DATA VISUALIZATION OF LOW VOLTAGE NETWORK DESIGNS USING A DISTRIBUTED VIRTUAL REALITY ARCHITECTURE

M. Moutzouris, B. Dwolatzky

Software Engineering Applications Laboratory
Department of Electrical Engineering
University of the Witwatersrand, Johannesburg
Private Bag 3, WITS, 2050
South Africa
m.moutzouris@ee.wits.ac.za,b.dwolatzky@ee.wits.ac.za http://www.seal.ee.wits.ac.za

ABSTRACT

South Africa and other developing countries are currently engaged in a mass electrification drive. This requires a good software infrastructure for designers. The current generation of tools are based on two-dimensions and consist of stand-alone tools. This paper introduces a data visualization software package that allows the designer to immerse himself into the 3D environment that has been created from a 2D CAD drawing. The electrical and physical parameters of the objects in the environment can be viewed and manipulated. The software is supported by a distributed virtual reality software architecture, which allows for networked low-cost personal computers to handle the burden of creating this environment and interacting with this world. The software presented in this paper will be tested to determine how useful data visualization is in this application.

Keywords: data visualization, virtual reality application, distributed virtual reality, electrical reticulation design tool.

BACKGROUND

In 1993 two-thirds of all homes in South Africa had no access to electrical energy. A major electrification programme was undertaken and is currently connecting 1000 houses per day to the national electricity grid. Many of these new connections are being made in informal settlements close to major cities and in rural areas.

Electrification has placed great demands on the designers of electrical distribution networks. Hundreds of new systems are being designed each year and designers are expected to make a significant contribution to keeping the cost of these systems as low as possible, while ensuring that high construction standards are maintained. Computer tools obviously have a significant role to play in supporting designers in this task.

INTRODUCTION

The methodology currently used by the designers of large low-cost electrical distribution networks depends on a range of non-integrated computer tools. A two-dimensional CAD system is used to prepare the town plan of the area to be electrified. The layout of the network is usually developed in pencil on hardcopies of the CAD drawing. Specialised stand-alone software is used to calculate voltage levels and currents. These programs require the manual input of data relating to conductor lengths and the connectivity of the network. Spreadsheets are used to determine the material and labour costs associated with a design. Detailed construction drawings are produced by a draughtsman on a CAD system once all design decisions have been finalised.

Over the past few years considerable effort has gone into developing software tools which streamline and improve this design methodology [Dwolatzky-Meyer98] with the goal of helping the designer to optimise his designs. In developing such tools the priority has been to integrate all of the design activities within a common software environment thus eliminating manual re-entry of data and other time-consuming and wasteful tasks. In this way the designer is free to easily and quickly explore different design options and thus produce cheaper networks.

An integrated design environment has been developed around a commercial CAD package [Dwolatzky-Meyer98] and is beginning to be used in South Africa by many of the designers working on electrification projects. A major limitation of this two-dimensional design tool, however, is that it is difficult for the designer the deal with the large amount of data relating to voltages, currents and costs. The move from a paper-based design methodology to an integrated CAD-based one has not had a significant impact on the visualisation of data. The designer still sees voltage and current values represented as numerical text on a flat drawing of the town's layout.

It has become apparent that, by correctly using the powerful graphical capabilities of the current generation of computers, data can be visualized in various ways that would allow a better understanding of a design.

This paper deals with an ongoing research project which aims to explore the application of advanced MMI techniques to the visualisation of data within an electrical distribution design environment. How, for example, could Virtual Reality help a designer to reduce the costs of an electrification project?

Virtual Reality (VR) allows us to interact with a reality that does not exist outside the environment of the computer on which it runs. VR allows us to visualize data in ways that would be impossible within the physical world. We cannot, for example, see the variations of voltage and current in an actual electrical network. Using VR, however, it would be possible to allow the designer to interact with the electrical network within a realistic 3D environment in which important parameters such as voltage and current are represented visually.

VR and other advanced MMI techniques are usually associated with expensive high performance graphical workstations. Given this the question is: how practical is it to suggest the application of such techniques to support the design of low-cost electrification projects? The approach adopted in the research project presented in this paper has been to develop a "Distributed VR Software Architecture" [Blumenow-Spanellis-Dwolatzky97] which allows computationally intensive advanced MMI techniques to be implemented on a network of standard "office" personal computers.

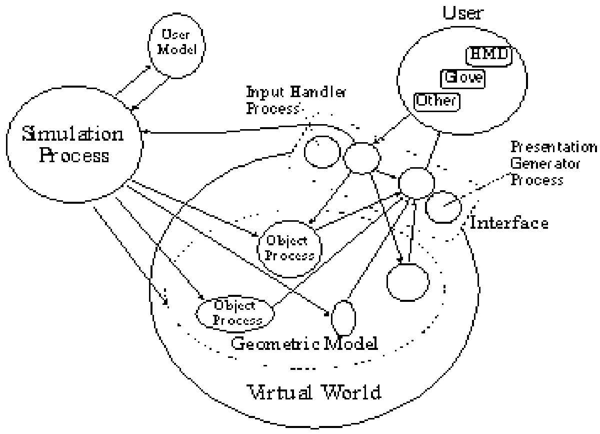
DISTRIBUTED VIRTUAL REALITY ARCHITECTURE

A Virtual Reality application can be implemented across a number of networked processors using a software architecture we have called the "Process Agent Model" (PAM) [Blumenow-Spanellis-Dwolatzky97]. The PAM makes details of the distributed processing and network communications transparent to the high level processes.

On each computer there exists either the authentic process or its agent process. The agent process acts as representation of the authentic process. The agent intercepts messages destined for the process it represents and passes these messages through the network to the authentic remote process.

The user of the system interacts with the virtual environment with input and output devices (see Fig.1). The interface between the user and the virtual world consists of the Presentation Generator and the Input Handler. A Geometric Model is implemented within the virtual environment from a number of object processes. The Input Handler can initiate changes to the Object Processes and also provide inputs into the Simulation Process. The Geometric Model incrementally communicates net changes to the Presentation Generator which in turn renders the virtual environment on the display. The Simulation Process handles the simulation aspect which may receive information from the Input Handler, and from objects in the Geometric Model. The conceptual system model decouples the simulation aspect from the rest of the system.

The network communications infrastructure makes use of raw Ethernet frames to communicate between remote processes on remote computers. An intelligent dual incoming queue mechanism handles the incoming and outgoing traffic on each computer. The mechanism is intelligent in that only the latest update from each process is retained.



Conceptual Virtual Reality System Model

Figure 1

VISUALIZATION

"DataVis" is the name that has been given to the data visualisation software described in this paper. The software interfaces with an integrated set of tools, called CART 3.0 [Dwolatzky-Meyer98] developed to assist the designer of electrification schemes. DataVis allows the designer to visualize the electrical quantities within a single transformer zone. A transformer zone is an area that encloses all those consumers connected to a single transformer.

DataVis can be run on any standalone or networked computers that are capable of running Microsoft Windows™ NT or 9x. This platform was chosen because the PAM (see previous section) was designed on this platform and because most designers of electrification networks work on Windows-based PCs.

Using CART 3.0 the designer is presented with a 2D CAD drawing showing the town plan of the area to be electrified. He places symbols representing transformers, poles and junction points on the map and draws lines representing the conductors which connect each consumer to a transformer. CART 3.0 has tools that can be invoked to allow the designer to determine the voltage at any point in the network and the current in any conductor. There are also tools to prepare detailed bills of materials and an accurate cost of the network.

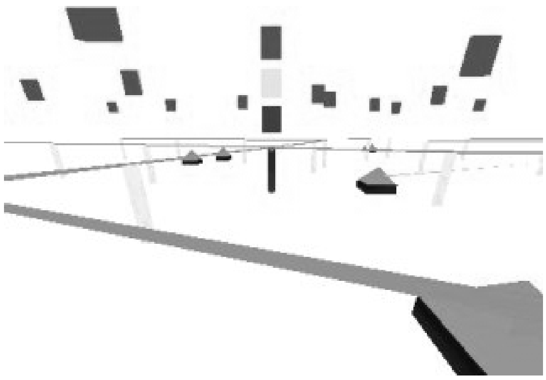
DataVis can be invoked from within CART 3.0. A text file containing the following data is produced for the currently selected transformer zone:

- Geographic co-ordinates (x-y co-ordinates) of each consumer's connection point, each pole and the transformer;

- The connectivity between the transformer, poles and consumers;

- Details of each pole (height and label) and the number of consumers connected to each electrical phase on the pole.

- Descriptions (type, resistance, maximum current rating, etc.) of each feeder and service conductor.



Sample screenshot from DataVis

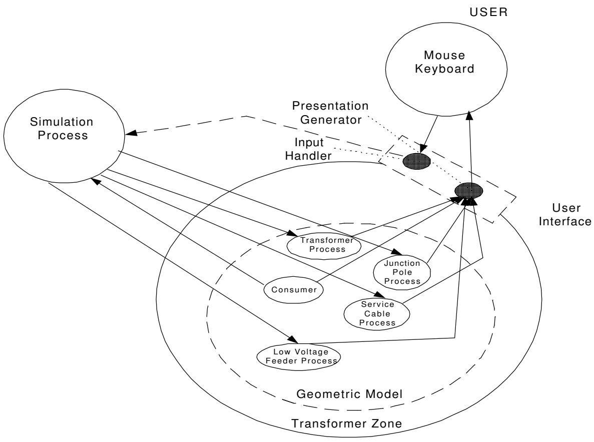
Figure 2

A Virtual Environment is produced from reading in the data as seen in Fig.2. The x-y co-ordinates are mapped onto the x-z plane in the environment and the y co-ordinate of the environment is derived from the heights of the poles as read from the text file. The transformer is positioned at the centre of the environment with all the other object's positions defined relative to the transformer's position. This allows for simpler calculations while navigating (moving from one object to another) within the virtual environment.

In setting up the virtual environment the positions and orientations of the feeders and service conductors have to be calculated. These are calculated by knowing the positions of the source pole and the pole or consumer to which the respective conductor must be attached.

Distributed Virtual Reality Architecture

The DataVis software has been implemented using the Distributed Virtual Reality Architecture or PAM as described earlier. The DataVis program



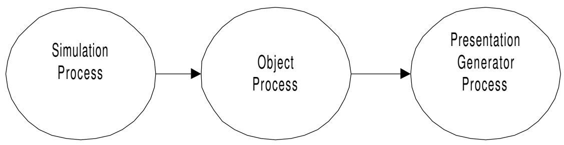
DataVis using the PAM

Figure 3

is run on each of the networked PCs used in the distributed implementation. The software knows which PCs are available and specific processes are allocated to machines. A certain PC may, for example, handle the Presentation Generator, another may handle the Input Handler and several others may have the object processes reside on them. This allocation of processes is invisible to the user.

DataVis makes no use of the User Model object catered for in the architecture since the system presents an identical interface to anybody using the software. The Input Handler will take care of the input from the mouse and keyboard. The architecture has been expanded to provide a collision detection method in the Presentation Generator that checks whether the camera motion and the junction pole motion in the virtual environment is valid. A bounding box is placed around both objects and their intersection is checked. The Presentation Generator will also render the image onto the display. Each object process (service conductor process, junction pole process etc.) contains the relevant data and methods for that object, i.e. the 3D model and the data associated with the particular object, and the method to move to the position of that object. The Simulation Process handles the walking from and to various objects in the environment.

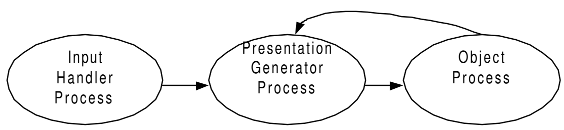
The architecture in its present version allows only for one-way communication between processes. This is illustrated in (Fig.4). For DataVis the architecture had to be expanded to allow for two-way communication, whereby one method is the initiator of the request for information and another method initiates the action upon receiving the information. This is used extensively when DataVis requires information on the object. A typical scenario is shown in Fig.5. The input process receives a left-mouse button double click message that is immediately dispatched to the Presentation Generator which in turns checks to see if an object was picked. If an object was picked, the Presentation Generator knows which object was picked and requests information (initiates a request) about that object from the corresponding object process. The object process will receive the request and send back the desired information to the Presentation Generator which will know the type of data being sent to it. It



One-way communication of processes

Figure 4

will then initiate an action which will in this example display the object's information at the bottom of the window. Another example of this two-way communication occurs when a flyby is requested. This two-way communication continues until the Presentation Generator has received all the x-y-z coordinates of all the junction poles. The Presentation Generator then calculates the path of the flyby and initiates the action.



Two-way communication of processes

Figure 5

DataVis makes use of Criterion Software's RenderWare™ API to render the environment. The API handles the mathematics behind rotating, scaling and translating objects. The API takes care of hidden surface removal, clipping, palette management and shading. The objects in the environment are texture mapped. DataVis takes care of collision detection so that the user cannot go through cables, poles or consumers.

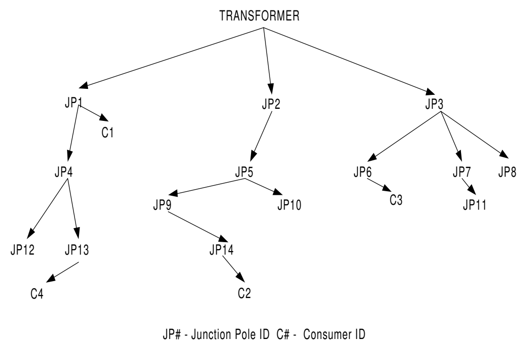
Navigation

DataVis allows the electrical parameters in the distribution network to be visualised by allowing the designer to "move" around the circuit watching how these parameters vary spatially. The physical positions of the junction poles, consumers and service cables can be visualised. The environment is navigated using the desktop mouse device. Pressing the left-mouse button and moving the mouse allows the user to pan the camera. While pressing the right-mouse button and moving the mouse tilts the camera. Positioning the mouse over a junction pole and double clicking the left-mouse button provides the user with information in a status box at the bottom of the screen. This allows for information to be displayed only when the user requests it. Otherwise during the navigation of the environment would result in unnecessary network traffic for information the user did not request. The label of the pole, the number of consumers on each electrical phase "downstream" from that pole, the distance from the transformer, the x-y-z co-ordinates of the pole, and the voltages of each phase at the pole are shown in the status box. Positioning the mouse over a low voltage feeder or service conductor, displays conductor-specific information in the status box.

The navigation tools and manipulation tools that are applicable to any object can be accessed from the menu or by double-clicking the right mouse button. This brings up a pop-up menu, which will position the camera at positions decided by the user. The user can drop onto the terrain and experience the feeling of walking on the ground and viewing the environment from that perspective.

There are three modes of navigation, namely WALK, MANIPULATE and FLYBY. WALK allows the user to transverse the environment either freely or forces a specific route by designating a start and end position. The user can instruct the software to start at a consumer or pole and move to some other pole or consumer. The software renders a path from the source (start) position to the destination (end) position.

Navigation along the network is accomplished by using information the system holds about the connectivity of the various objects in the environment. It is assumed that the network is connected in a radial or Christmas tree-like structure (Fig.6). The transformer is the "angel" on top of the tree, the conductors and feeders are the branches, the junction poles are the nodes and the consumers are the leaves in the tree. The network-tracing algorithm follows a path from the source and destination positions back to the transformer. It then determines whether the destination position is in the source path. If not, it then checks to see if the destination path has any elements in the source path. If so, it appends the destination path from the common element to the source path. If the algorithm has still not determined the total path, it then assumes that the destination position is in another branch of the transformer. It creates a new path with the source and destination path (without the transformer).



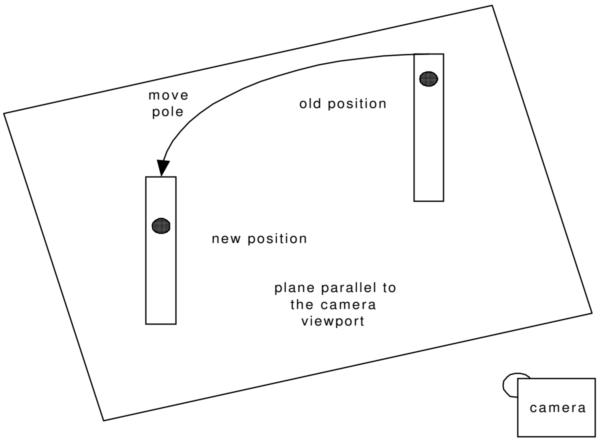
Junction Pole connectivity in radial format

Figure 6

FLYBY gives an aerial flyby of the transformer zone based on the dimensions (furthest object in the x and z plane in both directions and height of the objects) of the transformer zone. MANIPULATE allows the user to alter object parameters. This is explained further in the next section.

Immersive Design

DataVis doesn't allow general manipulation of all of the parameters. This is because DataVis was designed as a specific visualisation tool and not as a 3D Modeller ([Butteworth92] and [Mine97]). Junction poles can be selected and moved to new locations. The closest pole to the camera viewport, as projected onto the camera viewport, will be chosen. The 2D-device space point is to be converted into a 2D-camera viewport space. Because the pole is repositioned using the mouse the only movement allowed is along a plane parallel to the camera viewport (Fig.7). To obtain a precise position requires constrained movements and good perception (there are three dimensions to consider), therefore DataVis provides the designer with the option of moving the junction pole to a



Pole's motion constrained to a plane parallel to the camera viewport

Figure 7

specific co-ordinate in the environment. When the camera's Look-At vector is not perpendicular to the junction pole; the junction pole will be translated in all three dimensions. If the camera is looking directly above the junction pole, motion of the pole will be in the x-z plane, while positioning the camera on the terrain looking at the junction pole will also constrict motion to the x-z plane. Anything in between those viewpoints will result in the junction being moved in all three dimensions. DataVis will then reposition the feeder conductors attached to that pole and service conductors if the pole has consumers connected to it. This consequently alters the phase voltages and currents. The transformer cannot be moved from its position because its position is selected within CART 3.0 to be as close as possible to the centroid of the transformer zone [Dwolatzky-Meyer98].

The environment allows the designer to view the expected voltages on each electrical phase at a consumer or junction pole as a percentage volt drop from the transformer. This is accomplished with 3D bars for each phase, which are positioned above each junction pole and consumer. The height of the bar represents the percentage volt drop from the transformer. Any large or above nominal voltage can be clearly seen by the bar being taller than the others. A red bar is for the red phase, a white bar is for the white phase and a blue bar is for the blue phase.

The CART 3.0 software [Dwolatzky-Meyer98] assumes that all electrical loads on the network are at unity power factor and thus there is no phase angle associated with the voltage. If the phase angle were something other than unity this information could be shown by rotating the bar (phase *degrees*) around the x-axis (Fig.8).

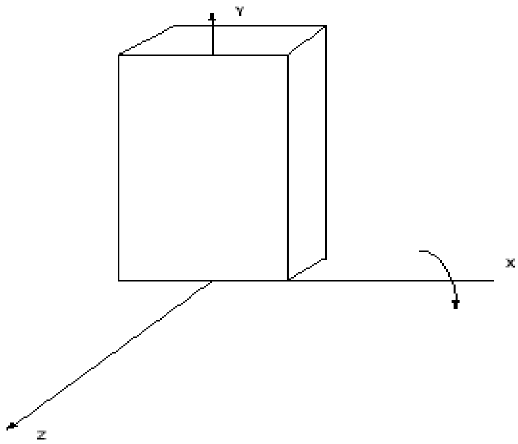


Illustration of rotation of bar due to phase voltage

Figure 8

DataVis allows the designer to see the voltage profile moving away from the transformer along a feeder or towards it using any of the navigation methods. In a conventional 2D representation plotting a graph showing voltage as a function of distance from the transformer would have represented this voltage data.

An important aspect that emerged from designing and implementing DataVis is that the environment should not be cluttered with useless information (objects). Users tend to navigate or steer round objects even if collision detection is not implemented. Therefore the type of objects represented in the environment was restricted to those which exhibit electrical properties. To aid in the viewing, most objects can be switched off from the rendering, thus allowing the designer to view only the poles, or consumers, or transformer or any combination of these.

Output

DataVis outputs a text file that contains the junction pole co-ordinates, feeder cable lengths and service cable lengths. CART 3.0 will read this file and update the 2D CAD drawing with any changes made.

TESTING AND EVALUATION

The DataVis prototype was implemented with two main objectives in mind. Firstly it was used to validate the use of the Distributed Virtual Reality Architecture or PAM and to measure its performance. Secondly, DataVis was developed as a tool to explore the value of a 3D Visualization tool as an aid in the development of electrification schemes.

To achieve the first objective, DataVis has been given a frames-per-second counter. In tests still to be conducted DataVis will be run on a standalone personal computer and then on a varying number of networked PCs with various process allocations. FLYBY and moving to a pole will be used to determine the frames per second on each type of configuration (standalone or networked) and a comparison made.

To achieve the second objective, a panel of experienced electrification network designers will be set up and given CART 3.0 together with DataVis to use. Their reaction to this new tool will be measured by filling out a questionnaire after using the tool. The level of difficulty in positioning the junction poles using the mouse will be recorded. Their perception of lag in using the tool on a standalone workstation and networked workstations will be measured.

The test results of DataVis in this environment will be compared with the findings of [Bowman98], who found with their particular tool that it was more useful for making changes to an existing design than for building a design from scratch.

REFERENCES

Blumenow, W., Spanellis, G., Dwolatzky, B. 1997 *The Process Agent Model and Message Passing in a Distributed Processing VR System*, VRST '97 International Conference, conference proceedings, pp. 165-171.

Bowman, D.A., Wineman, J., Hodges, L.F., Allison, D. 1998 *Designing Animal Habitats within an Immersive VE*, IEEE Computer Graphics and Applications, Vol. 18 No. 5, pp. 9-13.

Butterworth, J. et al.1992 *3DM: A Three-Dimensional Modeler Using a Head-Mounted Display*, Procedures of the Symposium on Interactive 3D Graphics, ACM Press, New York, pp.134-138.

Dwolatzky, B., Meyer, A.S. 1998, *A software based distribution design methodology supporting rural electrification in South Africa*, IEEE Conference on Rural Electricity, St Louis, USA, 27-29 April 1998, pp. B1.1-B1.4.

Mine, M. 1997 *Isaac: A Meta-CAD System for Virtual Environments*, Computer-Aided Design, Vol. 29 No. 8, pp. 547-553.