developed application are: after registering the IES files in the database for all light sources which will be used, it is necessary to choose and insert in the 3D digital model the points of light desired; select or create a face (plane) or a volume (box) where the analysis will be performed; choose density for the mesh points to be generated; and, finally, process the illuminance data and get the visualization. The generated mesh is of structured type and has square cells (pixel cells). From the color mapping result, this kind of structure allows to generate an image file in JPG, BMP or PNG that can be inserted into the model as a texture, which usually is more economic.

To illustrate the operation of the developed application, a simulation was performed for a lighting scheme, using an electrical power substation 3D model. Several configurations could be tested varying the position of the light sources, different luminaire and lamp types. Figure 5 shows both 2D (top) and 3D (bottom) illuminance field visualization.

In addition, it was possible to perform simulations taking into account the existence of opaque bodies which block the light in some points of the mesh. This was developed using available methods in the SolidWorks API. The methods used were the RayIntersections and GetRayIntersectionPoints that allow given a point and one direction, to verify if there is any object in that direction. With these methods the application was able to examine whether or not there is any object between the point of light source and the point for which the illuminance is to be calculated. If there is an object, the influence of the light source will be disregarded, or will only be computed for those light sources whose light rays focus directly on the point. An example performed is shown in Fig. 6.

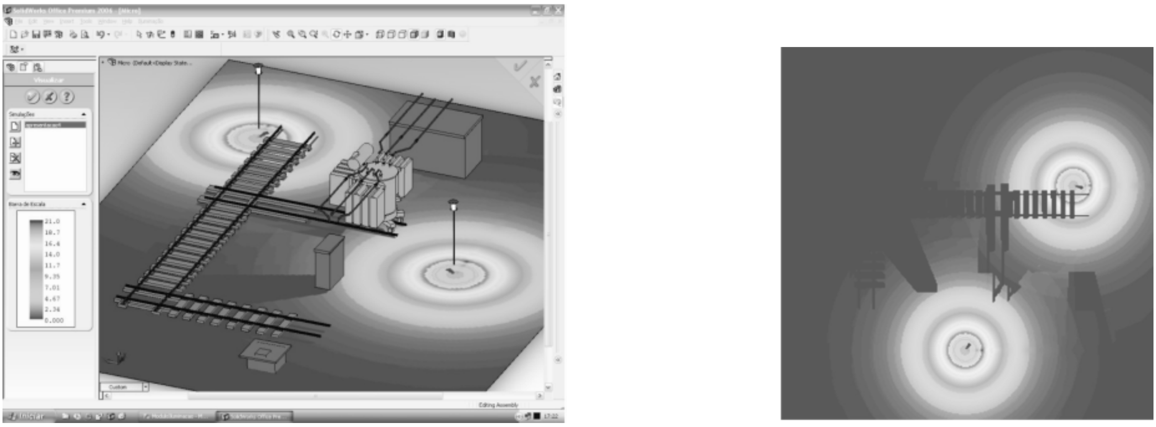
The used approach was without the influence of the light sources whose luminous intensity does not directly reach certain point. Despite of this, due to the amount of computational effort needed, it was only possible to make simulations in parts of the power substation, or in smaller models.

A complete data flow diagram of the developed lighting application is showed in Fig. 7.



(a) 2D (b) 3D

Fig. 5 2D and 3D illuminance field visualization



(a) (b)

Fig. 6 (a) Result of simulation performed considering the opaque body obstructions. (b) Image generated for 2D illuminance field visualization

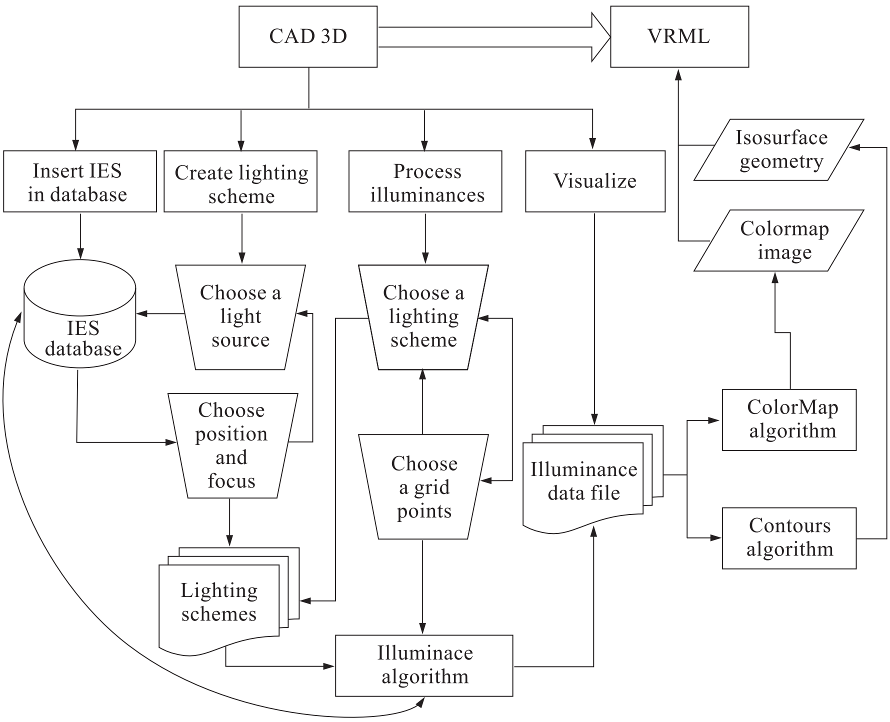


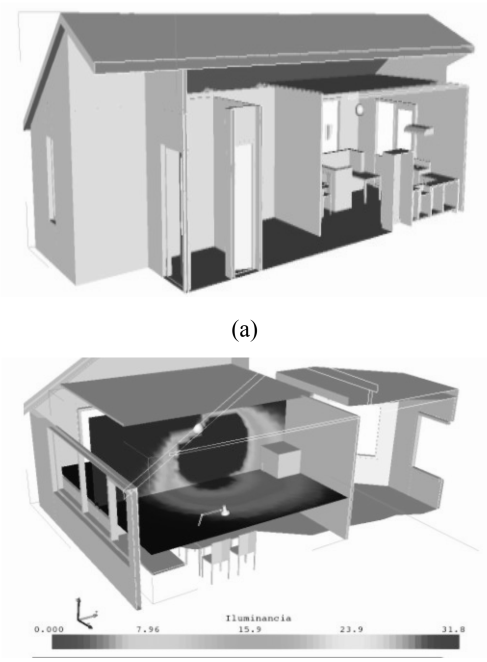
Fig. 7 Dataflow and use of the developed application

The main advantage of the developed application, in comparision with other existing programs, is that it is fully integrated with a 3D CAD system and is designed with an open source toolkit (VTK) which makes it more flexible and expandable, as well as free of charge.

4 Data Viewer Application

In order to investigate deeper the potentials of the VTK for development visualization systems, a program named Data Viewer was developed[10]. Data Viewer, or simply DV, is a visualization program based on VTK which provides a graphical user interface to control several parameters of visualization algorithms. DV can easily be used to load a dataset (in VTK legacy format file) and apply visualization techniques on it, such as slice planes, color mappers and contours. A sequence of filters and mappers can be applied for a visualization pipeline construct[6,10]. Finally, the resulting model can be exported in a VRML file.

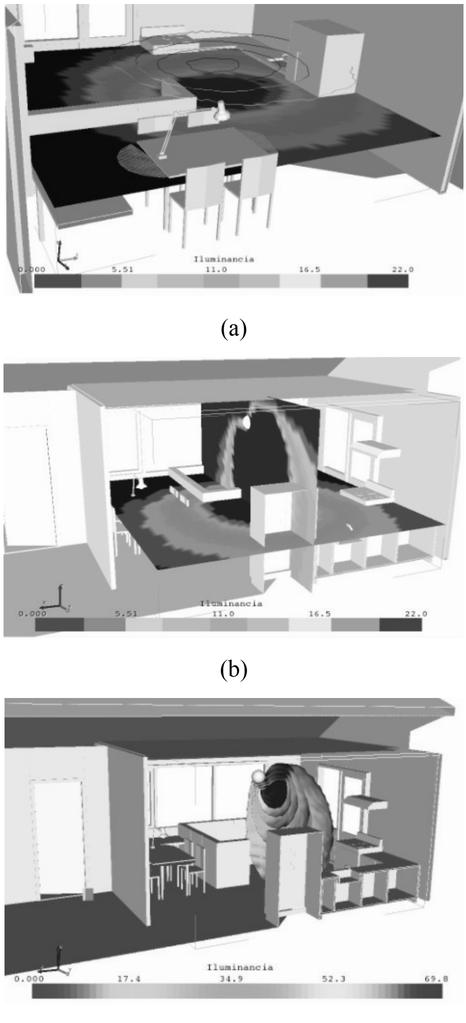
A simple application for illumination field visualization was created. As an example, the kitchen geometric model was constructed in a popular CAD program, the illuminance field was processed separately and then the results loaded in the DV application were the resulting illuminance field created. This procedure allowed a complete understanding of lighting data. Figures 8 and 9 show some views obtained by some DV capabilities such as 3D scalar fields and geometric model integration in VRML format.



(a)

(b)

Fig. 8 Illumination field visualization in a house kitchen: the CAD model and two orthogonal thin slices for the illuminance colormaps and a single thin slice viewed together with isolines



(a)

(b)

(c)

Fig. 9 Illumination field visualization in a house kitchen: two orthogonal thin slices of illuminance and colormaps, and illuminance isosurfaces

5 Conclusions and Future Work

Some basic concepts of visualization techniques and tools were addressed from the point of view of software integration with application in lighting analysis and design. A development for CAD and CAE systems integration using Scientific Visualization tools and techniques was presented.

This paper has shown that through customization of a 3D CAD it is possible to incorporate algorithms for pre-processing, processing and post-processing of numerical analysis in order to obtain a new application. A 3D CAD add-in for lighting analysis which uses a CAD model as 3D interface for creating a lighting scheme, processing and visualizing 2D or 3D illuminance fields was presented. Two examples of use were presented.

The integration of different software features into a single system environment (a CAD or a Virtual Reality system) increases productivity and may eliminate the need for different software.

In particular, CAD systems are commonly used in engineering for geometric modeling and can be customized, so as to seize the geometry built to generate the point meshes that could be used in numerical analysis. In addition, CAD systems that support use of textures, such as SolidWorks, allow numerical data to be represented as images in the 3D model, as shown for viewing the illuminance fields through color mapping.

The VTK toolkit provides algorithms for mesh generation and scientific data representation, and may be used in the development of software customization in order to insert new capabilities. In addiction, the VRML file exporter/writer provides some facilities with respect of communication and transfer between data visualization models.

References

[1] Savioja L, Mantere M, Olli I, et al. Utilizing virtual environments in construction projects. *ITcon*, 1993, **8**: 85-99.

[2] Malandrino D, Palmieri G, Scarano V, et al. Visualizing processes on the web. *Journal of Visual Languages and Computing Archive*, 2007, **18**(6): 592-612.

[3] McCormick B H, DeFanti T A, Brown M D. Visualization in Scientific Computing. *Computer Graphics*, 1987, **21**(6): 153.

[4] Brodlie K W, Carpenter L, Earnshaw R A, et al. (eds). Scientific Visualization, Techniques and Applications. New York: Springer-Verlag, 1992.

[5] Collins B M. Data visualization: Has it all been seen before? In: Earnshaw R A, Watson D, eds. Animation and Scientific Visualization: Tools & Applications. Academic Press, 1993.

[6] Schroeder W, Martin K, Lorensen B. The Visualization Toolkit, 3rd Ed. USA: Kitware, 2004.

[7] Web3d. Extensible 3D (X3D) —international draft standards. ISO/IEC FDIS 19775: 200x. http://www.web3d.org/x3d/specifications/index.htm, 2008.

[8] IESNA. Illuminating engineering society of North America. http://www.iesna.org/, 1993.

[9] Buriol T M. Scalar fields processing and visualization in virtual environments and 3D CAD systems applied to power substation illumination design plan [Dissertation]. Curitiba: Federal University of Parana, 2006. (in Portuguese)

[10] Silva Neto M A, Buriol T M, Scheer S. Um visualizador interativo para exploração de dados volumétricos obtidos em análise pelo método dos elementos finitos. In: XXVIII Congresso Ibero Latino Americano de Métodos Computacionais em Engenharia. Porto, Portugal, 2007: 1-17. (in Portuguese)