

## INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

**The quality of this reproduction is dependent upon the quality of the copy submitted.** Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps. Each original is also photographed in one exposure and is included in reduced form at the back of the book.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

# UMI

A Bell & Howell Information Company  
300 North Zeeb Road, Ann Arbor MI 48106-1346 USA  
313/761-4700 800/521-0600





Université d'Ottawa • University of Ottawa



# **Navigation and Visualization Techniques: A Case Study in VRML**

by

**Toufic Milan**

Thesis submitted to the  
School of Graduate Studies and Research  
in partial fulfilment of the requirements for the degree of

**Master of Computer Science**

Under the auspices of the  
Ottawa-Carleton Institute for Computer Science

University of Ottawa  
Ottawa, Ontario, Canada  
May 1998

© Toufic Milan, Ottawa, Canada, 1998



National Library  
of Canada

Acquisitions and  
Bibliographic Services

395 Wellington Street  
Ottawa ON K1A 0N4  
Canada

Bibliothèque nationale  
du Canada

Acquisitions et  
services bibliographiques

395, rue Wellington  
Ottawa ON K1A 0N4  
Canada

*Your file Votre référence*

*Our file Notre référence*

The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

L'auteur conserve la propriété du droit d'auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

0-612-36725-8

## **Abstract**

A sea of data engulfs and sinks today's user in ambiguity and indecisiveness. Few can swim and fewer can effectively surf the Internet, a vast and daily growing store house of data. Almost any desktop computer can instantaneously access this enormous unstructured and mostly redundant information.

The earliest user interface consisted of page after page of unformatted text --- a typical two-dimensional (2D) interface. Using links, Hyper Text Markup Language (HTML) excelled in presenting 2D interface on the World Wide Web (Web), indeed, HTML is still almost all-pervasive on the Web.

Two-dimensional textual interfaces are limiting: navigation is left to right, top to bottom with discontinuity from line to line, an interface tightly linked to the advent of printed material. On the other hand an ideal interface will simulate a human interface with navigation, motion, touch, feel and perception. Virtual Reality Modeling Language (VRML) emerged as a new three-dimensional (3D) standard for representing 3D objects on the Web.

With its latest release, "VRML 2.0" (also known as VRML 97) supports behavior including animation in 3D space triggered by events such as mouse actions. VRML blends space and time with user interface interaction and support for scripting languages.

This thesis addresses navigation and data visualization using VRML both for physical activities (such as visiting a museum) and for representation of statistical data (such as electoral results).

The principal contribution of this thesis is to create 3D interfaces to visualize complex data extracted from live databases on different servers across the Internet. This is realized by means of four experiments which I designed and implemented in VRML. Each experiment covers a different aspect of navigation and visualization.

## Figures

Figure 1 - System Architecture .....	37
Figure 2a - Database View .....	40
Figure 2b - Tables View .....	40
Figure 2c - Fields View .....	41
Figure 2d - Records View .....	41
Figure 3 - Museum Worlds Diagram .....	44
Figure 4- Viewpoint Popup .....	45
Figure 5 - Snapshots of the Five Museum Worlds .....	46
Figure 6 - Accessing Data from Multiple Servers .....	47
Figure 7 - Library World .....	48
Figure 8 - Computer Books View .....	49
Figure 9 - Elections Queries .....	50
Figure 10a - Candidates Table View .....	51
Figure 10b - Parties Table View .....	51
Figure 10c - Ridings Table View .....	52
Figure 10d - Provinces Table View .....	52
Figure 11 - Elections Data Model .....	53
Figure 12 - Elections Unified Table SQL Statement .....	54
Figure 13 - Candidates Unified Table View .....	55
Figure 14 - Ridings Screen .....	56
Figure 15 - Votes Received Screen .....	57
Figure 16 - Ontario's Map Far View .....	58
Figure 17 - Ontario's Map Close View .....	59

- **(Drill-Down World)** The Drill-Down World is a generic VRML front-end interface (consisting of labeled boxes) to access a SQL server on the back-end. It investigates drill-down support for a database by tracking the client state.
- **(Viewpoint World)** The Viewpoint World navigates among different VRML objects (in a photograph gallery) using “viewpoints” through animated navigation sequences.
- **(Multiple Databases/Multiple Interfaces)** This experiment queries a local database to generate VRML objects anchored to external databases on the Web. This embeds the VRML world (local library) in one browser’s frame and the HTML page (remote book seller) in a second frame, thereby combining VRML and HTML in a single interface.
- **(Abstract World)** This Abstract World visualizes statistical information (electoral outcomes) from a back-end database server and displays it through a landscape representation using VRML scene description language.

With a view to developing these worlds I shall review current navigation and visualization techniques. I give special consideration to VRML in context of enterprise computing and database-driven applications.

The best way to view this thesis is on-line at:

<http://www.csi.uottawa.ca/~tmilan/Thesis/ThesisPage.html> to experience the VRML worlds within.

## **Acknowledgements**

I would like to express my deepest gratitude to my supervisor, Prof. Ivan Rival. His interesting discussions and continuous support made my research project most enjoyable.

I would also like to thank my wife Josiane whose emotional support and complete understanding of why I am spending all these long hours (mostly over the weekends) working on my thesis, made my life a little bit easier.

## **Copyright Note**

CosmoPlayer is a registered trademark of Silicon Graphics Inc.

WorldView is a registered trademark of Intervista Software Inc.

Communicator is a registered trademark of Netscape Communication Corporation.

Windows 95, Windows NT, and Internet Explorer are registered trademarks of Microsoft Corporation.

Power Macintosh and Mac OS are registered trademarks of Apple Computer Inc.

All other referenced products are trademarks or registered trademarks of their respective companies.

## Table of Contents

Abstract .....	2
Acknowledgements .....	4
Figures .....	5
Copyright Note .....	6
Table of Contents .....	7
I - Introduction .....	9
A - Goal .....	9
B - Structure of the Thesis .....	11
C - Minimum Requirements to Run the Experiments .....	12
II - Overview of Data Visualization .....	13
A - Definition of Visualization .....	13
B - What Makes a Good Visualization? .....	14
C - Visualization Techniques .....	15
D - Navigation Techniques .....	17
III - VRML Scene Description Language .....	18
A - What is VRML? .....	18
B - Why is VRML becoming the New 3D File Format of the Web? .....	18
C - Using VRML Scene Description Language for 3D Visualization .....	19
IV - An Annotated Bibliography of Current Visualization and Data Navigation Systems .....	21
A - 2D Systems .....	21
B - VRML Systems, Proposals, and Papers .....	23
C - 3D (non-VRML) Systems, Proposals, and Papers .....	29
D - Where is VRML Used Today? .....	35
V - Visualization System Architecture .....	36
A - System Components and Architecture .....	36
1 - Client Side .....	36

2 - Server Side .....	36
B - Database Connectivity .....	37
C - Experiments .....	40
1 - Drill-Down World .....	40
2 - Viewpoint World .....	44
3 - Multiple Databases/Multiple Interfaces .....	47
4 - Abstract World .....	50
VI - Conclusion and Future Work .....	60
A - Conclusion .....	60
B - How can VRML be Improved to Better Handle Databases, Navigation Support and Overall Reliability? .....	63
C - How Can the Experiments be Improved? .....	65
References: .....	68
Appendix A - Glossary of VRML and Data Visualization Terms .....	73
Appendix B - Abstract World Source Code .....	76
Appendix C - Sample of the Generated VRML Source Code .....	94

## **I - Introduction**

### **A - Goal**

Can we visualize the world on a desktop computer? Are there any available tools to help us achieve that?

Let us say I want to create a fully functional virtual city consisting of buildings. Each building represents a functional element of our society. One building is a movie theatre where movies are displayed on demand. Another building is a library where a user can enter and wander around going through racks of books and select a specific book (either directly or through querying a search engine) and read it, gather comments about it, criticism, readers' opinions, number of sold copies, other books written by the author, etc.

Some of these buildings are open without any restrictions and some require a fee and/or access permission. The user is charged in pennies per operation such as for every opened page of a certain book, or for every opened article of a magazine, the library debits the user by 2 cents. This pay-per-use system support fine granularity.

Another building in that virtual world is a doctor's clinic. Accordingly, a user may want to make an appointment and ask questions regarding symptoms. To this end, the world displays the doctor's appointments schedule and allow the user to tentatively reserve a time for a visit.

Next to the doctor's clinic is a car dealership where new and old cars are on display. The user can view the displayed cars, select a specific model, select the color and few special options and view the resulting 3D model and even take the car through a virtual test drive. All statistical information regarding that car is available such as crash test endurance, insurance rates, etc.

There is an art museum next to this car dealership. Visitors can wander through its rooms and hallways to view its objets d'art. A visitor can view an artifact, gather information about it, its artist, related work and make links to other virtual museums with similar and related artifacts.

A ten minute drive (10 seconds in virtual terms) from the museum brings us to a residential construction site displaying home builders' models. The user can visit a model (open its front door, go up the stairs, walk through the corridor, ...). Once the user is interested in a model, she can virtually construct it with some additional options such as: exterior brick color and see how that affect the general look, viewing the house on a specific lot and see how does it fit with the other built houses, get immediate cost calculation based on the model and the added or removed options. The user can select components such as kitchen wood colors, appliances, etc.

The goal of this thesis is to create 3D interfaces to visualize complex data. Four experiments are conducted, each covers a different aspect of navigation and visualization. I will also review some of the existing data navigation and visualization techniques.

Several questions will be asked and answered to some extent including : Is VRML well suited for data navigation and visualization? Does it offer the necessary power? How can VRML be extended and improved?

All the above ideas will require access to back-end database servers. A user navigating through a virtual world can access the latest up-to-date information. This information is retrieved and formatted for VRML display and sent back to the user (client) for viewing. The question will be whether VRML is capable of supporting this needed functionality. Can these complex VRML worlds be downloaded fast enough across the Internet? Is it possible to query data dynamically from a database server and display the result in VRML format? How easy or how difficult will it be for an average user to efficiently navigate through these VRML worlds? Will the user get easily lost within? What will be the minimum requirement (hardware and software wise) to view these VRML worlds efficiently?

## **B - Structure of the Thesis**

Chapter II presents an overview of data visualization and provides several definitions of the term “Visualization”. Data visualization methodologies are introduced. It also describes the terms “Navigation techniques” and “Visualization techniques”.

In Chapter III, I will introduce VRML and provide an exhaustive and thorough description of this scene description language.

In Chapter IV I survey current navigation and visualization techniques and examine several visualization and navigation systems.

Chapter V is the heart of this thesis and this is where I report on my experiments to create visualization systems based on VRML.

Finally, Chapter VI will be the reality check stop point, where, after putting VRML under the microscope, I conclude about improvements to enhance VRML, both on the server and the client sides.

### **C - Minimum Requirements to Run the Experiments**

To properly run the four experiments, you require the following:

PCs	Macintosh
<b>Hardware</b> <ul style="list-style-type: none"><li>• Pentium 75 MHz or better</li><li>• 16MB RAM (32MB recommended) and 14MB hard disk space</li><li>• SVGA/256 colors or better</li></ul> <b>Software</b> <ul style="list-style-type: none"><li>• Windows 95/NT/98</li><li>• Netscape Communicator 4.05 or higher at: <a href="http://home.netscape.com/download/index.html">http://home.netscape.com/download/index.html</a> or Microsoft Internet Explorer 4.0 or higher at: <a href="http://www.microsoft.com/ic/download/">http://www.microsoft.com/ic/download/</a></li><li>• Silicon Graphics' CosmoPlayer 2.0 or higher VRML browser at: <a href="http://cosmosoftware.com/download/player.html">http://cosmosoftware.com/download/player.html</a></li></ul>	<b>Hardware</b> <ul style="list-style-type: none"><li>• Power Macintosh</li><li>• 40 MB DRAM, 20MB hard disk space</li></ul> <b>Software</b> <ul style="list-style-type: none"><li>• Macintosh OS v7.6.1, 8.0, or 8.1</li><li>• Netscape Navigator 4.04 or higher at: <a href="http://home.netscape.com/download/index.html">http://home.netscape.com/download/index.html</a></li><li>• Silicon Graphics' CosmoPlayer 2.0 or higher VRML browser at: <a href="http://cosmosoftware.com:888/cp21setup_Mac.hqx">http://cosmosoftware.com:888/cp21setup_Mac.hqx</a></li></ul>

The experiments can also be run on Silicon Graphics's IRIX 5.3 and 6.2 machines with CosmoPlayer 1.0.

## **II - Overview of Data Visualization**

### **A - Definition of Visualization**

Instead of giving just one definition of visualization, here follow several definitions of visualization as defined by different authors.

<b>Definition</b>	<b>Author</b>
Visualization is the graphical presentation of information, with the goal of providing the viewer with a qualitative understanding of the information contents.	[Ward]
To form a mental vision, image, or picture of something not visible or present to the sight, or of an abstraction; to make it visible to the mind or imagination.	[OXF89]
Scientific visualization is concerned with exploring data and information in such a way as to gain understanding and insight into the data. The goal of scientific visualization is to promote a deeper level of understanding of the data under investigation and to foster new insight into the underlying processes, relying on the humans powerful ability to visualize.	[BRO92]
A useful definition of visualization might be the binding (or mapping) of data to representations that can be perceived. The types of bindings could be visual, auditory, tactile, etc. or a combination of these.	[FOL94]
The primary objective in data visualization is to gain insight into an information space by mapping data onto graphical primitives.	[SEN94]

## **B - What Makes a Good Visualization?**

An important function of visualization is to provide insight into the scientific process through visual methods. The visualization should present complex ideas with clarity, precision, and efficiency. According to Edward Tufte [[Tufte2](#)], a good graphics display should :

- **(Present the Data)** Show the data
- **(Substance over Technology)** Induce the viewer to think about the substance rather than about methodology, graphic design, the technology of graphic production, or something else
- **(Direct Representation)** Avoid distorting the data
- **(Optimize Screen Space)** Present many numbers in a small space
- **(Organize Information)** Make large data sets coherent
- **(Compare/Contrast Information)** Encourage the eye to compare different pieces of data
- **(Drill-Down Support)** Reveal the data at several levels of detail, from a broad overview to the fine structure
- **(Clear Purpose)** Serve a reasonably clear purpose: description, exploration, tabulation, or decoration
- **(Mix Graphics and Text)** Be closely integrated with the statistical and verbal descriptions of a data set

## **C - Visualization Techniques**

In general, visualization is essentially a mapping process from computer representations to perceptual representations, choosing encoding techniques to maximize human understanding and communication. The goal of a viewer might be a deeper understanding of physical phenomena or mathematical concepts.

3D data visualization gives the ability to view multiple data sets in a single screen. On the other hand 2D tables and graphs limit the amount of information that can be viewed and that is due to the lack of the third dimension or z-axis. Data is represented by 3D objects that support different viewpoints.

Visualization is enhanced by interaction with the data sets and getting more in-depth information by drilling down as needed. Visualization success benefits from the display of much information on a single screen.

A user moves objects around and explores spaces interactively.

According to Brodlie et al. [[BRO92](#)], there are three parts to a visualization technique:

- The construction of an empirical model from the data. This construction may involve sampling theory considerations, such as the Nyquist theorem, and general mathematical interpolation schemes. If the data contains errors then this must be taken into account.
- The selection of some schematic means of depicting the model as some abstract visualization object, such as an image of a contour map
- The rendering of the image on a graphics display

The highlights of visualization are:

- **(Data Browsing)** It will compress a lot of data into one picture
- **(Reveal Correlations)** It can reveal correlations between different data sets both in space and time
- **(Selective/Interactive View)** It opens up the possibility to view the data selectively and interactively in 'real time'

## **D - Navigation Techniques**

Navigation became popular with hypermedia systems and the Web. Browsers took advantage of that feature by allowing hyperlinks to be inserted into pages. The ability to navigate made the web look like a sea (or a universe) of information and users wandering around are denoted as doing a surfing activity. That is how the expression “Surfing the Web” became such a popular sentence.

The Web triggered the evolution of a new kind of navigation.

As Keith Instone observed [[Instone](#)], five forms of Web navigation exist.

- **(Browser Navigation)** The Browser Navigation supports back, forward, history, etc.
- **(Content Navigation)** The Content Navigation supports hypertext links, buttons inserted in the page, etc.
- **(Within-page Navigation)** The Within-page Navigation involves scrolling, mainly, but with VRML and applets, this will begin to vary more. Especially interesting is the switch from 2D to 3D with VRML.
- **(Multi-browser Navigation)** The Multi-browser Navigation supports using a couple of browsers simultaneously
- **(Application Navigation)** The Application Navigation involves going from a word processor to a browser, and vice versa. Also applies to spreadsheets that embed hyperlinks at the cell level. Same for databases where a certain field can be denoted as a hyperlink. Almost every new application that exists has incorporated Web navigation within itself.

### **III - VRML Scene Description Language**

#### **A - What is VRML?**

The Virtual Reality Modeling Language (VRML) is a file format for describing interactive 3D objects and worlds. VRML was originally developed in 1994 by Marc Pesce and Toni Parisi, and it quickly became the Internet 3D de facto standard with the strong support of such companies as Netscape, Microsoft and Silicon Graphics.

VRML is designed to be used on the Internet, Intranets, and local client systems. It is a 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. [[VRML97-Intro](#)]

VRML is capable of representing static and animated dynamic 3D and multimedia objects with hyperlinks to other media such as text, sounds, movies, and images.

VRML browsers, as well as authoring tools for the creation of VRML files, are widely available for many different platforms.

#### **B - Why is VRML becoming the New 3D File Format of the Web?**

Unlike some of the other hot technologies on the Web, VRML is an open standard, not owned by any company, and VRML worlds can be created with nothing more than a plain text editor.

VRML came a long way since version 1; the latest version 2 added behavior and scripting support.

VRML can represent the data as predefined graphical objects, using color, motion and shape to convey information. Behavior can be attached to objects and thus animation and drill-down support can be added.

VRML's leading advantage is its platform independence and Web accessibility.

### **C - Using VRML Scene Description Language for 3D Visualization**

The same way HTML became the de facto standard for 2D information representation on the Web, VRML is becoming the de facto standard for 3D information representation. Becoming the 3D Web standard, it is no surprise why VRML is used for 3D visualization.

The world of visual computing is undergoing a major shift in technology. Once confined to high-end research labs with big budgets, the technologies of 3D computer graphics and data visualization are rapidly becoming available to the world at large. In addition, the explosion of the Internet has created an efficient and available infrastructure for the delivery of real-time data and that is where VRML's role will be apparent.

With VRML 2.0, you can design objects that react to user actions, such as doors that open when clicked. Sensors can also be included that respond when the user approaches a certain area--triggering an alarm, for instance, or starting an animation. As summarized by [[Hartman](#)], VRML 2.0's most noticeable features are :

- Movies and 3D sounds
- Visual effects such as fog and scenic panoramas with mountains, plains, or collision detection, which prevents users from walking through walls
- Sensors that keep track of the passage of time and respond to user actions such as clicking an object or moving to a certain location
- Interpolators, which make it easy to include key-frame animation in your worlds
- The Script node, which allows writing of mini-programs in a language such as JavaScript or Java, is used to build logic into a VRML world

- VRML increases the speed of visualization based on using several representations for one object. VRML contains a specific node called level of detail (LOD). The level of detail defines how the object appears on display with respect of viewing distance. It means that the rendering software substitutes one model with less or more details as user goes through the scene. Furthermore, objects can be defined as invisible from a certain distance.
  
- A new Billboard node embeds VRML objects. Objects within (mostly text objects) are confined to always face the active view. So an object can have a text label that will always rotate to face the current view when the user rotates the world.

## **IV - An Annotated Bibliography of Current Visualization and Data**

### **Navigation Systems**

#### **A - 2D Systems**

##### **[Kaptelinin] A Comparison of Four Navigation Techniques in a 2D Browsing Task:**

Four window navigation techniques were compared in this experiment: scroll bars, dragging, and two modifications of a "bird's eye view." It was found that standard scroll bar navigation was associated with the slowest performance and was rated by the subjects as least preferable. The best performance and the highest preference ranks were observed under the "bird's eye view" conditions.

##### **[Furnas] Space-Scale Diagrams: Understanding Multiscale Interfaces:**

Introduced space-scale diagrams as a new technique for understanding multiscale interfaces. Their defining characteristic and principal virtue is that they represent scale explicitly. It showed how they can aid the analysis of pans and zooms because they take a temporal structure and turn it into a static one: a sequence of views becomes a curve in scale-space. This has already helped in the design of good pan/zoom trajectories for Pad++ [Benderson]. This paper also showed how the diagrams can help visualization of semantic zooming by showing an object in all its scale-dependent versions simultaneously. This will be used as an interface for designing semantically zoomable objects. It also suggested that diagrams may be useful for examining other non-flat multiscale representation, such as fisheye views.

##### **[Johnson] A Comparison of User Interfaces for Panning on a Touch-Controlled Display:**

An experiment was conducted to determine which of several candidate user interfaces for panning is most usable and intuitive: panning by pushing the background, panning by pushing the view/window, and panning by touching the side of the display screen. Twelve subjects

participated in the experiment, which consisted of three parts: 1) subjects were asked to suggest panning user interfaces that seemed natural to them, 2) subjects each used three different panning user interfaces to perform a structured panning task, with experimenters recording their performance, and 3) subjects were asked which of the three panning methods they preferred.

One panning method, panning by pushing the background, emerged as superior in performance and user preference, and slightly better in intuitiveness than panning by touching the side of the screen. Panning by pushing the view/window fared poorly relative to the others on all measures.

Push background panning was the clear winner of the accuracy test: it resulted in faster times, fewer corrective movements, and fewer initial direction-errors than the other two panning methods tested.

### **[Lokuge] GeoSpace: An Interactive Visualization System for Exploring Complex Information Spaces:**

This paper presented a reactive interface display for interactively exploring complex information spaces. The knowledge representation scheme using presentation plans and information seeking goals, combined with the activation spreading network, provides the information display with a reactive capability.

The mechanism can implicitly chain presentation plans by hierarchically spreading activation energy, and can respond to an immediate shift of interest by spreading negative energy to conflicting plans. The system can also direct a user's attention in a fluid manner without losing overall context, by gradually changing the states of activation. Dynamic use of various visual techniques, such as translucency, type size and color, are directly associated with activation levels of plans and visually clarify the display. A learning mechanism was also presented as an integral part of the system, which allows users to customize the information display. These features make a user's exploration of complex information spaces a more dynamic experience.

**Other systems include:**

**[Twidale] Collaborative browsing and visualisation of the search process**

**[Donskoy] Window navigation with and without animation: a comparison of scroll bars, zoom, and fisheye view.**

**[Lee] A Dynamic, Schema-Independent Web Interface for a Relational Database**

**[Schartz] Information Retrieval in Digital Libraries: Bringing Search to the Net**

### **B - VRML Systems, Proposals, and Papers**

**[Munzner] Visualizing the Structure of the World Wide Web in 3D Hyperbolic Space:**

Traditional two-dimensional Web browsers provide a way to focus on an individual document.

Moving up to three dimensions will allow viewing both multiple documents and the links among them. The Web is far too large to see all at once, but it can be explored in sections and by building 3D graphical representations which can be viewed in a 3D Web browser.

This paper implemented a system for visualizing information hierarchies, specifically directed graphs with cycles, in 3D hyperbolic space. This system was used to construct graphical representations of the structure of the World-Wide Web and Unix filesystems. It offers a seamless framework for both a visual gestalt of the system and closeup detail at any level of the hierarchy.

**[Moore] Using Java to interact with geo-referenced VRML within a Virtual Field Course:**

The ability to model, access and control spatial data in VRML provides the first link in developing geo-referenced virtual worlds and promises a powerful tool for spatial scientists.

VRML using Java functionality is freely available and distributable and has strong user community support.

Using Java as the driving language and VRML as the output scene description language will result in the creation of more robust Internet applications. Through Java, a VRML scene can be manipulated in two different ways: 1) Through the internal Java node or 2) through the External Authoring Interface (EAI). The authors complain that by using EAI, an application has to decide on which VRML plugin browser to be used since each behaves differently and certain application that runs on a certain browser might fail to run on another browser.

**[Shilman] Moby: Hierarchical Information Visualization:**

Moby is a software application for rapidly navigating and searching hierarchical structures. This software incorporates a variety of popular techniques from the field of information visualization including a distortion-based viewing technique for focus/context navigation, an extensible filter mechanism, level-of-detail for information culling, and several ways of organizing the information on the display.

The user interface was evaluated using qualitative measurements, explored the use of non-traditional input devices for hierarchy navigation.

**[Ressler] Using VRML to Access Manufacturing Data:**

The system consists of VRML objects that can act as an interface to the database. Queries are effectively hard coded as URL requests. The data associated is stored in a relational database.

One of the key advantages of the Visual Interface to Manufacturing system (VIM) is that it is primarily intended to enable the user to query for data through the visualization. The visualization is the interface to data.

There are two ways to use the VRML models. One for querying the database called "navigation"

and the second for a type of visualization called "manipulation."

Although focusing on the VRML interaction Ressler's prototype served also as a systems integration project.

**[Coors] Using VRML as an Interface to the 3D Data Warehouse:**

As VRML does not provide support for interaction and on-line access to databases, even though these are two fundamental issues for business applications, a system called GOOVI-3D was created that provides access and interaction with a 3-D spatial database over the WWW. GOOVI-3D is implemented in Java and VRML using CORBA for accessing the data warehouse. Building on the experience with GOOVI-3D, the authors propose two lightweight extensions of VRML that can make the development of business applications much easier: An integrated name dictionary management and a SQL node.

**[VRML-DBGroup] VRML Database Working Group:**

The mission of this working group is to define a set of standard interfaces for implementing a variety of database functionalities within VRML. VRML developers will be able to leverage the benefits of databases without struggling with low-level issues of database and network integration. This technology will open up significant commercial business opportunities for VRML.

Their proposal defines two distinct, complementary interfaces for enabling database access from VRML. Both interfaces are fully compliant with the existing VRML specification, and introduce no incompatibilities or new language features.

The two interfaces address two components of the database extensions "Request For Proposal".

**Embedded SQL:** SQL scripting provides a mechanism for executing arbitrary SQL statements within a VRML application.

**Server Side Includes:** The server redirect node provides a mechanism for embedding data-driven components within a VRML world delivered from a server.

**[Oracle VRML Cartridge] Oracle VRML Cartridge White Paper:**

The focus of the VRML development community has been on client-side technologies for efficient modeling and deployment of 3D. The techniques and technologies of client/server and distributed application environment are not yet available to the VRML author. This lack of access to server technologies manifests itself in a variety of ways:

- Limitations on the size and robustness of a virtual world
- No high-level constructs for communicating with back-end services
- The inability to decouple application content from application logic

Oracle's solution was to introduce the VRML cartridge - a complete platform based on Network Computing Architecture for building and deploying business applications in VRML. It delivers a three-tier architecture, combining database servers, application servers, and thin clients to enable VRML applications for the enterprise.

- The database server is a VRML Data Repository that manages persistent, scalable, and secure VRML worlds
- The application server is a VRML Logic Repository that decouples VRML scripting and event generation from the client
- The thin client is a set of standard VRML PROTOs and Java cartridges, designed to work with any VRML 2.0 compliant browser. It is used to communicate with Logic and Data Repositories.

**[Siebert] Parable: A Multi-dimensional, VRML Based, Financial Visualization**

**Application:**

A system was developed by which an investor can skim through a portfolio of securities and, at a glance, assimilate meaningful information from the different data contained in an account.

Special techniques have been investigated and developed to help the user gain more insight from dynamic, multi-dimensional information presented in the three spatial dimensions, most notably the use of 3D symbolic glyphs (virtual objects).

**[Pape] Pape's Topographic VRML 2.0 Map Generator:**

This system allows to interactively generate VRML 2.0 scenes of topographic maps of the world. The surface can be colored based on elevation, or texture-mapped with various other data sets.

**[Tisny] US VRML Statistical Map:**

This is an example of US Statistical Map that generates VRML output file. It allows different states selection, selecting various visual parameters such as colormap and background color and finally selecting statistics and pressing "Generate" button will create a VRML US Map. Moving cursor over each state activates the information available and pressing button on the state will bring the HTML page with detailed, tabulated information.

**[Planet9]** A commercial site that specializes in creating virtual cities. The models are faithful reproductions of actual buildings. New Orleans and San Francisco are two among many cities that are now present on-line in 3D. You can fly over the city and visit your favorite street or square (such as Union Square in San Francisco). This site shows most of VRML features such as bitmap textures, animation and sound. The worlds were designed using Autocad and 3D-Studio and exported to VRML.

The site contains 3D models for 20 different cities.

**[Milan]** For a complete bibliography visit my home page at: <http://www.csi.uottawa.ca/~tmilan>

**Other VRML systems include:**

**[Cave] CAVE Virtual Reality System**

**[Povary] Forest Stand Data Visualization with POVRAY.**

**[Daessler] Visualization of Abstract Information**

**[Leigh] Issues in the Design of a Flexible Distributed Architecture for Supporting Persistence and Interoperability in Collaborative Virtual Environments**

### **C - 3D (non-VRML) Systems, Proposals, and Papers**

[[Tuftel](#)] "The world is complex, dynamic, multidimensional; the paper is static, flat. How are we to represent the rich visual world of experience and measurement on mere flat land? "

3D can convey more information than 2D but it has its own limitation as well. 3D viewing encounters a fundamental problem in the display of information in that it is possible for an object of interest to be partially or wholly occluded by other objects. Current solutions provide access to such internal details through the use of cutting planes, layer removal, fly-through, and transparency. In this thesis special emphasis will be accorded to fly-through and viewpoint change.

[[Darken93](#)] **A Toolset for Navigation in Virtual Environments:**

This paper investigates design principles for aids to navigation in virtual environments. It considered how humans and birds use environmental cues to aid navigation in the real world. It also looked at the principles of cognitive map formation and map design and understanding developed by cartographers and planners. Based on this background a system consisting of a set of tools for navigation in a very simple virtual environment was created. An informal empirical study of the tools for a small set of searching tasks supports the following general conclusions:

- People tend to take advantage of environmental cues in predictable ways. They use them to partition spaces as an aid to exhaustive search. They use them to maintain direction relations performing best when the cue is statically positioned or highly predictable in its motion and when it is visible from the entire environment.
- Cues in different modalities, e.g. visual and audible can be combined to make targets easier to find

**[Darken96] Wayfinding Strategies and Behaviors in Large Virtual Worlds:**

An experiment was conducted to determine whether people use physical world wayfinding strategies in large virtual worlds. The study measures subject performance on a complex searching task in a number of virtual worlds with differing environmental cues.

The methods employed in each subtask were profoundly affected by the stimuli presented. The lack of directional cues and spatial organization in the control treatment led to ineffective search strategies and frequent disorientation. The radial grid provided enough information to successfully execute a search but required reinforcing actions to maintain orientation. The map provides a simultaneous geocentric perspective augmenting the egocentric perspective and fostering the use of geographical landmarks and optimizations to search methods. General conclusions drawn from this work include:

- When not given an adequate source of directional cues, disorientation will inhibit both wayfinding performance and spatial knowledge acquisition
- A large world with no explicit structure is difficult, if not impossible, to search exhaustively. This was shown by repeated reacquisition behavior in the control treatment.
- A conceptual coordinate system is often imposed on the world to act as a divider. This is a side-effect of not being able to divide the world explicitly. A structure must be imposed on the world if an organized exhaustive search is to be attempted.
- The observations made support the notion that path following is a natural spatial behavior. Subjects frequently used features such as coastlines or grid lines as if they were paths.
- A map allows for optimizations to be made to search strategies. This is because it

can be considered a supplement to survey knowledge.

- Dead reckoning was observed to be an intuitive and natural part of navigation; all subjects exhibited this behavior even though frequently unaware of it. The ability to infer position from a past location and constant velocity over time, while sometimes complex in reality, appears to be more easily understood and implemented in virtual spaces.

**[Cowperthwaite] Visual Access for 3D Data:**

Unlike 2D techniques, 3D viewing encounters a fundamental problem in the display of information in that it is possible for an object of interest to be partially or wholly occluded by other objects.

Current solutions provide access to such internal details through the use of cutting planes, layer removal, fly-through, and transparency. However, such techniques result in the loss of contextual information.

This paper goal was to allow interactive access to 3D information spaces while maintaining context. This is achieved by providing a viewer aligned visual access distortion which clears a line of sight to the object of interest, permitting examination from all angles.

**[Bowman] THE VIRTUAL VENUE PROJECT: Advanced User Interface Techniques for Information-Rich Virtual Environments:**

The Virtual Venue project is a testbed for research into many aspects of user interfaces for immersive virtual environments. These include:

1. **Navigation of the environment:** Provides a flying chair metaphor for travel within the space. The user is seated in a physical chair, which is also represented visually in the virtual world. When the user selects a destination from a pre-defined list (much like

viewpoints) the chair "flies" to that location using a smooth animation.

2. **Issuing commands:** Users can travel to new locations, retrieve text documents, set controls, start animated sequences, and so on. All actions can be performed using two simple interaction techniques - one direct and one indirect.
  - **The direct technique is ray-casting:** The user holds a stylus in one hand, and presses the stylus button to cause a light ray to emanate from the stylus. The user directly picks objects in the environment by highlighting them with the light ray.
  - **The indirect technique:** A menu system using physical props was implemented. The user holds a physical tablet on which the virtual menu system appears. By touching the tablet with the stylus, the user can highlight and select menu items.
3. **Information embedding and access:** There are many different levels at which information is embedded within the virtual venue such as 1) audio, 2) text information which is accessed by choosing a menu command, 3) extended the hyperlink metaphor into the virtual environment, 4) 2D image data, 5) 3-D information, and 6) Embedded animated 3D information
4. **Retrieval of audio annotations:** Developed tools to allow programmers to define audio annotations of various types and methods for playback. Specifically, audio playback may be triggered in three ways. 1) a sound may be played when the user selects an object. 2) the user may hear a sound when he/she travels to a specific location and playback will stop if the user leaves that location while the sound is still being played. 3) an object enters a pre-defined location in the environment.

**[Hibbard] Vis5D a system for interactive visualization of large 5-D gridded data sets:**

Vis5D is a system for interactive visualization of large 5-D gridded data sets such as those produced by numerical weather models. One can make isosurfaces, contour line slices, colored

slices, volume renderings, etc. of data in a 3-D grid, then rotate and animate the images in real time. There is also a feature for wind trajectory tracing, a way to make text annotations for publications, support for interactive data analysis, etc.

**[SAGE] SAGE (System for Automated Graphics and Explanation):**

Stands for System for Automated Graphics and Explanation. It is a mixed-initiative presentation system that supports visualization creation. Inputs are a characterization of the information to be visualized and a user's information viewing goals. Design operations include selecting techniques based on expressiveness and effectiveness criteria, and composing and laying out graphics appropriate to information and goals. SAGE enhances user-directed design of visualizations by giving users the choice between having the system:

- Create the novel displays that users specify
- Complete partial specifications
- Retrieve previously created graphics based on their appearance and/or their data content, and design graphics completely autonomously when users request them

**[Chuah] SDM: Selective Dynamic Manipulation of Visualizations:**

This paper presented a set of interactive techniques for 2D and 3D visualizations. This set of techniques is called SDM (Selective Dynamic Manipulation). Selective, indicating the goal for providing a high degree of user control in selecting an object set, in selecting interactive techniques and the properties they affect, and in the degree to which a user action affects the visualization. Dynamic, indicating that the interactions all occur in real-time and that interactive animation is used to provide better contextual information to users in response to an action or operation. Manipulation, indicating the types of provided interactions, where users can directly move objects and transform their appearance to perform different tasks.

These techniques enable users to:

- View selected objects in detail while keeping them in context with the rest of the environment
- View occluded objects by elevating them or by reducing the widths and heights of surrounding objects
- View different object sets in different scales
- Add new information to the visualization that is not part of the underlying data
- Compare the widths and heights of objects even when they are positioned far from each other

**[Stoakley] Virtual Reality on a WIM: Interactive Worlds in Miniature:**

The WIM interface gives the user of an immersive three-dimensional environment a chance to operate at several different scales through several different views at once without engaging explicit modes. Some of the most important features of this technique are:

- Intuitive use of the WIM metaphor
- Multiple, simultaneous points of view
- Multiple scales for selecting and manipulating objects
- Easy manipulation and selection of objects at a distance
- Improved visual context
- Removal of occlusion and line of sight problems
- General applicability across a wide range of application domains

**Other systems include:**

**[Dieberger] Navigation in Textual Virtual Environments using a City Metaphor**

**[OLAPS] 3-D Visualization Research at National Weather Service Forecast Office**

**[Hix] Pre-Screen Projection: From Concept to Testing of a New Interaction Technique**

**[Disz] Sharing Visualization Experiences among Remote Virtual Environments**

### **D - Where is VRML Used Today?**

- **(Entertainment)** [Dilbert] animated cartoon at:  
<http://umweb1.unitedmedia.com/comics/dilbert/vrml/>, Floop animated cartoon at:  
<http://cosmosoftware.com/galleries/floops/main.html>
- **(Tourism)** [Planet9] representation of 20 North American cities at:  
<http://www.planet9.com/indexnetscape.htm>
- **(Business Decision Making)** Visual Decisions Inc.[VDI] at:  
<http://www.vdi.com/products/gallery/index.htm>
- **(Scientific Representations)** Atoms viewer at:  
<http://ws05.pc.chemie.th-darmstadt.de/vrml/vib/>
- **(Medical)** Skeleton model at:  
<http://www.qin.com/virtual.gallery/vrml/skeleTt/skeleTt.html> , Biological Macromolecules at: <http://www.imb-jena.de/VRML.html>
- **(Educational)** Student Genome Project at:  
[http://www.cat.nyu.edu/sgp/vrml/index\\_main.html](http://www.cat.nyu.edu/sgp/vrml/index_main.html)
- **(Space Exploration)** NASA at: <http://liftoff.msfc.nasa.gov/vr/vr.html> , Silicon Graphic's Mars Exploration Mission at: <http://cosmosoftware.com/galleries/mars/>
- **(Demos)** Product Demo at: <http://www.intervista.com/vrml/gallery/product.html>

## **V - Visualization System Architecture**

### **A - System Components and Architecture**

These are the main components of the visualization system [Figure 1].

#### **1 - Client Side**

- **Web Browser:** Netscape Communicator or Microsoft's Internet Explorer
- **VRML Plugin:** Silicon Graphics' Cosmo Player or Intervista's WorldView. These browsers are widely available and Cosmo Player ships with the Professional version of Communicator 4.0 of which 20 million copies are now downloaded. A VRML Plugin will also be pre-equipped and pre-installed on Windows 98 making it even more accessible and easier to use.

Through the Web browser, the client can access the server side to query the data on the server. The resulting query is returned as a VRML world that the client displays.

#### **2 - Server Side**

- **Web Server:** Through CGI scripts, the client can pass commands to the server to query the data
- **SQL Database Server:** The system interfaces to the data through the MiniSQL database server [Hughes Technologies], where tables are stored. MiniSQL is a well known shareware database server powerful enough to handle the visualization system.

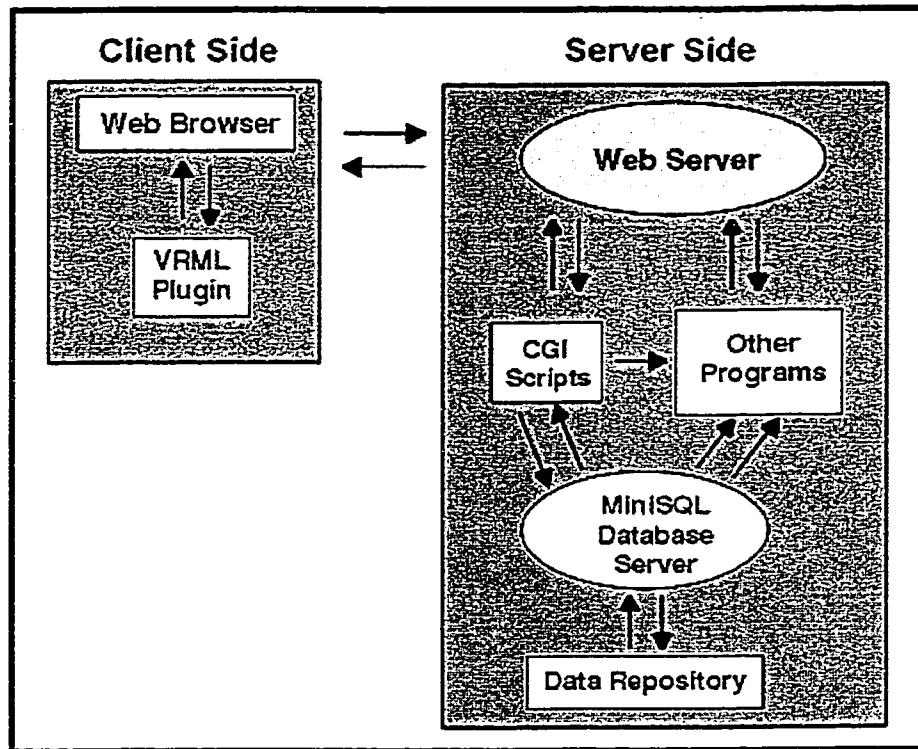


Figure 1 - System Architecture

### **B - Database Connectivity**

VRML has no built-in support to access databases. Many companies and individuals are trying to address this deficiency. Oracle and others propose to add SQL database access to VRML.

See <http://www.vrml.org/WorkingGroups/dbwork/dbex.html> for more details.

For a variation see <http://ece.uwaterloo.ca:80/vrml98/papers/content.html#paprlistDB3>

Since accessing a live database is an essential requirement for my research on VRML, I faced the same limitations and I opted to find the simplest solution. I use the same logic as the Internet search engines (such as Yahoo, Altavista, Lycos and HotBot) through CGI scripts. CGI is an abbreviation for Common Gateway Interface and through it a client can run a script on the server. The CGI script will process the user's search string and will return the search results as an

HTML document. The returned document can contain hyperlinks which can have more embedded calls to additional CGI scripts.

Let us say for example that you are using Lycos and your search string is “VRML Research”. The search engine will process your search string and will display the first 10 matches. You can get additional results by clicking on “Next Page” hyperlink. Have you wondered how the search engine keeps track of where you are among the search result pages? Well the answer is simple enough and that is the “Next Page” hyperlink is another call to a CGI script where the same search string is passed along and the script has an additional variable to tell it which search result to start at. So let us say your first screen has 10 search results and by clicking “Next Page” search results 11 to 20 will be displayed. Take a closer look at this embedded CGI script call:

<http://www.lycos.com/cgi-bin/pursuit?first=11&query=VRML+Research&matchmode=and>

This CGI script “pursuit” is called along with three parameters of which one is the original search string and another is the starting search result entry. Knowing that the first page displayed search results 1 to 10, it is no surprise that the “Next Page” hyperlink informs the search engine to skip the first 10 records.

The search engine server does not need to keep track of every user search. The generated HTML file, which is sent back to the client, has within it all the necessary scripts’ arguments to keep going. So when the first 10 search results are displayed, the script knows in advance how many matching results are displayed and then the “Next Button” script call will have first argument =11.

This method is efficient as the search engine can serve multiple requests by running multiple processes without any need to store additional data.

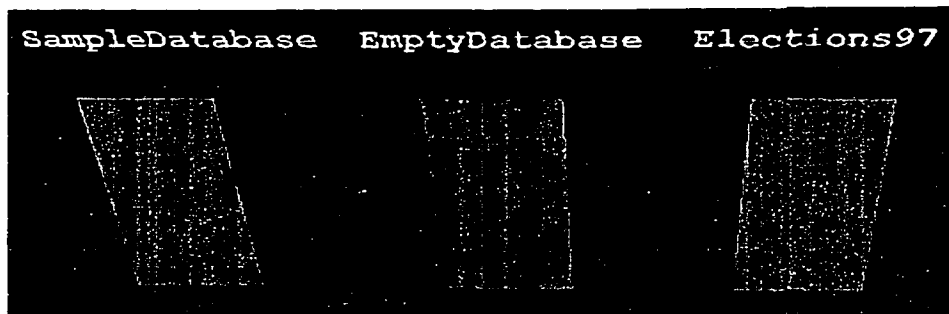
**How does that relate to VRML?**

Fortunately, VRML has support for Anchor nodes which are simply hyperlinks. As hyperlinks, Anchors can have embedded CGI script calls with all the necessary variables and their respective values.

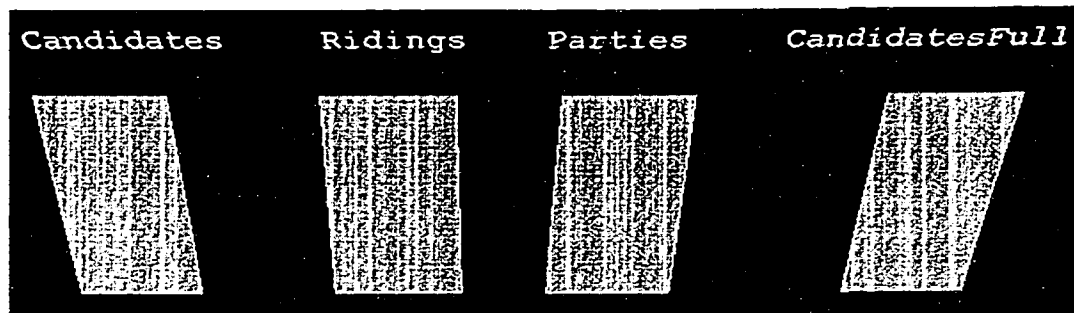
## **C- Experiments**

### **1 - Drill-Down World**

**Mission Statement:** Create a simple VRML front-end interface to access a SQL server on the back-end. The system should allow drill-down by keeping track of the client state. See [Figure 2] for snapshot of different screen captures or you can directly go through the VRML world by visiting this URL: <http://grdb.csi.uottawa.ca:1050/cgi-bin/dbFrontEnd?Type=Database>



**Figure 2a - Database View**

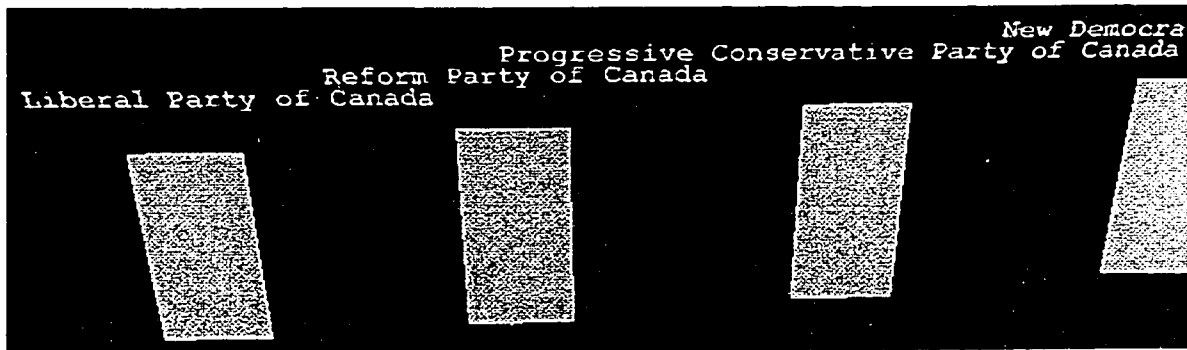


**Figure 2b - Tables View (as a result of selecting Elections97 Database from figure 2a)**



A screenshot of a VRML interface showing a table with two columns: 'PartyNo' and 'PartyName'. The table is currently empty, with only the column headers visible.





Figure 2c - Fields View (as a result of selecting Parties table from figure 2b)



A screenshot of a VRML interface showing a table with one column: 'PartyName'. The table contains four records, each represented by a textured rectangular block. The party names are listed above the blocks: 'Liberal Party of Canada', 'Reform Party of Canada', 'Progressive Conservative Party of Canada', and 'New Democra'.

Figure 2d - Records View (as a result of selecting PartyName field from figure 2c)

Let us describe in detail the above example and show how all the system components will interact together to give us the results that are shown in [Figure 2].

Client Side		Server Side		
<p>Embedded in the original HTML page is a call to dbFrontEnd CGI script having Type = Database</p> <p>The user can launch the CGI script by clicking on that link.</p>		<p>Web server traps the call and executes the script that will query the MiniSQL database.</p>		<p><b>MiniSQL Engine</b></p> <p>will process the SQL query by listing all its available databases and sending a list back to the CGI script.</p>
<p>The Constructed VRML world is sent back to the browser who will display it within its VRML plugin. [Figure 2a]</p>		<p>For every received record, the CGI script will create a VRML object that has a hyperlink embedded within. So in this instance the next logical step will be to select a table so Type=Table and Database=DatabaseName as returned by the SQL query.</p>		
<p>Once the generated world is sent back to the browser, clicking on any of the VRML objects will end up calling another CGI script and the steps listed above will be in effect once more with different script arguments. That will subsequently result in three additional steps as shown in [Figure 2b], [Figure 2c] and [Figure 2d]</p> <p>Through the received VRML world, the client keeps track of its state for subsequent drill-downs.</p>				

In this example, the user first selects a database. The CGI script displays all active databases as boxes with the name of the database floating above. After processing this information, the server sends back the generated VRML world back to the client. The Anchor value of the created objects keeps track of the current database selected. This allows drill-down while keeping the context of the previous selection.

So once the user select one of the database VRML objects, the CGI script will replace the VRML world with a new world consisting of Table objects. Selecting a Table object will generate a list of fields objects and selecting any of these fields objects will finally generate record objects.

Experiment 1 CGI script is: <http://grdb.csi.uottawa.ca:1050/cgi-bin/dbFrontEnd>

This script can accept these parameters.

- 1- "Type" can be either "Database", "Table" or "Field"
- 2- "Database" is the name of the database project folder
- 3- "Table" is the name of the Table
- 4- "Field" is the name of the field

File size for the generated VRML world is: 4 K

When drilling down and reaching the field level, every generated VRML object will be around 1K.

## **2 - Viewpoint World**

**Mission Statement: Navigation between objects using Viewpoints.**

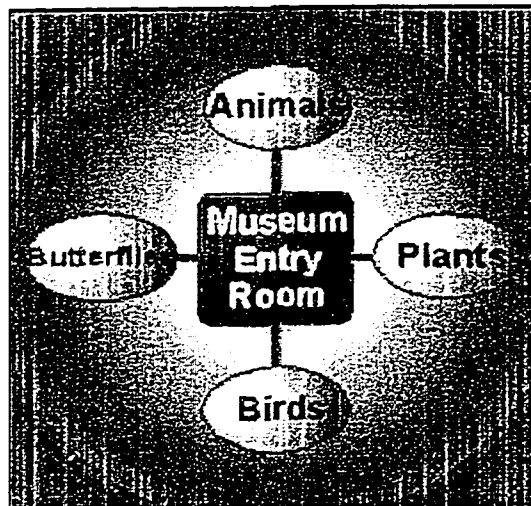
The system should allow animated navigation sequences when moving between different objects. Also drill-down support is needed to load a new world when selecting an object.

Viewpoints are important to help the user navigate through a large complex world and to prevent submersion and avoid loss of direction.

The system consists of five rooms where one of these rooms is the entrance to the other four.

See [Figure 3] for a diagram showing the five different worlds. [Figure 5] shows a snapshot of different screen captures.

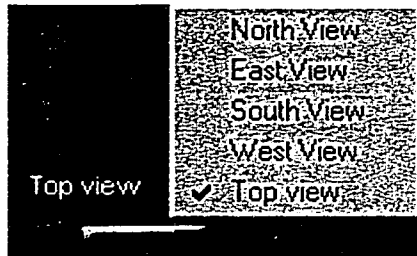
To view the VRML world visit <http://grdb.csi.uottawa.ca:1050/cgi-bin/museum?world=Museum>



**Figure 3 - Museum Worlds Diagram**

The main room has four types of objects each representing a world. The object types are:

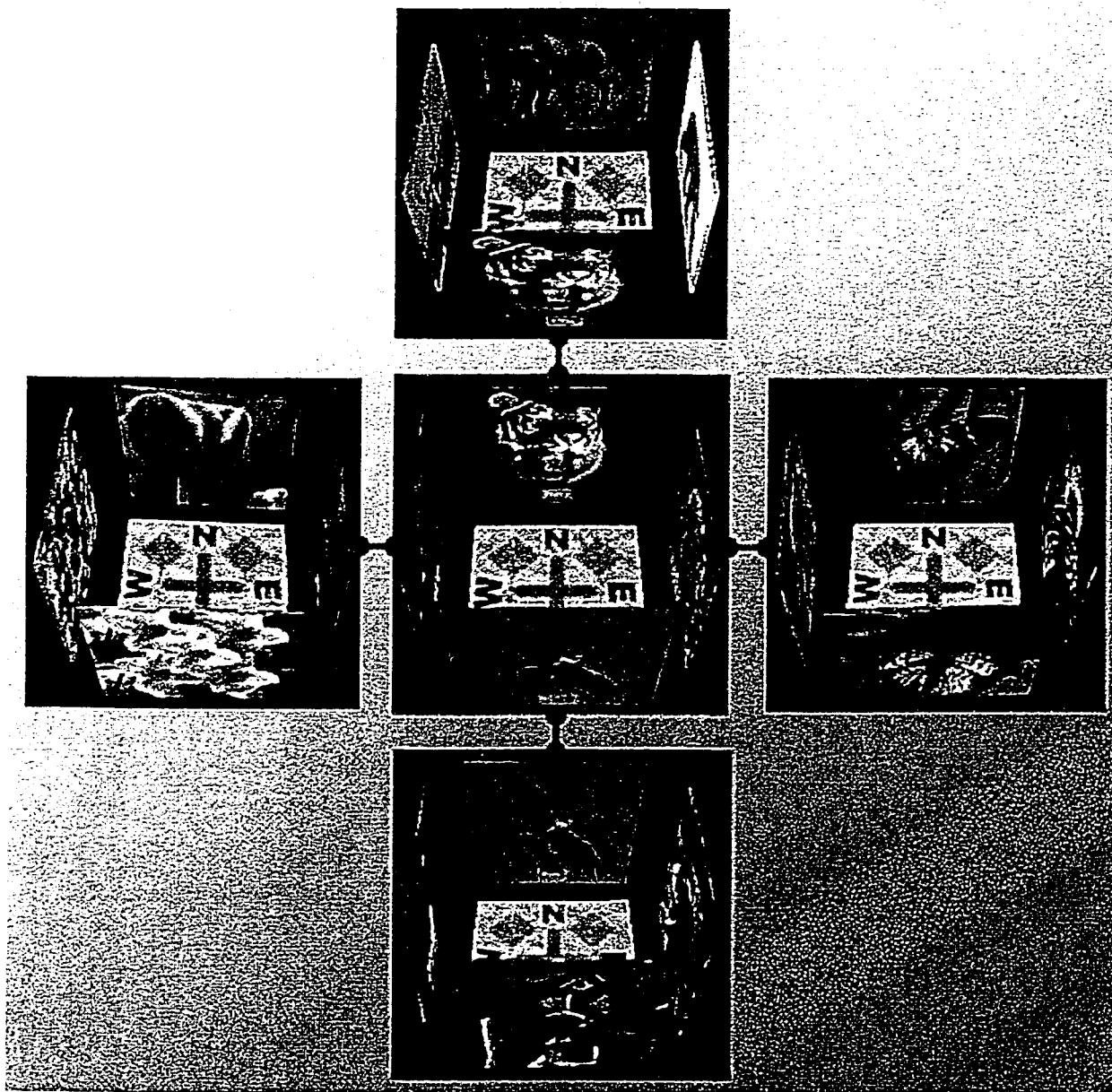
animals, plants, birds and butterflies.



Within each world, five viewpoints are presented: North View (default view), East View, South View, West View and Top View. Each of these viewpoints moves the active view to a special object. Every object in the scene has a viewpoint

associated with it so that manual navigation is not necessary.

The picture below [Figure 5] represents a snapshot of all five worlds. The red links between worlds represent gateways through which a new world can be loaded. These gateways can be activated by simply clicking on a specific object.

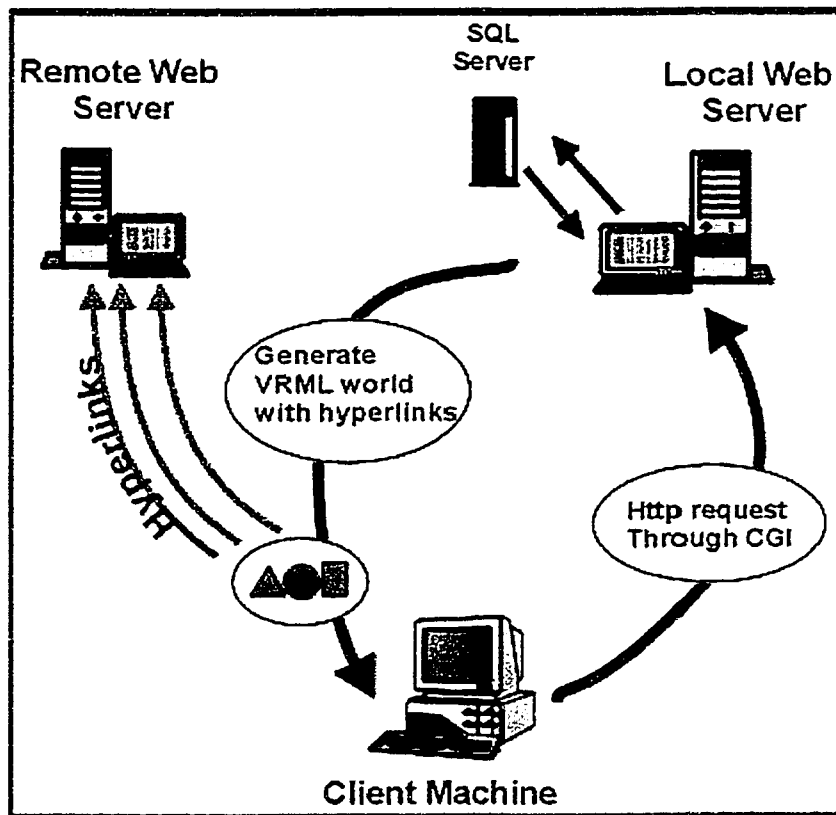


**Figure 5 - Snapshots of the Five Museum Worlds**

### **3 - Multiple Databases/Multiple Interfaces**

**Mission Statement:** Combining VRML world and textual content by embedding the VRML world in a frame and displaying relevant textual description in another HTML frame. All this is done while querying a back-end SQL server and generating VRML objects that can reference other databases on the net.

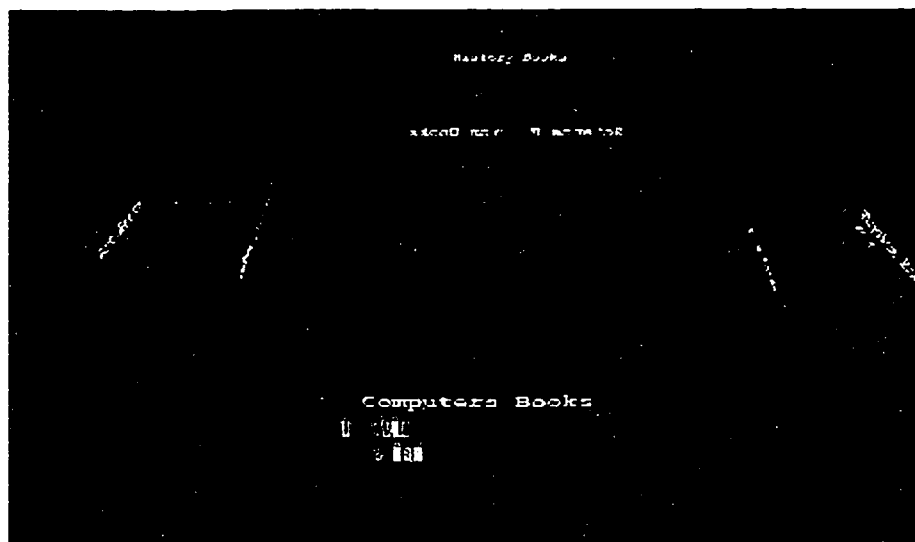
[Figure 6] illustrates a configuration in which a local database query result can be converted to a VRML world and in which each object within that world can be anchored to an external database.



**Figure 6 - Accessing Data from Multiple Servers**

VRML supports text nodes that are represented as 3D objects. The text node is only useful for viewing text and inputting text is not possible. To work around that and leverage the usefulness of HTML, both VRML and HTML can coexist, each residing in a different frame. One frame is used to display the 3D scene and another frame is used to display the textual information in HTML format.

The library example is a case where 3D objects (mostly books and shelves) are laid out in a 3D scene, grouped by topic category [Figure 7].



**Figure 7 - Library World**

The user can navigate through these worlds either by free navigation or by viewpoints. Each viewpoint represents a category of books and, by selecting a viewpoint, the user is transported to that library section through a smooth animation. Selecting a book will fetch its data from [Amazon] and display it in an HTML frame window.

So if the user selected “Computer Books” as a viewpoint, the VRML portion of the screen will

display books related to that topic. For example, clicking on the “Late Night VRML 2 with Java” book displays relevant information in the left portion of the screen [Figure 8].

To view the VRML world visit: <http://www.csi.uottawa.ca/~tmilan/Thesis/LibraryMain.html>

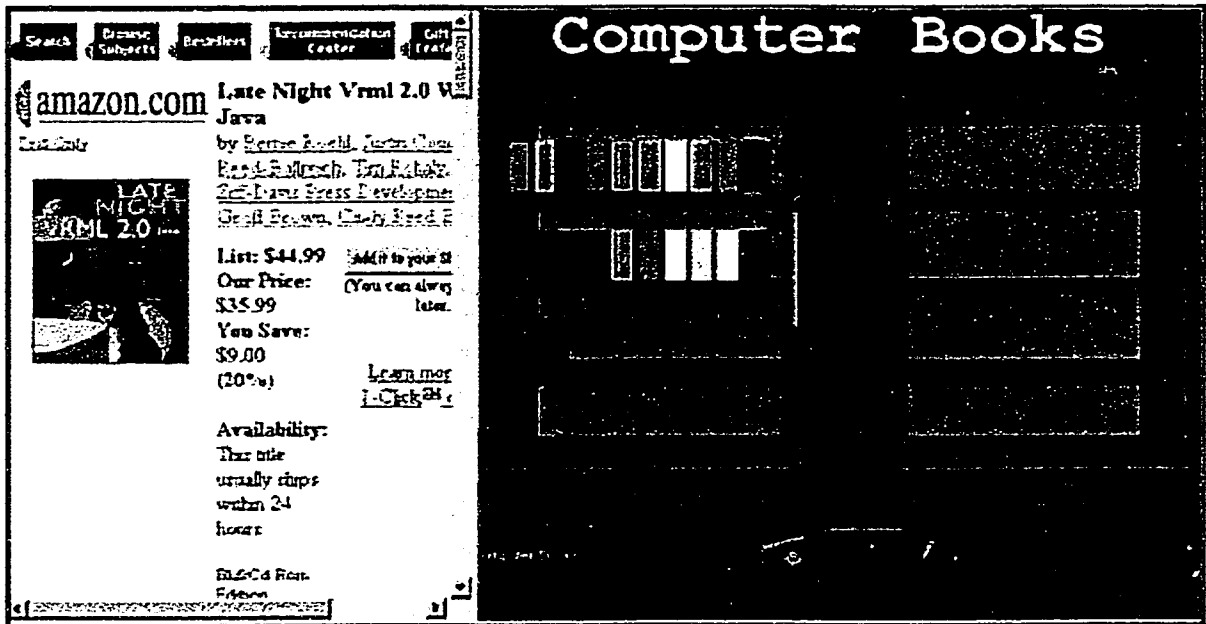


Figure 8 - Computer Books View

#### **4 - Abstract World**

**Mission Statement:** Visualize statistical information by accessing data from a back-end database server and display it through a landscape representation using VRML scene description language.

This experiment uses Canada's [Elections 97] database for the province of Ontario. Information can be displayed based on:

- Riding
- Party
- Votes Received

The chart below [Figure 9] outlines all possible selections.

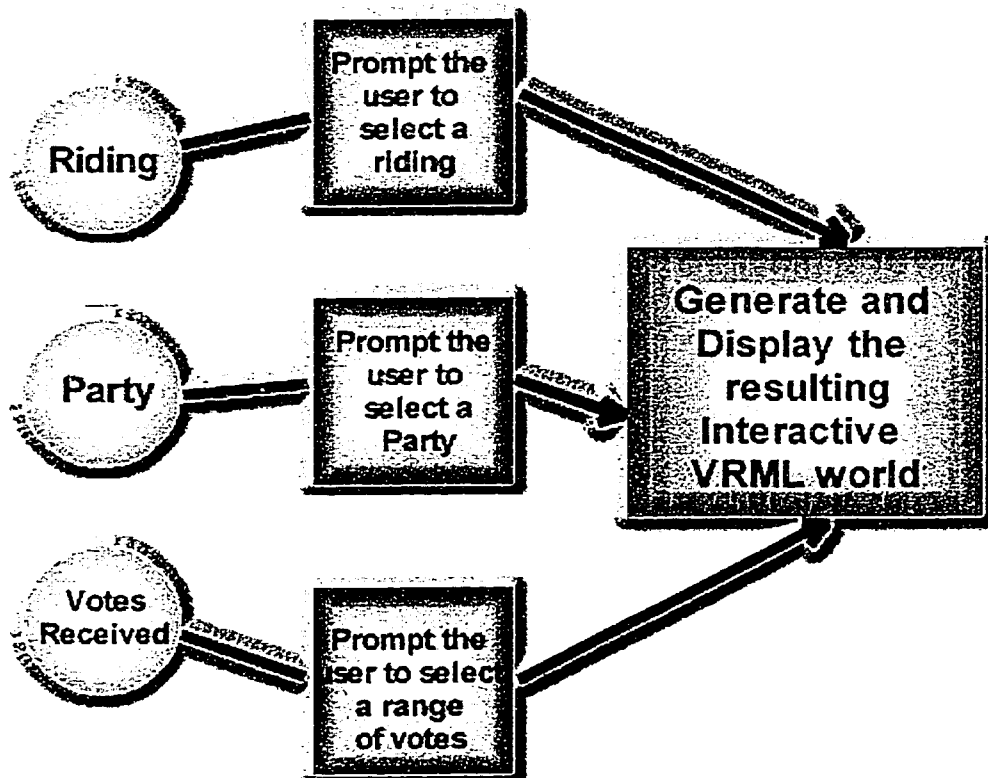


Figure 9 - Elections Queries

You can visit the VRML world at

<http://www.csi.uottawa.ca/~tmilan/Thesis/Elections/ElectionsMain.html>

This experiment requires four tables:

- Candidates table [Figure 10a]
- Parties table [Figure 10b]
- Ridings table [Figure 10c]
- Provinces table [Figure 10d]

	LastName	FirstName	Party	RidingNo	Votes
1	Abou-Dib	Mariam	10	35013	173
2	Adams	Peter	1	35067	25571
3	Aggus	Gerry	3	35095	13486
4	Ainsworth	Bonnie	2	35002	16015
5	Alderson	Victor	3	35039	6850
6	Aldridge	Doug	2	35041	9931
7	Alexander	Rick	9	35065	134
8	Ambas	Tom	2	35077	7854
9	Amos	Bill	9	35052	178
10	Andersen	John	3	35056	12232
11	Annis	Peter	3	35059	8642
12			4	35016	

Figure 10a - Candidates Table View

	PartyNo	PartyName
1	1	Liberal Party of Canada
2	2	Reform Party of Canada
3	3	Progressive Conservative Party
4	4	New Democratic Party
5	5	The Green Party of Canada
6	6	Christian Heritage Party of Canada
7	7	Canadian Action Party
8	8	Marxist-Leninist Party of Canada
9	9	Natural Law Party of Canada
10	10	No Affiliation
11	11	Independent

Figure 10b - Parties Table View

RidingNo	RidingName	Prov	Regists	Votes	KLoc	VLoc
1	35001 Algoma--Manitoulin	6	53506	33429	45	25
2	35002 Barrie--Simcoe--Bradford	6	82976	54269	62	8
3	35003 Beaches--East York	6	66349	45559	67	15
4	35004 Bramalea--Gore--Malton	6	65931	40818	64	16
5	35005 Brampton Centre	6	62511	38102	60	14
6	35006 Brampton West--Mississ	6	72883	45511	64	13
7	35007 Brant	6	71632	45164	59	9
8	35008 Broadview--Greenwood	6	60601	42426	66	15
9	35009 Bruce--Grey	6	69454	48763	47	12
10	35010 Burlington	6	70704	49783	63	11
11	35011 Cambridge	6	73543	48100	59	10
12	35012 ...	6	40271	25699	62	
		6	63973	43087		

Figure 10c - Ridings Table View

ProvinceNo	ProvinceName
1	Newfoundland
2	P.E.I.
3	Nova Scotia
4	New Brunswick
5	Quebec
6	Ontario
7	Manitoba
8	Saskatchewan
9	Alberta
10	British Columbia
11	Yukon
12	Northwest Territories

Figure 10d - Provinces Table View

The tables are linked based on the data model presented in [Figure 11]

All tables are linked using one- to-one mapping; i.e. every record in one table corresponds to one record in the linked table.

The numeric field "PartyNo" links "Candidates" table to "Parties" table.

The numeric field "RidingNo" links "Candidates" table to "Ridings" table.

The numeric field "ProvinceNo" links "Ridings" table to "Provinces" table.

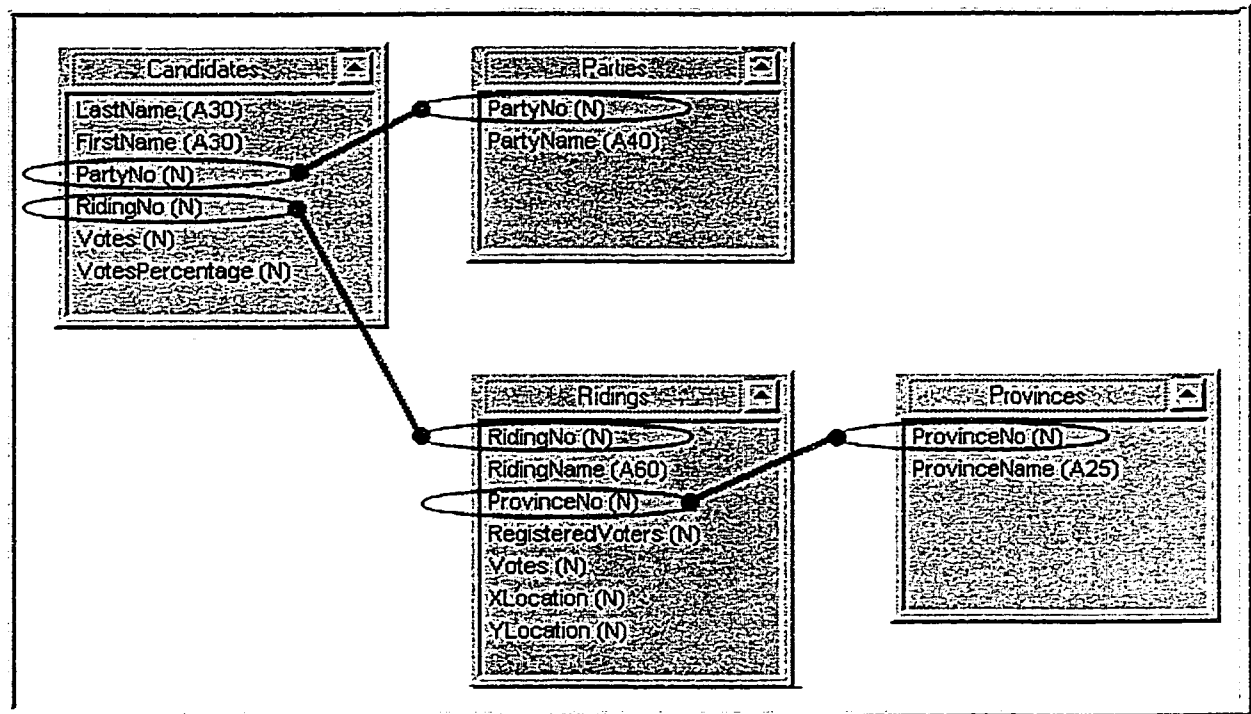


Figure 11 - Elections Data Model

The data model translates into a SQL statement [Figure 12]

The SQL statement includes only the fields that will be required by the experiment.

Executing the SQL statement generates a table [Figure 13] showing the result of the SQL statement and renders additional queries faster in the sense that only one table needs to be accessed at once.

```
SELECT Candidates.LastName, Candidates.FirstName, Candidates.Votes,  
Candidates.RidingNo, Ridings.RidingName, Ridings.ProvinceNo,  
Provinces.ProvinceName, Ridings.RegisteredVoters, Ridings.Votes,  
Candidates.PartyNo, Parties.PartyName, Ridings.XLocation,  
Ridings.YLocation  
FROM "Candidates" Candidates  
INNER JOIN "Ridings" Ridings  
ON (Candidates.RidingNo = Ridings.RidingNo)  
INNER JOIN "Parties" Parties  
ON (Candidates.PartyNo = Parties.PartyNo)  
INNER JOIN "Provinces" Provinces  
ON (Ridings.ProvinceNo = Provinces.ProvinceNo)
```

Figure 12 - Elections Unified Table SQL Statement

	LastName	FirstN	Votes	RidNo	RidingName	Pr	Provinc	Regist	Vo1	PN0	PartyName	XL	YL
1	Wildman	Jody	7902	35001	Algoma--Manitou	6	Ontario	53506	33429	4	New Democratic Party	45	25
2	MacDonald	Rosean	3367	35001	Algoma--Manitou	6	Ontario	53506	33429	3	Progressive Conservativ	45	25
3	Jeffery	Jim	8353	35001	Algoma--Manitou	6	Ontario	53506	33429	2	Reform Party of Canada	45	25
4	St. Denis	Brent	13807	35001	Algoma--Manitou	6	Ontario	53506	33429	1	Liberal Party of Canada	45	25
5	Woods	Ian	326	35002	Barrie--Simcoe--B	6	Ontario	82976	54269	7	Canadian Action Party	62	8
6	McComb	Peggy	2581	35002	Barrie--Simcoe--B	6	Ontario	82976	54269	4	New Democratic Party	62	8
7	Vander Kooi	Dan	451	35002	Barrie--Simcoe--B	6	Ontario	82976	54269	6	Christian Heritage Party	62	8
8	Trotter	John	10708	35002	Barrie--Simcoe--B	6	Ontario	82976	54269	3	Progressive Conservativ	62	8
9	Carroll	Aileen	23582	35002	Barrie--Simcoe--B	6	Ontario	82976	54269	1	Liberal Party of Canada	62	8
10	Ainsworth	Bonnie	16015	35002	Barrie--Simcoe--B	6	Ontario	82976	54269	2	Reform Party of Canada	62	8
11	Sternberg	Marie	606	35002	Barrie--Simcoe--B	6	Ontario	82976	54269	5	The Green Party of Can	62	8
12	Scheer	John	586	35003	Beaches--East Yo	6	Ontario	66349	45559	5	The Green Party of Can	67	15
13	Miller	Gary	6539	35003	Beaches--East Yo	6	Ontario	66349	45559	2	Reform Party of Canada	67	15
14	Watkins	Mel	10715	35003	Beaches--East Yo	6	Ontario	66349	45559	4	New Democratic Party	67	15
15	Minna	Maria	21844	35003	Beaches--East Yo	6	Ontario	66349	45559	1	Liberal Party of Canada	67	15
16	Simpson	Jack	5611	35003	Beaches--East Yo	6	Ontario	66349	45559	3	Progressive Conservativ	67	15
17	Fredeen	Donald	264	35003	Beaches--East Yo	6	Ontario	66349	45559	9	Natural Law Party of C	67	15
18	Majeed	Abdul	2411	35004	Bramalea--Gore--I	6	Ontario	65931	40818	4	New Democratic Party	64	16
19	Florence	Darlene	8671	35004	Bramalea--Gore--I	6	Ontario	65931	40818	2	Reform Party of Canada	64	16
20	Ford	Beryl	10650	35004	Bramalea--Gore--I	6	Ontario	65931	40818	3	Progressive Conservativ	64	16
21	Malhi		8907	35004	Bramalea--Gore--I	6	Ontario	65931	40818	1	Liberal Party of Canada	64	16
22					Bramalea--Gore--I	6	Ontario	65931	40818	8	M	64	16

Figure 13 - Candidates Unified Table View

After the user makes a selection, an HTML page is dynamically constructed. Once the user selects "Riding", a query is executed and a list of ridings is retrieved from the back-end database server. The list is displayed in a combo box through which the user can select a specific riding.

After selecting a riding from the combo box, a dynamic VRML world is generated consisting of bar charts, each of a different color representing different candidates/parties [Figure 14]. The height of the bar is proportional to the number of received votes. The candidate with the highest bar wins. By keeping two different frames on screen, one containing the list of ridings and the other containing the VRML view, context is preserved.

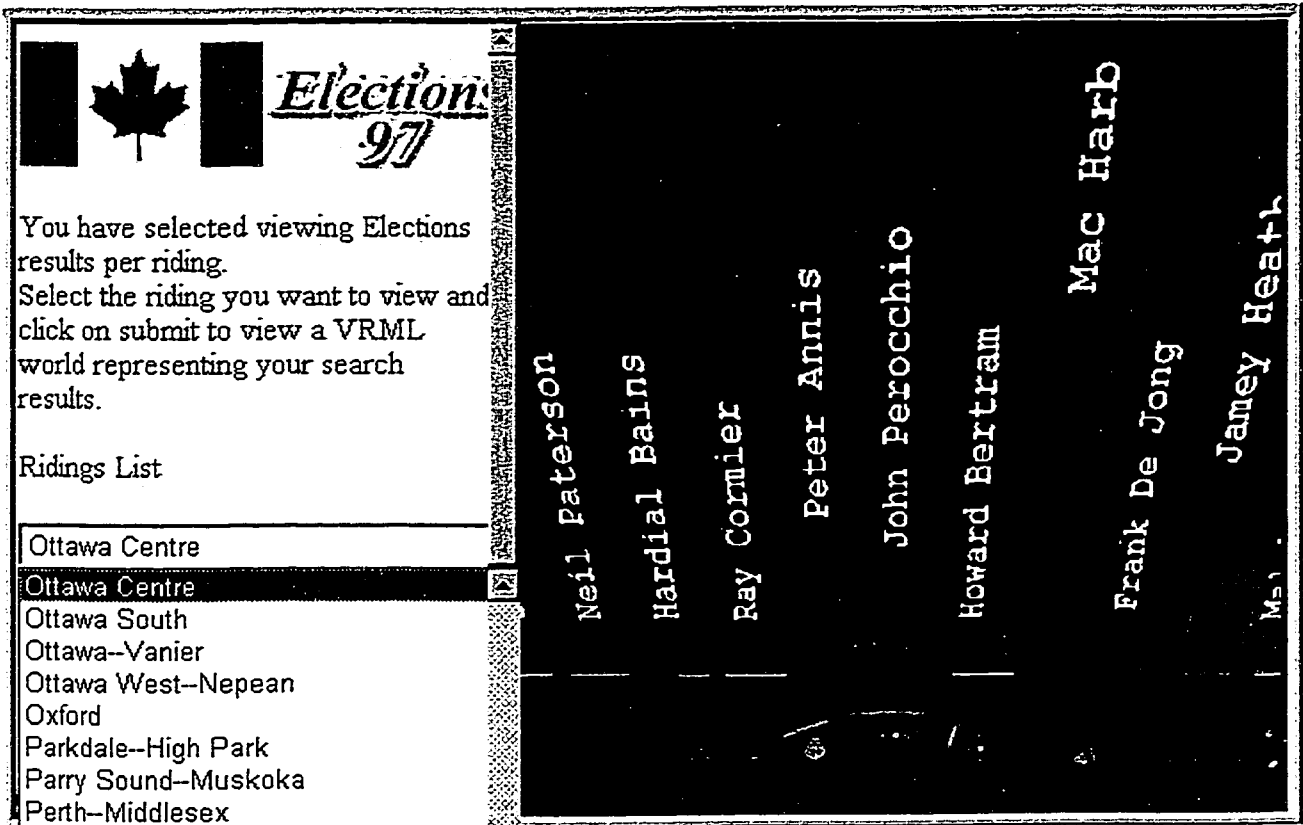


Figure 14 - Ridings Screen

Returning to the previous level, once the user selects "Votes Received", she will be prompted to [Figure 15]:

- select a range of votes
- enable/disable showing candidates' names
- enable/disable showing ridings' names and allow that to be displayed as a plane text or as a "billboard"
- link the generated VRML object to "Elections Canada" Riding 2D map. This feature shows how a 3D world can be hooked into an external site.
- enable/disable showing the number of votes that every candidate received


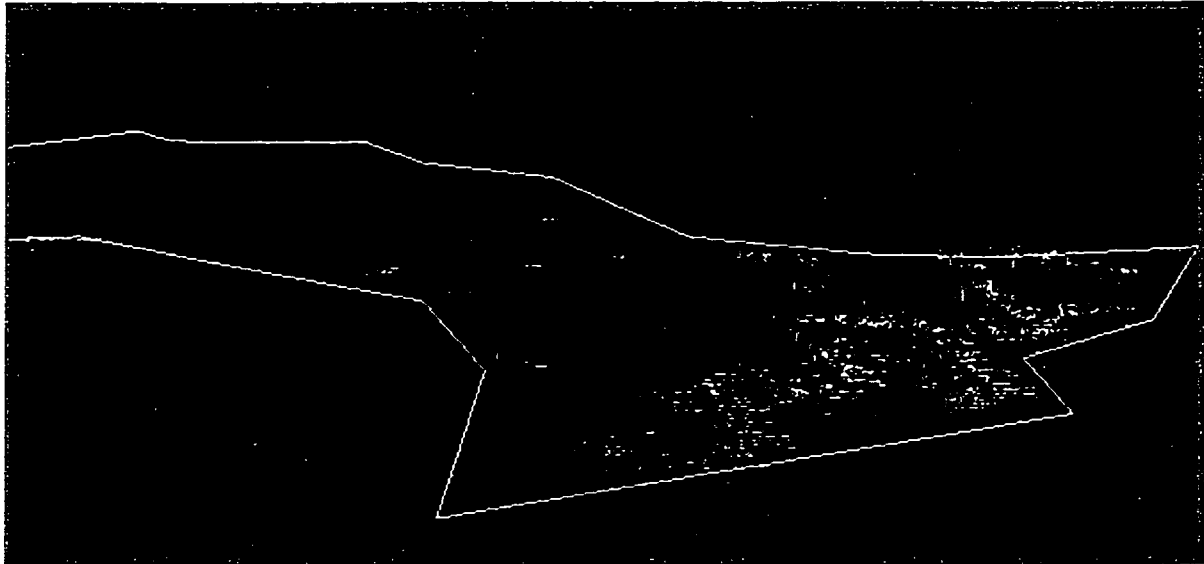
<p>You have selected viewing Elections results based on received votes. Select the range of votes you want and click on submit to view a VRML world representing your search results.</p> <p>Select Candidates who have received between <input type="text" value="0"/> and <input type="text" value="100000"/> votes.</p> <p><input checked="" type="checkbox"/> Show Candidates Names <input checked="" type="checkbox"/> Show Ridings Names <input checked="" type="checkbox"/> Use Billboard to display Ridings Names <input checked="" type="checkbox"/> Link Objects to Elections Canada Site <input checked="" type="checkbox"/> Show Received Votes per Candidate</p> <p><input type="button" value="Submit"/></p>	 <p>Check Canada's latest Parliamentary el For now, only Ontario's elections result You can view candidates by Riding, Pa</p> <p>View Elections results based on:</p> <ul style="list-style-type: none"><li>• <u>Riding</u></li><li>• <u>Party</u></li><li>• <u>Votes Received</u></li><li>• <u>Candidate overall result</u></li></ul> <p>Check the sketch below for a quick preview</p>
--	--

Figure 15 - Votes Received Screen

The above steps are essential in keeping the generated VRML world small enough to be useful for navigation. If all of the above options are selected at the same time with the range kept between 0 and 100000, the resulting world will be big enough (approximately 500 K) to render navigation almost impossible. Lots of calculations and rendering power will be required to refresh this complex world in addition to the initial world generation and download times.

Clicking the "Submit" button generates a VRML world that includes a bar chart for every candidate matching the range of received votes [Figure 16].



**Figure 16 - Ontario's Map Far View**

Notice the concentration of ridings in the south of Ontario. The generated Ontario map is skewed and that is why you see the Southern part bigger than the northern part. Let us zoom in more to view a specific riding in more detail.

If “Link Objects to Elections Canada Site” check box was selected before generating the 3D world, the resulting VRML objects will have hyperlinks pointing to ridings’ pages on “Elections Canada” site [Figure 17] .

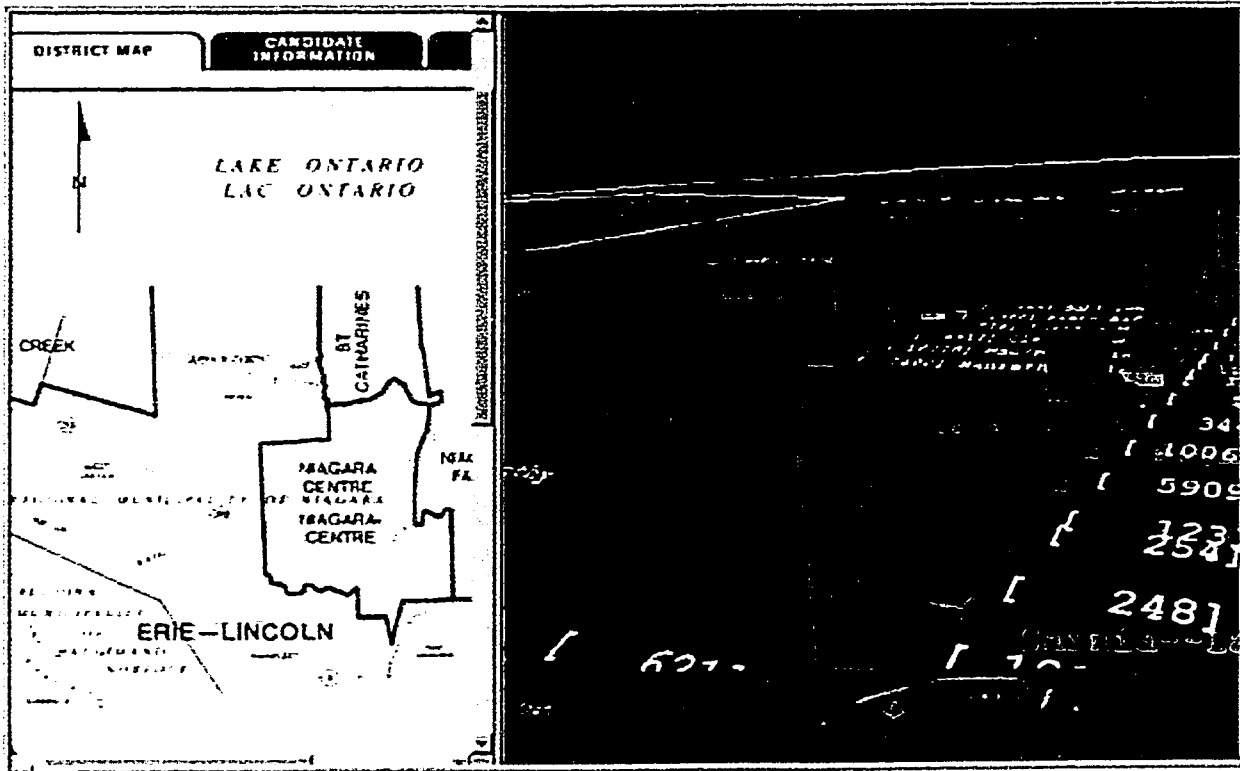


Figure 17 - Ontario's Map Close View

Once the VRML world is loaded, clicking on the riding name or on any of the riding's candidates within that world will fetch its 2D map and a candidates' list of that riding from the "Elections Canada" site and display it in the HTML frame to the left of the VRML world.

Similar to "Multiple Databases/Multiple Interfaces" experiment, this demonstrates the power of mixing VRML and HTML and also shows the possibilities of creating a 3D world that can be hooked to existing commercial (e.g. [Amazon](#)) or non-commercial (e.g. Elections Canada) sites.

## **VI - Conclusion and Future Work**

### **A - Conclusion**

VRML is the Internet 3D file format standard. It offers the essential flexibility that is needed to create a full navigation and visualization environment.

Viewpoints performance is better than manual navigation which requires extensive adjustments. This is demonstrated through "Viewpoint World" and "Multiple Databases/Multiple Interfaces World" experiments. Viewpoints help the user navigate through a large complex world and prevent submersion and avoid loss of direction. The smooth animation that is shown when moving between different books categories is fast and efficient. It showcases how transition between different objects in a 3D complex world can be continuous. Presenting the books on different shelves creates a scene that is simple to visualize. Navigation becomes intuitive when the user is familiar with her surroundings and knows that through viewpoints corrections can be made to prevent getting submerged through manual navigation.

On the other side, manual navigation in 3D space with 2D devices like the mouse and the keyboard can be challenging and frustrating. The mouse supports 2 degrees of navigation and lacks the necessary 6 degrees of navigation that are needed for seamless 3D experience. This is demonstrated again through Experiment 3 if the user tries to navigate between 2 bookshelves. Loss of direction is common in that situation since the scene looks more like a maze and the user can get easily lost within.

Visualizing data in 3D space and linking 3D objects to different data sources, that can be scattered all across the web, proved useful in leveraging the extensive amount of information that is available to the world at large without a need to duplicate it locally. This is presented through "Viewpoint World", "Multiple Databases/Multiple Interfaces World", and "Abstract World" experiments. Creating a 3D object, such as a riding or a candidate, and anchoring it to an external data source helps keep storage requirements to a minimum. The external reference can be

accessible as long as it will keep the same interface (such as books ISBN ) that can be referenced by a client site.

The local database keeps track of the book's title, ISBN, category, and author name. The VRML parser queries the local database and packages every entry into a VRML object that is linked to an external web server through its ISBN field. The external site keeps additional information such as: price, co-authors, customers reviews, cover, outline, publisher, ordering information, availability, other books by the same author, other books with same topic, ...

The "Drill-Down World" experiment, which keeps track of the client state for subsequent drill-downs all the while accessing a back-end SQL server, proves that VRML can support database enabled applications. By drilling down from "Databases" to "Tables" to "Fields" to "Records", context is preserved. This can support multiple simultaneous http requests without a need to keep context information locally on the server since this information is passed on to the client as embedded CGI parameters that will be used for subsequent drill-downs.

Large VRML worlds can result in rendering navigation impossible. This is demonstrated in "Abstract World" experiment where the generated VRML scene can be up to 500 KB in size and may consist of 1000 3D bars and 1000 3D text objects. Under these conditions, navigating either manually or through viewpoints is practically impossible due to the large scene that needs to be constantly rendered for every movement. Also the time required to generate the VRML world and download it can be a negative factor especially for slow Internet connections.

VRML's performance depends on existing browsers and on the reliability of both the client and the server machines. Basic VRML worlds work fine under different system configurations; however, once Java and/or JavaScript nodes are used, various hosting browsers behave differently. Various versions of the same browser (such as Netscape's Communicator 4.05 and 4.06) may cause some VRML code to behave inconsistently.

VRML browsers render worlds differently on different machines and different VRML browsers

render worlds differently on the same machine. So, based on the browser, the VRML viewer, and the platform, results can end up being different. This is forcing VRML Web sites to have browser specific VRML worlds.

## **B - How can VRML be Improved to Better Handle Databases, Navigation Support and Overall Reliability?**

The viability of VRML depends on its adoption in enterprise computing and database driven applications. Oracle is spearheading the database working group effort to propose a database node extension to VRML. One apparently important requirement that the database working group dropped from their proposal is “built-in support for database triggers”. Database triggers are an essential part of the “push technology”. When the data changes, a trigger event informs the client to update its content. The VRML scene responds by updating its scene accordingly. The work group argued that a Java applet achieves the same result by trapping the database trigger event and forcing the VRML world to update its scene.

Another important prerequisite for VRML survival is overall reliability.

This can be achieved by:

- **(Consistency)** All major browsers and VRML viewers must agree on additional standards
  - Rendering mechanism
  - Support for Java and Java script nodes
  - External Authoring Interface (EAI)
- **(Stability)** The crash of VRML browser viewing a world is common and because the crash occurs from within a browser it is hard to lay blame and therefore fix

VRML is constantly improving and there are currently 17 active work groups [[Trevett](#)] promoting new areas and ideas such as

- Built-in database support through a new SQL node

- Humanoid animation
- Living worlds (multiuser environments)
- VRML-DHTML (dynamic HTML) integration

Here are several further requirements that demand attention if VRML is to become ready for prime time.

- **(Faster Rendering)** Improvements in 3D accelerators will lead to faster rendering.
- **(More Bandwidth/Faster Downloads)** More Bandwidth and Faster Downloads are consequences of:
  - faster Internet connection and/or
  - smaller compressed VRML (10 to 50 times smaller than the original text file format). According to IBM and Paragraph Inc. this is a reality and their latest proposition was adopted by the VRML Consortium.
- **(Level Of Details)** Design the VRML worlds using Level Of Details (LOD) node. A VRML world can be designed using LOD node where objects closer to the current viewpoint appear in the highest detail and distant objects appear in lower detail. This results in faster rendering since a lesser number of polygons are drawn.
- **(3D input devices)** The mouse, a 2D input device, is not designed to handle 3D navigation. Standardized and affordable 3D devices are way overdue. There are interesting prototypes emerging such as the “Rockin Mouse” [[Balakrishnan](#)]
- **(Improved Hardware)** VRML will require faster machines and more RAM

### **C - How Can the Experiments be Improved?**

The purpose of the experiments is to illustrate the interaction between VRML and databases. The system's architecture is heavily based on CGI, a relatively old technology, now in its third year. In fact CGI is still the most used Internet technology to retrieve/update data from/to a database.

The most notable limitation of this technology is that every http request will launch a separate process on the server. Typically, once a CGI script is launched through a Web browser form request, it will establish a link to a database on the server and this operation might take a few seconds just to launch the SQL server instance. It is easy to run out of processes as a result of too many http requests and this is when the user receives the annoying "Server Too Busy" notification.

Other alternatives to CGI:

1 - Using Servelets instead of CGI scripts:

- **(Advantages)** The advantage behind this technology is that the Servlet application is always running in memory and for every http request, a new thread is generated. Threads requires less time to initialize than processes and require less memory since they share the memory context and resources of the process they are contained in. Response time will be faster as the Servlet would have already established a link to the SQL server avoiding re-initialization with every http request.
- **(Disadvantages)** Servlets can only be written in Java and using C/C++ or Perl is not possible. Also updating the Servlet may require shutting down the Web server.

2 - Instead of running scripts or Servlets on the server, a Java applet can be downloaded and executed on the client.

- **(Advantages)** Applet establishes connection to the database server and most of the processing time will be spent on the client machine ,thus relieving the server from all the additional work. This frees up the server to only handle database requests and results in faster response time. Also, this results in additional User Interface (UI) controls, supported through Java, that can be used on the client side.
- **(Disadvantages)** Client browser requires a Java Virtual Machine to interpret the Java code. Java applet classes have to be downloaded to the client machine.

How can individual experiments be extended?

Since HTML has a limited set of UI functions, limited to cover only fields, edit boxes, radio buttons and check boxes, it is not possible to design a custom made UI to handle specific things like adding sliders, tree controls, ... Java can solve this problem by designing an applet to produce all the UI bells and whistles. This applet can then communicate with another VRML world embedded in another browser frame using the External Authoring Interface (EAI).

Individual experiments can benefit from a variety of improvements.

- **(Viewpoint World)**
  - Use book color to parametrize publication date
  - Use book color to match keywords
  - Use book size to parametrize number of pages
- **(Abstract World)**
  - Use different axes or directions to display statistical information in 3D

- Use charts to display seats won or lost by parties
- Use special highlighting for the candidate with most votes and for the candidate with least votes
- Display more information about the electors such as sex and age
- Connect every riding to a viewpoint where the user can navigate to different ridings by simply selecting the riding name from a list (probably through a popup context menu list)
- Allow the user to select a scale where text font size and object dimensions can be specified at run time

**References:**

[**Amazon**] Amazon books. [www.amazon.com](http://www.amazon.com)

[**Ames**] A. Andrea, D. Nadeau and J. Moreland. The VRML Sourcebook. John Wiley & Sons Inc., 1996

[**Balakrishnan**] R. Balakrishnan. The Rockin' Mouse: Integral 3D Manipulation on a Plane. *In Proceedings of CHI 97*, 1997. <http://www.acm.org/sigchi/chi97/proceedings/paper/rbl.htm>

[**Benderson**] B. Bederson and J.D. Hollan. Pad++: A zoomable graphical interface system. *Proceedings of CHI 96*, 1996  
[http://www.acm.org/sigchi/chi95/Electronic/documnts/demos/pad\\_bdy.htm](http://www.acm.org/sigchi/chi95/Electronic/documnts/demos/pad_bdy.htm)

[**Bowman**] D. Bowman, L. Hodges and J. Bolter. THE VIRTUAL VENUE PROJECT Advanced User Interface Techniques for Information-Rich Virtual Environments. *Georgia Tech's Graphics Visualization and Usability center*, 1997  
<http://www.cc.gatech.edu/gvu/virtual/Venue/Venue.html>

[**BRO92**] - Brodlie, K.W. et al. Scientific Visualization, Techniques and Applications, Springer 1992

[**Cave**] D. Pape, The CAVE Virtual Reality System, 1997. <http://www.evl.uic.edu/pape/CAVE/>

[**Chuah**] M. Chuah, S. Roth, J. Mattis and J. Kolojejchick. SDM: Selective Dynamic Manipulation of Visualizations. School of Computer Science, Carnegie Mellon University, 1995  
<http://www.cs.cmu.edu/Groups/sage/SDM-UIST95/SDM-UIST95.html>

[**Coors**] V. Coors and V. Jung. Using VRML as an Interface to the 3D Data Warehouse. *In Proceedings of the VRML 98 symposium*, 1998.  
<http://ece.uwaterloo.ca:80/vrml98/papers/content.html#paprlistDB3>

[**Cowperthwaite**] D. Cowperthwaite, M. Sheelagh, T. Carpendale, and F. Fracchia. Visual Access for 3D Data. *Proceedings of CHI 96*, 1996.  
<http://www.acm.org/sigchi/chi96/proceedings/shortpap/Cowperthwaite/djc.htm>

[**Daessler**] R. Daessler. Visualization of Abstract Information, 1995.  
<http://banzai.msi.umn.edu/~roelf/tut.html>

[**Darken93**] R. Darken, and J. Sibert. A Toolset for Navigation in Virtual Environments. *Proceedings of ACM User Interface Software & Technology*, 1993. pp.157-165.  
[http://www.seas.gwu.edu:80/faculty/sibert/darken/publications/Toolset\\_UIST93/navigation\\_tools\\_et.html](http://www.seas.gwu.edu:80/faculty/sibert/darken/publications/Toolset_UIST93/navigation_tools_et.html)

[**Darken96**] R. Darken and J. Sibert (1996). Wayfinding Strategies and Behaviors in Large Virtual Worlds. *In Proceedings of ACM SIGCHI 96*, 1996

[http://www.seas.gwu.edu:80/faculty/sibert/darken/publications/Strategies\\_CHI96/strategies.html](http://www.seas.gwu.edu:80/faculty/sibert/darken/publications/Strategies_CHI96/strategies.html)

[Dieberger] A. Dieberger. Navigation in Textual Virtual Environments using a City Metaphor. Doctor of Technical Sciences Thesis, Submitted at the Vienna University of Technology. Faculty of Technology and Sciences, 1994. <http://www.lcc.gatech.edu/~dieberger/Thesis/Thesis.html>

[Dilbert] VRML recreational clips. <http://umweb1.unitedmedia.com/comics/dilbert/vrml/>

[Disz] T. Disz, M. Papka, M. Pellegrino, and R. Stevens. Sharing Visualization Experiences among Remote Virtual Environments. In *Proceedings of the International Workshop on High Performance Computing for Computer Graphics and Visualization*, 1995. [http://www.mcs.anl.gov/FUTURES\\_LAB/VR/PAPERS/WALES/share.html](http://www.mcs.anl.gov/FUTURES_LAB/VR/PAPERS/WALES/share.html)

[Donskoy] M. Donskoy, and V. Kaptelinin. Window navigation with and without animation: a comparison of scroll bars, zoom, and fisheye view. In *Proceedings of CHI 97*, 1997. <http://www.acm.org/sigchi/chi97/proceedings/short-talk/mdo.htm>

[Elections 97] Election Canada Site at: [www.elections.ca](http://www.elections.ca)

[FOL94] J. Foley and W. Ribarsky. Next-Generation Data Visualization Tools. Scientific Visualization, Academic Press Ltd. 1994. pp. 103-127

[Furnas] G. Furnas and B. Bederson. Space-Scale Diagrams: Understanding Multiscale Interfaces. In *Proceedings of CHI 95*, 1995. [http://www.acm.org/sigchi/chi95/Electronic/documnts/papers/gwf\\_bdy.htm](http://www.acm.org/sigchi/chi95/Electronic/documnts/papers/gwf_bdy.htm)

[Hamit] F. Hamit. Virtual Reality and the Exploration of Cyberspace. Sams Publishing, 1993

[Hartman] J. Hartman and J. Wernecke. The VRML 2.0 Handbook: Building Moving Worlds on the Web. Addison-Wesley Pub Co.1996

[Hix] D. Hix, J. Templeman and R. Jacob. Pre-Screen Projection: From Concept to Testing of a New Interaction Technique. In *Proceedings of CHI 95*, 1995. [http://www.acm.org/sigchi/chi95/Electronic/documnts/papers/dh\\_bdy.htm](http://www.acm.org/sigchi/chi95/Electronic/documnts/papers/dh_bdy.htm)

[Hibbard] B. Hibbard and B. Paul. Vis5D a system for interactive visualization of large 5-D gridded data sets. Space Science and Engineering Center, Madison, Wisconsin, 1997. <http://www.ssec.wisc.edu/~billh/vis5d.html>

[Hughes Technologies] Home of mini SQL at: <http://www.Hughes.com.au/>

[Instone] Keith Instone. HCI and the Web. *CHI 96 Workshop*, 1996. <http://www.acm.org/sigchi/web/chi96workshop/poster/navresearch.html>

[Johnson] J. Johnson. A Comparison of User Interfaces for Panning on a Touch-Controlled Display. In *Proceedings of CHI 95*, 1995. <http://www.acm.org/sigchi/chi95/proceedings/papers/jaj2bdy.htm>

[Kaptelinin] V. Kaptelinin. A Comparison of Four Navigation Techniques in a 2D Browsing Task. *In Proceedings of CHI 95*, 1995.

[http://www.acm.org/sigchi/chi95/Electronic/documnts/shortppr/vk\\_bdy.htm](http://www.acm.org/sigchi/chi95/Electronic/documnts/shortppr/vk_bdy.htm)

[Lee] R. Lee and S. Petrov. A Dynamic, Schema-Independent Web Interface for a Relational Database. *Computer Science & Mathematics Division, Intelligent Systems Section, Oak Ridge National Laboratory*, 1995. <http://re7alpha.epm.ornl.gov/WWW95/>

[Leigh] J. Leigh, A. Johnson and T. DeFanti. Issues in the Design of a Flexible Distributed Architecture for Supporting Persistence and Interoperability in Collaborative Virtual Environments. *In Proceedings of SC97: High Performance Networking and Computing*, 1997.

<http://scxy.tc.cornell.edu/sc97/program/TECH/LEIGH/INDEX.HTM>

[Lokuge] I. Lokuge and S. Ishizaki. GeoSpace: An Interactive Visualization System for Exploring Complex Information Spaces. *In Proceedings of CHI 95*, 1995.

[http://www1.acm.org/sigchi/chi95/proceedings/papers/il\\_bdy.htm](http://www1.acm.org/sigchi/chi95/proceedings/papers/il_bdy.htm)

[Milan] T. Milan, 1998. [www.csi.uottawa.ca/~tmilan](http://www.csi.uottawa.ca/~tmilan)

[Moore] K. Moore, J. Dykes and J. Wood. Using Java to interact with geo-referenced VRML within a Virtual Field Course. *ICA Visualisation Commission meeting*, 1997.

<http://www.geog.le.ac.uk/mek/usingjava.html>

[Munzner] T. Munzner and P. Burchard.. Visualizing the Structure of the World Wide Web in 3D Hyperbolic Space, 1996. <http://www.geom.umn.edu/docs/research/webviz/>

[Murdock] K. Murdock, L. Lemay and J. Couch, 3D Graphics and VRML 2. Sams Publishing, 1996

[OLAPS] 3-D Visualization Research at National Weather Service Forecast Office. National Weather Service Forecast Office, Norman OK, 1995.

<http://www.nssl.uoknor.edu/~nws/3-dvis.html>

[Oracle VRML Cartridge] Oracle Corporation. Oracle VRML Cartridge white Paper, 1997.

<http://www.olab.com/vrml/html/whitepaper.html>

[OXF89] - The Oxford English Dictionary, Second Edition, CLARENDON PRESS - OXFORD 1989

[Pape1] Pape's Topographic VRML 2.0 Map Generator, Electronic Visualization Laboratory of the UIC EECS Department, 1997. <http://www.evl.uic.edu/pape/vrml/etopo/>

[Pimentel] K. Pimentel and K. Teixeira, Virtual Reality through the new looking glass, McGraw-Hill Inc., 1993

[Planet9] Planet 9 Studios at: <http://www.planet9.com/indexnetscape.htm>

[Povary] T. Bradley. Forest Stand Data Visualization with POVRAY, 1997.

<http://www.silvafor.org/povray/povray.htm>

[Ressler] S. Ressler, Q. Wang, S. Bodarky, C. Sheppard and G. Seidman. Using VRML to Access Manufacturing Data. VRML97 conference, 1997.

<http://www.nist.gov/itl/div894/ovrt/projects/mfg/SIMA/vrml97/vimhtml.fm.html>

[Roehl] B. Roehl, J. Couch, C. Reed-Ballreich, T. Rohaly and G. Brown. Late Night VRML 2.0 with Java. Ziff-Davis Press, 1997

[Schartz] B. Schartz. Information Retrieval in Digital Libraries: Bringing Search to the Net. Bioinformatics, Vol. 275, January 1997

[SAGE] M. Derthick, SAGE (System for Automated Graphics and Explanation). School of Computer Science, Carnegie Mellon University, 1996.

<http://www.cs.cmu.edu/Groups/sage/project/sage.html>

[SEN94] H. Senay and E. Ignatius. A Knowledge-Based System for Visualization Design, IEEE Computer Graphics & Applications Nov Vol:14 No:6 pp.36, 1994

[Shilman] M. Shilman, A. Sinha and W. Hsiao. Moby: Hierarchical Information Visualization, 1997

[Siebert] L. Siebert, Parable: A Multi-dimensional, VRML Based, Financial Visualization Application. Master's Thesis, University of Manchester, 1995

<http://anubis.cs.umbc.edu/~cikm/1995/npiv/loren/paper.html>

[Stoakley] R. Stoakley, M. Conway and R. Pausch. Virtual Reality on a WIM: Interactive Worlds in Miniature. In *Proceedings of CHI 95*, 1995.

[http://www.acm.org/sigs/sigchi/chi95/proceedings/papers/rws\\_bdy.htm](http://www.acm.org/sigs/sigchi/chi95/proceedings/papers/rws_bdy.htm)

[Tisny] US VRML Statistical Map, Transaction Information System, 1997

<http://www.tisny.com/vrml/usamap2.html>

[Trevett] The VRML Standard & the VRML Consortium, The Future of 3D on the Net.

<http://www.vrml.org/brief/apr98/>

[Tuft1] E. Tufte, *Envisioning Information*, Graphics Press, PO Box 430, Cheshire, CT 06410, 1990. <http://www.gbn.org/BookClub/Envisioning.html>

[Tuft2] E. Tufte. *The Visual Display of Quantitative Information*. Graphics Press, Cheshire, CT, 1983, page 13

[Twidale] M. Twidale and D. Nichols. Collaborative Browsing and Visualization of the Search Process. *Aslib Proceedings*, 1996, 48(7-8), 177-182

<http://www.comp.lancs.ac.uk/computing/research/aai-aiid/information/ariadne/docs/elvira96.htm>

1

[VDI] Visible Decision Inc. [www.vdi.com](http://www.vdi.com)

[VRML97-Intro] VRML97 ISO/IEC 14772-1:1997 Specifications -- Introduction at:  
<http://www.vrml.org/Specifications/VRML97/part1/introduction.html>

[VRML-DBGGroup] VRML Database Working Group,  
<http://www.vrml.org/WorkingGroups/dbwork/>

[Ward] M. Ward. <http://cs.wpi.edu/~matt/courses/cs563/talks/datavis.html>

## **Appendix A - Glossary of VRML and Data Visualization Terms**

**2-D or 2D:** Described by dimensions of width and height, but not depth. A 2-D data point exists on a flat surface represented as an (x,y) data value. Images are almost always 2-D, even in animated sequences.

**3-D or 3D:** Having dimensions of width, height and depth. A 3-D data point is represented as an (x,y,z) tuple. 3-D space represents the world in which we live and the synthetic model space used for animation sequences.

**4-D or 4D:** Containing the spatial dimensions of a 3-D representation with the additional element of time which describes the parameter of movement through 3-D space. Real-time interactive graphics and computer animation are examples of 4-D space.

**6 DOF :** Six degrees of freedom; refers to the number of simultaneous directions or inputs a sensor can measure. Typically used to describe the combination of spatial position (X,Y,Z) and orientation (roll, pitch, yaw) measured by many tracking sensors.

**Aliasing :** A term applied to a variety of undesirable visual artifacts which appear in computer generated images. These artifacts are the result of under sampling in the spatial and/or temporal domains. Aliasing is much more noticeable in animation because the eye is very sensitive to rapid changes in the visual field.

**Anti-aliasing :** The removal of unwanted visual artifacts, normally referred to as "jaggies" from lines and/or edges of planes in a 2-D image.

**Backdrop :** In a virtual world, this is the stationary background, analogous to the backdrop in a stage show - the boundary of the world. It cannot be moved or broken into smaller elements.

**BOOM :** A 3-D display device suspended from a weighted boom that can swivel freely about so the viewer does not have to wear an HMD; instead, it steps up to the viewer like a pair of binoculars. The boom's position communicates the user's point of view to the computer.

**CGI :** The Common Gateway Interface is a standard for external gateway programs to interface with information servers such as HTTP servers. For more information on CGI you can check [Mike Smith CGI tutorial](#) or [Newnham College CGI tutorial](#).

**DataGlove :** A glove wired with sensors and connected to a computer system for gesture recognition. Also known generically as a wired glove. It enables navigation through a virtual environment and interaction with 3-D objects within it. DataGlove is a trademark of VPL Research.

**Eye point** : A point in world coordinate space that represents the location of the virtual camera.

**Fractal** : A self-similar pattern generated by using the same rules at various levels of detail.

**HMD** : Abbreviation for "Head Mounted Display," special goggles worn during Virtual Reality. HMDs usually have a tracking device so the VR system can monitor head movements and adjust the video shown in the HMD's display.

**HTML** : HTML stands for hyper-text markup language. It is the standard 2D language which defines Web pages.

**Hyperlink** : A hyperlink is a special area on a Web page which can be activated (usually with a mouse). Most hyperlinks take you to another Web page. Other hyperlinks perform special functions, such as sending email, submitting a form, accessing an ftp site, execute a database query, or access a Usenet newsgroup.

**Immersion** : As applied to virtual reality, this is when one or more of a user's sensors (eyes and ears generally) are isolated from the surrounding environment and fed only information coming from the computer.

**MiniSQL** : A SQL database engine that run on the server side.

**Occlusion** : Objects closer to the viewer overlap and occlude objects in the background.

**Parallax** : Refers to the difference in viewing angle created by having two eyes looking at the same scene from slightly different positions. The combined input to the brain help create a sense of depth.

**SQL** : Structured Query Language.

**Submodalities** : These are the smallest categories into which the senses can divide experience. For example, visual information can be broken down into brightness, saturation, location, and hue. The ears can differentiate sounds by their frequency, rhythm, tone, and duration.

**Telepresence** : The experience of being in another location. Usually accomplished by transmitting the user's view through the eyes of a camera. This can include the operation of remote machinery through the computer translation of human movements into equipment commands.

**Viewpoint** : This represents the point of view of an observer in a virtual world.

**Virtual world** : Created three-dimensional environments that exist within a computer within which the user can directly manipulate objects.

**Visualization :** Formation of an image that can not be seen. The ability to represent abstract data that would normally appears as text and numbers graphically on a computer.

**VRML :** Virtual Reality Modeling Language.

**VR :** Abbreviation for Virtual Reality.

## **Appendix B - Abstract World Source Code**

```

/*****
File Name:      VotesRangeOnMap.c
Written by:     Toufic Milan
Description:    Queries a miniSQL database, extract information and
                generate a VRML world.
                This is the code for "Votes Received" selection which
                is part of Experiment #4.
Last Modified:  May 24, 1998
*****/

// Include Section *****/

#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include "time.h"
#include <timelib.h>
#include <mysql.h>

// Define Section *****/

#define MAXVOTES 100000
#define MINVOTES 0

// Global Variables Section *****/

// Set the Port that will be used to access the MiniSQL Server throughout
// this program. Notice that the server and socket are hardcoded due to
// limitations and security issues in CSI Web server.
char pstrPort[] = "grdb:6545";

// URL pathes
char pstrURLPathFirst[] = "http://www.elections.ca/cgi-win/riding.dll?EN+";
char pstrURLPathLast[] = "+MAPS+False";

// Default Database
```

```
char pstrDatabase[] = "Elections97";

// Declare Candidates Full Table
char pstrTable[] = "CandidatesFull";

// Define the global variables that will determine what the VRML will include.
int bShowCandidateName = 0;
int bShowRidingName = 0;
int bUseBillboard = 0;
int bLinkToElectionsCanada = 0;
int bShowReceivedVotes = 0;

// Declare the color table values.
char pColorStrings[][40] = { "diffuseColor 0.9 0.8 0.7",
                             "diffuseColor 1 0 0",
                             "diffuseColor 0 0 1",
                             "diffuseColor 0.1 0.5 0.1",
                             "diffuseColor 0.5 0.1 0.9",
                             "diffuseColor 1 0 1",
                             "diffuseColor 0 1 1",
                             "diffuseColor 1 1 1",
                             "diffuseColor 0 1 0",
                             "diffuseColor 0 1 0",
                             "diffuseColor 1 1 0",
                             "diffuseColor 1 1 0",
                             "diffuseColor 1 1 0",
                             "diffuseColor 1 1 0" };

// Socket pointer
int nSock;

// Functions Prototype Section *****
void ExtractParam( char *pSrcString, char* varType, char* pDestString );
void GenerateViewPoints( );
void DisplayRidingTitle( char *pstrTranslation, char* pstrRidingName,
                        char* pstrURLString, char * pstrURLDesc );

void CreateURLObject( char* AnchorURL, char* AnchorDesc,
                    char* AnchorTranslation, char* AnchorSize,
```

```
char* AnchorText, char* Color, double dTextLocation );

void ExecuteRangeResults( int nMinVotes, int nMaxVotes );
void DisplayOntarioMap();

// Functions Body Section *****

/*****
Function Name: GenerateViewPoints
Description:   Generate the View Points for the VRML world with the active
              viewpoint being the first one
*****/
void GenerateViewPoints()
{
    printf("\nDEF Start Viewpoint {\n");
    printf("    position 65 20 80\n");
    printf("    description \"Start\"\n");
    printf(" }\n");

    printf("\nDEF entry Viewpoint {\n");
    printf("    position 65 20 80\n");
    printf("    orientation -1 0 0 0.1\n");
    printf("    description \"entry\"\n");
    printf(" }\n");

    printf("\nDEF VIEW0 Viewpoint {\n");
    printf("    position 140 0 +58\n");
    printf("    description \"View -1-\"\n");
    printf(" } # NORTHVIEW\n");

    printf("\nDEF VIEW00 Viewpoint {\n");
    printf("    position 100 40 100\n");
    printf("    description \"View -2-\"\n");
    printf(" } # NORTHVIEW\n");

    printf("\nDEF VIEW11 Viewpoint {\n");
    printf("    position 12 0 12\n");
    printf("    orientation 0 0 1 0.4\n");
    printf("    description \"View -3-\"\n");
}
```

```
printf(" } # NORTHVIEW\n");

printf("\nDEF VIEW12 Viewpoint {\n");
printf("    position 6 0 +6\n");
printf("    orientation 1 1 1 -0.4\n");
printf("    description \"View -4-\"\n");
printf(" } # NORTHVIEW\n");
}

/*****
Function Name: ExtractParam
Description:   Extract the variables that are passed from the HTML form

Param:
    pSrcString:  Form get string that is appended to the CGI script
                 request
    varType:     Name of the Variable
    pDestString: if the variable name was found, this string will
                 return the variable value; otherwise returns NULL
*****/
void ExtractParam( char *pSrcString, char* varType, char* pDestString )
{
    char* pDest = NULL;
    char* pLocalPtr = NULL;
    int nPointer = 0;
    int nStrLen = 0;

    pLocalPtr = pDestString;
    pDest = strstr( pSrcString, varType );

    if ( pDest == NULL )
        return;

    while ( *pDest != '=' )
        pDest++;

    // Sanity check
    if ( *pDest != '=' )
        return;
}
```

```
// Skip the '=' character
pDest++;

// Remove all starting blanks
while ( *pDest == ' ' || *pDest == '+' )
    pDest++;

// Read the Param value until we reach the NULL terminator
// and store it in pDestString
while ( *pDest != '&' && *pDest != '\0' )
{
    if ( *pDest == '+' )
    {
        // Replace + with Blank.
        *pLocalPtr = ' ';
    }
    else
        *pLocalPtr = *pDest;

    pDest++;
    pLocalPtr++;
}

// Make sure the copied string is NULL terminated
*pLocalPtr = '\0';

// Check for trailing blanks.
pLocalPtr--;

// Read the string backward and remove all trailing blanks.
nStrLen = strlen ( pDestString );

if ( nStrLen > 0 && *pLocalPtr == ' ' )
{
    while ( nStrLen > 0 && *pLocalPtr == ' ' )
    {
        nStrLen--;
        pLocalPtr--;
    }
}
```

```
pLocalPtr++;

// Make sure the copied string is NULL terminated
*pLocalPtr = '\0';
}
}

/*****
Function Name: CreateURLObject
Description:   Create the VRML anchor objects that can be used as a
              bridge to other Worlds

Param:
AnchorURL:    Contains URL hyperlink
AnchorDesc:   Contains the URL description that will be shown
              on the status bar once the user moves the mouse
              over a VRML object
AnchorTranslation: Position of the VRML object relative to the
              active pen location
AnchorSize:   Size of the object
AnchorText:   3D Text that is displayed on top of the VRML
              object
Color:        Color of the object
dTextLocation: Location of 3D text with respect to the object
*****/
void CreateURLObject( char* AnchorURL, char* AnchorDesc,
                    char* AnchorTranslation, char* AnchorSize,
                    char* AnchorText, char* Color, double dTextLocation )
{
printf("\nAnchor {\n");
printf("  url      \"%s\"\n", AnchorURL );
printf("  parameter [\"target=Left_Frame\"] \n");
printf("  description \"%s\"\n", AnchorDesc );
printf("  children  {\n");
printf("    Transform {\n");
printf("      translation %s\n", AnchorTranslation );
printf("    children  {\n");
printf("      Shape {\n");
printf("        geometry Box {\n");
```

```
printf("          size %s\n", AnchorSize );
printf("      }\n");
printf("          appearance Appearance {\n");
printf("              material Material { %s }\n", Color );
printf("          } # appearance\n");
printf("      } # Shape\n");

if ( bShowCandidateName || bShowReceivedVotes )
{
printf("          Transform {\n");
printf("              translation +0.3 %6.2f 0\n", dTextLocation );
printf("              rotation 1 0 0 -1.20 \n");
printf("              children [\n");
printf("                  Shape {\n");
printf("                      geometry Text {\n");
printf("                          string \"%s\" \n", AnchorText );
printf("                          fontStyle FontStyle {\n");
printf("                              family \"TYPEWRITER\"\n");
printf("                              justify \"LEFT\"\n");
printf("                              style \"BOLD\"\n");
printf("                              size 0.3 \n");
printf("                      }\n");
printf("                  }\n");
printf("              } # Shape\n");
printf("          ]\n");
printf("      } # Transform \n");
}

printf("    ] # children\n");
printf("    }\n");
printf(" ] # children\n");
printf("}\n");
}
```

/\*\*\*\*\*

Function Name: DisplayRidingTitle

Description: Create a 3D text object that displays the Riding Name

Param:

pstrTranslation: Position of the text object relative to the active pen location  
pstrRidingName: String to be displayed  
pstrURLString: Contains URL hyperlink that will be activated if the user clicks on Riding Text  
pstrURLDesc: Contains the URL description that will be shown on the status bar once the user moves the mouse over the Riding Text

```
*****/
void DisplayRidingTitle( char *pstrTranslation, char* pstrRidingName,
                        char* pstrURLString, char * pstrURLDesc )
{
    printf("\nAnchor {\n");
    printf(" url    \"%s\"\n", pstrURLString );
    printf(" parameter [\"target=Left_Frame\"] \n");
    printf(" description \"%s\"\n", pstrURLDesc );
    printf(" children {\n");
    printf("     Transform {\n");
    printf("         translation %s \n", pstrTranslation );
    printf("         rotation 0 0 1 -1.57 \n");
    printf("         children {\n");

    if ( bUseBillboard )
    {
        printf("         Billboard {\n");
        printf("         axisOfRotation 0 0 0 \n");
        printf("         children {\n");
    }

    printf("         Shape {\n");
    printf("             geometry Text {\n");
    printf("                 string \"%s\" \n", pstrRidingName );
    printf("                 fontStyle FontStyle {\n");
    printf("                     family \"TYPEWRITER\"\n");
    printf("                     justify \"LEFT\"\n");
    printf("                     style \"BOLD\"\n");
    printf("                     size 0.3 \n");
    printf("                 }\n");
    printf("             }\n");
    printf("         appearance Appearance {\n");
```

```
printf("          material Material { diffuseColor 0 1 0 }\n");
printf("          } # appearance\n");
printf("          } # Shape\n");

if ( bUseBillboard )
{
  printf("          ]\n");
  printf("          } # Billboard \n");
}

printf("          ]\n");
printf("          } # Transform \n");
printf("          ] # children\n");
printf(")\n");
}

/*****
Function Name: ExecuteRangeResults
Description:  Retrieves a list candidates who received a certain range
              of votes and map them to VRML objects.

Param:
          nMinVotes:  Minimum number of votes
          nMaxVotes:  Maximum number of votes

N.B. Based on the table definition:
          row[0]  = Name
          row[2]  = votes
          row[11] = XLocation
          row[12] = YLocation
*****/
void ExecuteRangeResults( int nMinVotes, int nMaxVotes )
{
  m_result *pResult = NULL;
  m_row row;
  int i;
  char pstrObjectTranslation[80];
  char pstrObjectSize[80];
  char pstrRidingTitleLocation[80];
  char pstrLocalURL[256];
```

```
char pstrQuery[256];
char* pDatabase = pstrDatabase;
char* pTable = pstrTable;
char* pFieldName = "RidingName";
double dVotes = 0.0;
int nRealVotes = 0;
char pstrFullName[128];
char pstrRidingName[80];
char pstrURLString[256];
char pstrURLDesc[256];
double dRow = 0.0;
double dColumn = 0.0;
double dPartyNo = 0.0;
int nPartyNo = 0;
int nNewColumn = 0;
double dXLocation = 0;
double dYLocation = 0;
double dTempNo = 0;

// Initilize pstrRidingName with a fictitious name
strcpy( pstrRidingName, "XYXYXY");

// Load the main Database
mysqlSelectDB( nSock, pDatabase );

// Construct the SQL query based on the passed in minimum and maximum range
sprintf( pstrQuery, "SELECT * FROM %s WHERE Votes >= %d and Votes<= %d" ,
        pTable, nMinVotes, nMaxVotes );

if ( mysqlQuery ( nSock, pstrQuery ) == - 1 )
{
    // An error ocured while executing the SQL query
    printf("Content-type: text/html\n\n");
    printf(" \n Error in Query : %s ", pstrQuery );
    return;
}

pResult = mysqlStoreResult();

// Iterate through all retrieved record
```

```
for ( i=0; i<mysqlNumRows(pResult); i++ )
{
    row = mysqlFetchRow( pResult );
    if ( strcmp ( pstrRidingName, row[4] ) != 0 )
    {
        // We have a new Riding.
        dRow = 0;
        strcpy( pstrRidingName, row[4] );
        nNewColumn = 1;
    }
    else
    {
        // Still in the same riding.
        // We have another candidate, so we need to display another row.
        dRow += 1.5;
        nNewColumn = 0;
    }

    dVotes = atof( row[2] );
    nRealVotes = (int) dVotes;

    // Every VRML display unit is mapped to 10000 votes.
    dVotes = dVotes / 10000;

    dXLocation = atof( row[11] );
    dYLocation = atof( row[12] );

    // Bump up the scale by a factor of 2 so to have the results more
    // dispersed on the map.
    // Hopefully this will result in less run-on and less overlapping entries.
    dXLocation *= 2;
    dYLocation *= 2;

    dColumn = dXLocation;

    // Preparing the Anchor URL string and its description
    dTempNo = atof( row[3] );
    sprintf( pstrURLString, "%s%5.0f%s", pstrURLPathFirst,
            dTempNo, pstrURLPathLast );
    sprintf( pstrURLDesc, "%s 2D-MAP", row[4] );
```

```
if ( nNewColumn == 1 )
{
    // We need to display the Riding title
    sprintf( pstrRidingTitleLocation, "%5.2f %7.4f %7.4f ",
            dColumn /* * 4 */ , - 2.0, -dYLocation );

    if ( bShowRidingName )
    {
        if ( bLinkToElectionsCanada )
            DisplayRidingTitle( pstrRidingTitleLocation, pstrRidingName,
                                pstrURLString, pstrURLDesc);
        else
            DisplayRidingTitle( pstrRidingTitleLocation, pstrRidingName, "", "" );
    }
}

sprintf( pstrObjectTranslation, "%5.2f %7.4f %7.4f ", dColumn,
        dVotes/2 - 1, - dRow *0.5 - dYLocation);
sprintf( pstrObjectSize, "0.4 %7.4f 0.4 " , dVotes );

strcpy ( pstrFullName, "" );

if ( bShowReceivedVotes )
{
    sprintf ( pstrFullName, "[%6d] ", nRealVotes );
}

if ( bShowCandidateName )
{
    strcat ( pstrFullName, row[0] );
    strcat ( pstrFullName, " " );
    strcat ( pstrFullName, row[1] );
}

dPartyNo = atof( row[9] );
nPartyNo = (int) dPartyNo; // safe casting since PartyNo is an integer.

// Placing the actual call that will actually create the VRML object.
if ( bLinkToElectionsCanada )
```

```
{
    CreateURLObject( pstrURLString, pstrURLDesc, pstrObjectTranslation,
                    pstrObjectSize /*"0.5 9 0.5"*/, pstrFullName,
                    pColorStrings[nPartyNo], - dVotes/2 );
}
else
{
    CreateURLObject( "", "", pstrObjectTranslation, pstrObjectSize,
                    pstrFullName, pColorStrings[nPartyNo], - dVotes/2 );
}

}

// Clear the queries result from the SQL engine memory buffer.
mysqlFreeResult( pResult );
}

/*****
Function Name: DisplayOntarioMap
Description:   Displays ontario map on X and Z axis
*****/
void DisplayOntarioMap()
{
    printf(" \n ");
    printf("DEF OntarioMap Transform {\n");
    printf("children [\n");
    printf("Shape {\n");
    printf("  appearance NULL\n");
    printf("  geometry IndexedLineSet {\n");
    printf("  coord Coordinate {\n");
    printf("  point [ 90 0 5 \n");
    printf("          93 0 -20\n");
    printf("          88 0 -40\n");
    printf("          50 0 -68\n");
    printf("          30 0 -64\n");
    printf("          18 0 -70\n");
    printf("          0 0 -70\n");
    printf("          0 0 -116\n");
    printf("          34 0 -160\n");
```

```
printf("      40 0 -150\n");
printf("      50 0 -144\n");
printf("      76 0 -144\n");
printf("      86 0 -120 \n");
printf("     104 0 -106\n");
printf("     116 0 -66\n");
printf("     132 0 -57\n");
printf("     144 0 -55\n");
printf("     167 0 -59\n");
printf("     148 0 -32\n");
printf("     133 0 -22\n");
printf("     131 0 -10 ]\n");
printf("    ]\n");
printf("    coordIndex [\n");
printf("      0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 0 -1\n");
printf("    ]\n");
printf("  ]\n");
printf(" ]\n");
printf("]\n");
printf("}\n");
}
```

```
/******
Function Name: main
Description:  First function to be called when the program is executed.
*****/
main()
{
  char pRidingName[80];
  char DATA[200];
  char *buf = NULL;
  char pMaxVotes[80];
  char pMinVotes[80];
  char pTempBuff[126];
  int nMaxVotes = 0;
  int nMinVotes = 0;
  int size = 0;
  int i;
  int nTemp;
```

```
pRidingName[0] = '\0';
bShowCandidateName = 0;
bShowRidingName = 0;
bUseBillboard = 0;
bLinkToElectionsCanada = 0;
bShowReceivedVotes = 0;

// Get the CGI form request extra info
buf = getenv("QUERY_STRING");

if ( !buf )
{
    // Trying to execute the CGI script directly without filling in
    // the necessary form variables first.
    // This typically happens when the URL of the CGI script is
    // selected without having additional variables attached to the
    // URL string
    printf("Content-type: text/html\n\n");
    printf("Error : A CGI program was called directly\n");
    printf("Should be called from within a form\n");
    return(0);
}

// Get the size of QUERY_STRING buffer
size = strlen( buf );

// Store the retrieved buffer locally
for ( i=0; i<size; i++ )
{
    DATA[i] = *buf;
    buf++;
}

// Terminate the string with a NULL terminator
DATA[i] = '\0';

// Extract passed parameters.
ExtractParam( DATA, "Riding=", pRidingName );
ExtractParam( DATA, "MinVotes=", pMinVotes );
ExtractParam( DATA, "MaxVotes=", pMaxVotes );
```

```
nMinVotes = atoi ( pMinVotes );
nMaxVotes = atoi ( pMaxVotes );

pTempBuff[0] = '\0';
ExtractParam( DATA, "ShowCandidateName=", pTempBuff );

if ( pTempBuff[0] != '\0' )
    bShowCandidateName = 1;

pTempBuff[0] = '\0';
ExtractParam( DATA, "ShowRidingName=", pTempBuff );

if ( pTempBuff[0] != '\0' )
    bShowRidingName = 1;

pTempBuff[0] = '\0';
ExtractParam( DATA, "UseBillboard=", pTempBuff );

if ( pTempBuff[0] != '\0' )
    bUseBillboard = 1;

pTempBuff[0] = '\0';
ExtractParam( DATA, "LinkToElectionsCanada=", pTempBuff );

if ( pTempBuff[0] != '\0' )
    bLinkToElectionsCanada = 1;

pTempBuff[0] = '\0';
ExtractParam( DATA, "ShowReceivedVotes=", pTempBuff );

if ( pTempBuff[0] != '\0' )
    bShowReceivedVotes = 1;

// Sanity check.
// Making sure that Min is Less than Max. If for some reason the range
// of received numbers is invalid, we use Min=MINVOTES and Max=MAXVOTES
if ( nMinVotes < MINVOTES || nMinVotes > MAXVOTES )
    nMinVotes = MINVOTES;
```

```
if ( nMaxVotes < MINVOTES || nMaxVotes > MAXVOTES )
    nMaxVotes = MAXVOTES;

if ( nMaxVotes < nMinVotes )
{
    nTemp = nMinVotes;
    nMinVotes = nMaxVotes;
    nMaxVotes = nTemp;
}

if ( nMaxVotes < nMinVotes || ( nMaxVotes == 0 && nMinVotes == 0 ) )
{
    nMinVotes = MINVOTES;
    nMaxVotes = MAXVOTES;
}

// Connect to miniSQL server.
nSock = mysqlConnect(pstrPort);

if (nSock < 0)
{
    printf("Content-type: text/html\n\n");
    printf(" \n Error connecting to mSQL server \n ");
    exit(0);
}

// At this stage all initialization succeeded.
// The content type is set to VRML so that the generated
// text will be routed to the VRML plugin.

printf("Content-type:x-world/x-vrml \n\n");
printf("#VRML V2.0 utf8\n\n");

GenerateViewPoints();
DisplayOntarioMap();
ExecuteRangeResults( nMinVotes, nMaxVotes );

// Close connection to mSQL server.
mysqlClose(nSock);
```

```
return(0);  
}
```

## **Appendix C - Sample of the Generated VRML Source Code**

The following VRML world source code was generated based on:

- Experiment 4, "Votes received" selection,
- Votes between 10000 and 10500
- Show Candidates names
- Link objects to Elections Canada site
- Show received votes

```
#VRML V2.0 utf8

DEF Start Viewpoint {
  position 65 20 80
  description "Start"
}

DEF entry Viewpoint {
  position 65 20 80
  orientation -1 0 0 0.1
  description "entry"
}

DEF VIEW0 Viewpoint {
  position 140 0 +58
  description "View -1-"
} # NORTHVIEW

DEF VIEW00 Viewpoint {
  position 100 40 100
  description "View -2-"
} # NORTHVIEW

DEF VIEW11 Viewpoint {
  position 12 0 12
  orientation 0 0 1 0.4
  description "View -3-"
} # NORTHVIEW

DEF VIEW12 Viewpoint {
  position 6 0 +6
  orientation 1 1 1 -0.4
  description "View -4-"
} # NORTHVIEW

DEF OntarioMap Transform {
  children [
    Shape {
      appearance NULL
      geometry IndexedLineSet {
        coord Coordinate {
          point [ 90 0 5
                 93 0 -20
                 88 0 -40
```

```

    50 0 -68
    30 0 -64
    18 0 -70
    0 0 -70
    0 0 -116
    34 0 -160
    40 0 -150
    50 0 -144
    76 0 -144
    86 0 -120
    104 0 -106
    116 0 -66
    132 0 -57
    144 0 -55
    167 0 -59
    148 0 -32
    133 0 -22
    131 0 -10 ]
}
coordIndex [
  0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 0 -1
]
}
]
}
}
}
Anchor {
  url "http://www.elections.ca/cgi-win/riding.dll?EN+35007+MAPS+False"
  parameter ["target=Left_Frame"]
  description "Brant 2D-MAP"
  children [
    Transform {
      translation 118.00 -0.4819 -18.0000
      children [
        Shape {
          geometry Box {
            size 0.4 1.0361 0.4
          }
          appearance Appearance {
            material Material { diffuseColor 0 0 1 }
          } # appearance
        } # Shape
        Transform {
          translation +0.3 -0.52 0
          rotation 1 0 0 -1.20
          children [
            Shape {
              geometry Text {
                string "[ 10361] Houssar Dan"
                fontStyle FontStyle {
                  family "TYPEWRITER"
                  justify "LEFT"
                  style "BOLD"
                  size 0.3
                }
              }
            }
          ] # Shape
        } # Transform
      ]
    }
  ]
}
```

```
    ] # children
  }
] # children
}

Anchor {
url "http://www.elections.ca/cgi-win/riding.dll?EN+35018+MAPS+False"
parameter ["target=Left_Frame"]
description "Elgin--Middlesex--London 2D-MAP"
children [
  Transform {
translation 114.00 -0.4843 -16.0000
children [
  Shape {
geometry Box {
size 0.4 1.0314 0.4
}
appearance Appearance {
material Material { diffuseColor 0 0 1 }
} # appearance
} # Shape
  Transform {
translation +0.3 -0.52 0
rotation 1 0 0 -1.20
children [
  Shape {
geometry Text {
string "[ 10314] Van Der Veen John"
fontStyle FontStyle {
family "TYPEWRITER"
justify "LEFT"
style "BOLD"
size 0.3
}
}
} # Shape
} # Transform
] # children
}
] # children
}

Anchor {
url "http://www.elections.ca/cgi-win/riding.dll?EN+35022+MAPS+False"
parameter ["target=Left_Frame"]
description "Etobicoke--Lakeshore 2D-MAP"
children [
  Transform {
translation 118.00 -0.4759 -26.0000
children [
  Shape {
geometry Box {
size 0.4 1.0482 0.4
}
appearance Appearance {
material Material { diffuseColor 0.1 0.5 0.1 }
} # appearance
} # Shape
  Transform {
```

```
translation +0.3 -0.52 0
rotation 1 0 0 -1.20
children [
  Shape {
    geometry Text {
      string "[ 10482] Donley Charles"
      fontStyle FontStyle {
        family "TYPEWRITER"
        justify "LEFT"
        style "BOLD"
        size 0.3
      }
    }
  } # Shape
] # children
}

Anchor {
  url "http://www.elections.ca/cgi-win/riding.dll?EN+35055+MAPS+False"
  parameter ["target=Left_Frame"]
  description "Northumberland 2D-MAP"
  children [
    Transform {
      translation 136.00 -0.4755 -36.0000
      children [
        Shape {
          geometry Box {
            size 0.4 1.0490 0.4
          }
          appearance Appearance {
            material Material { diffuseColor 0 0 1 }
          } # appearance
        } # Shape
        Transform {
          translation +0.3 -0.52 0
          rotation 1 0 0 -1.20
          children [
            Shape {
              geometry Text {
                string "[ 10490] Matthews A1"
                fontStyle FontStyle {
                  family "TYPEWRITER"
                  justify "LEFT"
                  style "BOLD"
                  size 0.3
                }
              }
            } # Shape
          ] # children
        } # Transform
      ] # children
    }
  ] # children
}

Anchor {
```

```
url "http://www.elections.ca/cgi-win/riding.dll?EN+35068+MAPS+False"
parameter ["target=Left_Frame"]
description "Pickering--Ajax--Uxbridge 2D-MAP"
children [
  Transform {
    translation 140.00 -0.4803 -40.0000
    children [
      Shape {
        geometry Box {
          size 0.4 1.0394 0.4
        }
        appearance Appearance {
          material Material { diffuseColor 0.1 0.5 0.1 }
        } # appearance
      } # Shape
      Transform {
        translation +0.3 -0.52 0
        rotation 1 0 0 -1.20
        children [
          Shape {
            geometry Text {
              string "[ 10394] Lewis Leanne"
              fontStyle FontStyle {
                family "TYPEWRITER"
                justify "LEFT"
                style "BOLD"
                size 0.3
              }
            }
          } # Shape
        ] # children
      } # Transform
    ] # children
  }
] # children
}
```

```
Anchor {
url "http://www.elections.ca/cgi-win/riding.dll?EN+35071+MAPS+False"
parameter ["target=Left_Frame"]
description "Sarnia--Lambton 2D-MAP"
children [
  Transform {
    translation 108.00 -0.4969 -10.0000
    children [
      Shape {
        geometry Box {
          size 0.4 1.0062 0.4
        }
        appearance Appearance {
          material Material { diffuseColor 0 0 1 }
        } # appearance
      } # Shape
      Transform {
        translation +0.3 -0.50 0
        rotation 1 0 0 -1.20
        children [
          Shape {
            geometry Text {
              string "[ 10062] Christie Dave"
            }
          }
        ]
      }
    ]
  }
]
```

```
        fontStyle FontStyle {
            family "TYPEWRITER"
            justify "LEFT"
            style "BOLD"
            size 0.3
        }
    } # Shape
} # Transform
] # children
} # children
}

Anchor {
    url "http://www.elections.ca/cgi-win/riding.dll?EN+35072+MAPS+False"
    parameter ["target=Left_Frame"]
    description "Sault Ste. Marie 2D-MAP"
    children [
        Transform {
            translation 82.00 -0.4857 -54.0000
            children [
                Shape {
                    geometry Box {
                        size 0.4 1.0287 0.4
                    }
                    appearance Appearance {
                        material Material { diffuseColor 0.5 0.1 0.9 }
                    } # appearance
                } # Shape
                Transform {
                    translation +0.3 -0.51 0
                    rotation 1 0 0 -1.20
                    children [
                        Shape {
                            geometry Text {
                                string "[ 10287] Dietrich Phyllis"
                                fontStyle FontStyle {
                                    family "TYPEWRITER"
                                    justify "LEFT"
                                    style "BOLD"
                                    size 0.3
                                }
                            }
                        } # Shape
                    ]
                } # Transform
            ] # children
        }
    ] # children
}

Anchor {
    url "http://www.elections.ca/cgi-win/riding.dll?EN+35082+MAPS+False"
    parameter ["target=Left_Frame"]
    description "Stoney Creek 2D-MAP"
    children [
        Transform {
            translation 128.00 -0.4911 -16.0000
```

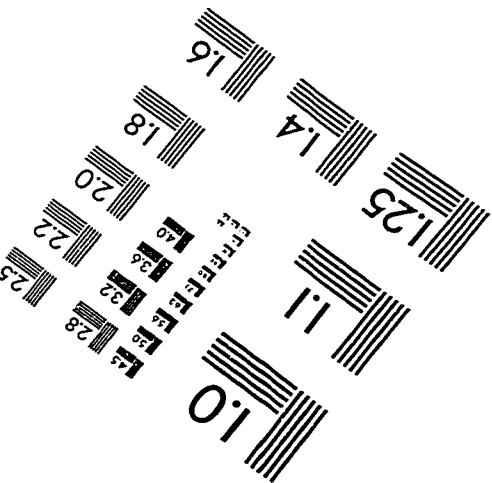
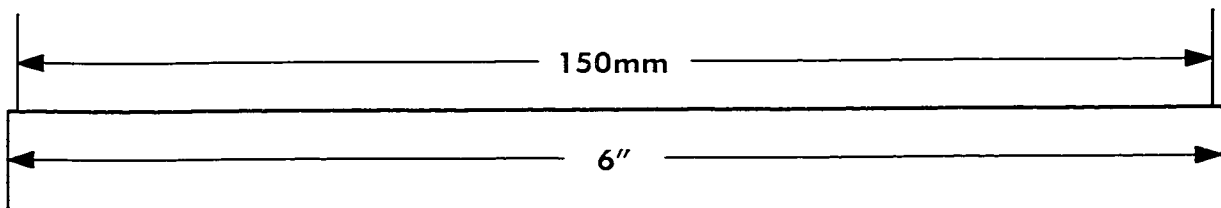
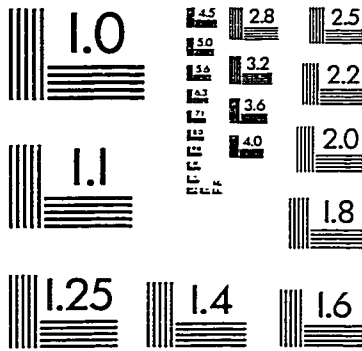
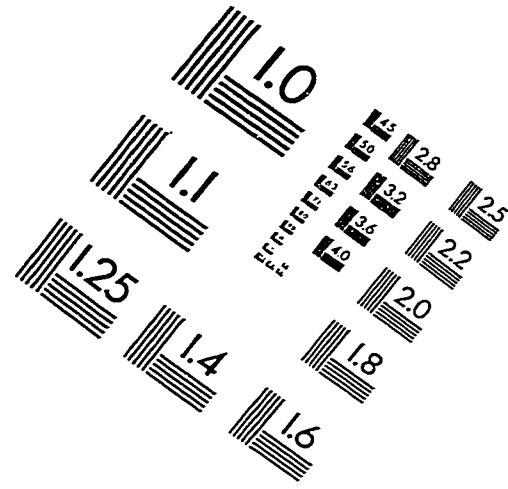
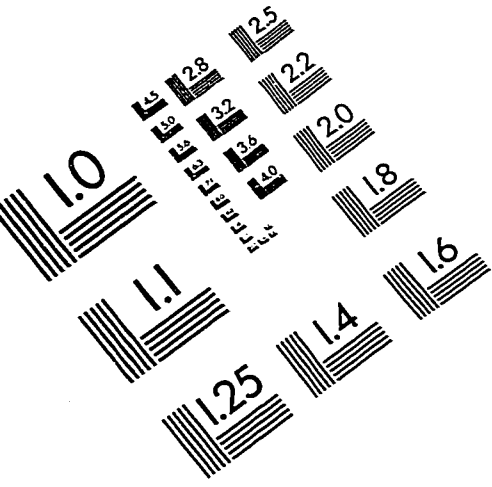
```
children [
  Shape {
    geometry Box {
      size 0.4 1.0177 0.4
    }
    appearance Appearance {
      material Material { diffuseColor 0 0 1 }
    } # appearance
  } # Shape
  Transform {
    translation +0.3 -0.51 0
    rotation 1 0 0 -1.20
    children [
      Shape {
        geometry Text {
          string "[ 10177] Downes Clay"
          fontStyle FontStyle {
            family "TYPEWRITER"
            justify "LEFT"
            style "BOLD"
            size 0.3
          }
        }
      } # Shape
    ]
  } # Transform
] # children
}

Anchor {
  url "http://www.elections.ca/cgi-win/riding.dll?EN+35095+MAPS+False"
  parameter ["target=Left_Frame"]
  description "Wentworth--Burlington 2D-MAP"
  children [
    Transform {
      translation 122.00 -0.4889 -24.0000
      children [
        Shape {
          geometry Box {
            size 0.4 1.0222 0.4
          }
          appearance Appearance {
            material Material { diffuseColor 0 0 1 }
          } # appearance
        } # Shape
        Transform {
          translation +0.3 -0.51 0
          rotation 1 0 0 -1.20
          children [
            Shape {
              geometry Text {
                string "[ 10222] Lonn Allan"
                fontStyle FontStyle {
                  family "TYPEWRITER"
                  justify "LEFT"
                  style "BOLD"
                  size 0.3
                }
              }
            }
          ]
        }
      ]
    }
  ]
}
```

```
        }
      } # Shape
    ]
  } # Transform
] # children
}
] # children
}

Anchor {
  url "http://www.elections.ca/cgi-win/riding.dll?EN+35096+MAPS+False"
  parameter ["target=Left_Frame"]
  description "Whitby--Ajax 2D-MAP"
  children [
    Transform {
      translation 142.00 -0.4947 -36.0000
      children [
        Shape {
          geometry Box {
            size 0.4 1.0106 0.4
          }
          appearance Appearance {
            material Material { diffuseColor 0.1 0.5 0.1 }
          } # appearance
        } # Shape
        Transform {
          translation +0.3 -0.51 0
          rotation 1 0 0 -1.20
          children [
            Shape {
              geometry Text {
                string "[ 10106] Snyder Frank"
                fontStyle FontStyle {
                  family "TYPEWRITER"
                  justify "LEFT"
                  style "BOLD"
                  size 0.3
                }
              }
            } # Shape
          ]
        } # Transform
      ] # children
    } # children
  ]
}
```

# IMAGE EVALUATION TEST TARGET (QA-3)



APPLIED IMAGE, Inc  
1653 East Main Street  
Rochester, NY 14609 USA  
Phone: 716/482-0300  
Fax: 716/288-5989

© 1993, Applied Image, Inc.. All Rights Reserved

