DIVE-ON: From Databases to Virtual Reality

by Ayman Ammoura

Introduction

For many years virtual reality has been associated with science fiction, fantasy, and entertainment. On the fictional starship Enterprise, the crew uses the Holodeck to learn and experiment with new concepts in a "natural" way. Have we, in reality, reached a stage where virtual reality environments can be put to use for "real work" applications? This article will take you on a basic tour of virtual reality technologies and introduce a novel approach where virtual reality is used as an environment for interactive visual data mining. Our research goal is to use virtual reality technology to enable users to increase the amount of information that can be extracted from databases. The extracted information is presented in a manner that takes advantage of the human visual system, which is unrivaled as a processor of spatial data and as a pattern recognizer. We present information in the form of 3D geometric objects in an Immersed Virtual Environment (**IVE**), where the user learns concepts by walking, flying and interacting with the objects making up the virtual world.

"Data mining in an Immersed Virtual Environment Over a Network", **DIVE-ON**, is the name of a system that utilizes advances in virtual reality, databases, and distributed computing to experiment with a new approach to visual data mining. This article is organized into three main sections. The concept of immersion in a virtual environment is first presented along with the state of the art IVE system, the CAVE theater. In the second segment we describe how a typical Database Management System (**DBMS**) is transformed to accommodate knowledge discovery operations with respect to a theme of interest. The last segment presents the main system architecture and how its components, remote and local, are integrated to provide a transparent working environment for data analysis and exploration.

Background

Virtual Reality

Virtual Reality (**VR**) is a field in computer science built around the human visual and sensorimotor systems. To better understand these systems, let us do an experiment. Walk into an area where there is no one else but you and look around at the objects that make up the space: walls, chairs, pictures and anything else that happens to be present. Fix your eyes at a particular point or object and start to reposition your head by standing, sitting, or walking. You will notice that every single point around you will "come to life" and your view is under constant change as long as you are moving. Comparing this to a computer graphics scene, it seems that as you move your entire surroundings get "redrawn" at an incredible refresh rate. This is the essence of Immersed Virtual Environments (IVE). The environment is called 'virtual' because it is computer generated, and 'immersed' because it provides its user with a *sense of presence*. With this understanding it should be clear that the ability to render realistic imagery is only a secondary measure of the quality of an IVE system. The speed and accuracy of coordinating the appropriate image transformation with the user's motion are of primary importance.

To produce such environments, VR technology incorporates specialized input and output devices that allow users to interact with and experience an artificial environment as if it were the real world. The user wears a tracker that records the head location in (x, y, z) and the polar coordinates of its orientation (T1 in Figure 1). This information is collected at a very high sampling rate and put into a data structure that is fed to the graphics engine once per cycle. DIVE-ON uses MR-Toolkit [7] to obtain this data stream in the form of pre-formatted data structures.

Within the VR environment another tracker is used that is functionally equivalent to a typical mouse. This device is often referred to as a 3D pointer, or hand-held tracker. The data structure that is received from this tracker (T2 Figure 1) is used by the VR system to draw a pointer in the virtual world. Since the tracker is capable of delivering its orientation along with its position, the pointer drawn provides a true 3D mapping of the user's hand motion. As you will see later, this device can be used to select and interact with the virtual world and its objects.

HMD: IVE You Wear

Since the early years of VR research, great research and development effort was directed to Head Mounted Displays (**HMD**) to create an immersive virtual environment. HMD consists of a helmet that the user wears which includes a display. This helmet is connected to a computer that updates the internal display. Some HMD helmets are equipped with a single display while others provide a display for each eye to enable stereoscopic graphics. The input to a typical HMD system is obtained from a digital glove. This glove is equipped with a tracker and several finger sensors that can be used to form gesture commands. The helmets in most of the older HMD systems were tethered or attached to a ceiling-mounted boom; however, most of the later models developed are geared towards free ranging helmets.

CAVE: IVE You View

While the gathering and building of information can be done from any location, the actual visualization experience takes advantage of a sophisticated virtual reality environment called VizRoom (formally known as CAVE Theater, CAVE and VizRoom are used interchangeably). **CAVE** is a recursive acronym (Cave Automatic Virtual Environment) [3] and refers to a visualization environment that places the user within three (9.5 X 9.5 feet) walls (Figure 1). Each of these walls is back-projected with a high-resolution projector that delivers the rendered graphics at 120 frames per second (Figure 4). The graphics projected are in stereo (60 frames per second for each eye), which enables DIVE-ON to create stereoscopic views that can be seen by the user by wearing lightweight shutter glasses. The CAVE in VizRoom is powered by two SGI Onyx2 InfiniteReality Rack systems with 4-processors and specialized graphics engines running at 195MHz R10000 IP27.

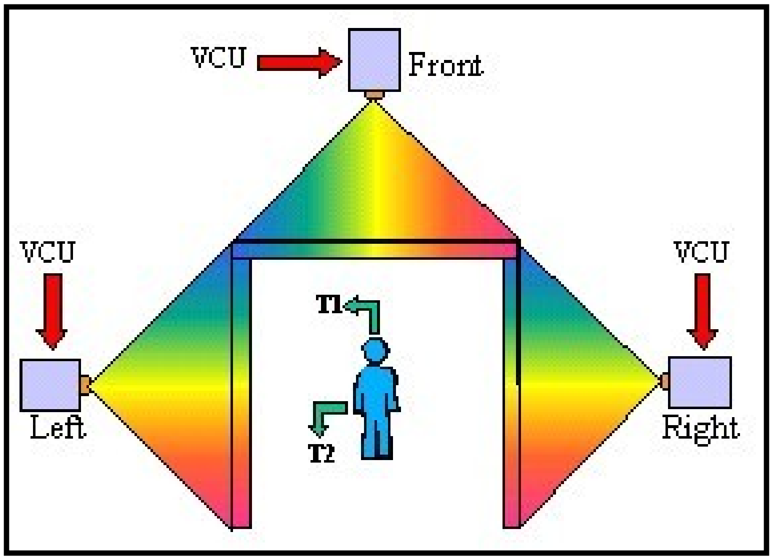

Figure 1: A CAVE user within the three back-projected walls
T1, T2: The head and hand-held tracker data stream respectively (Real-time)

This type of IVE was chosen for the DIVE-ON over HMD systems for several reasons including:

- The user is free to move naturally without the constraints of the HMD.
- As mentioned above, the DIVE-ON is essentially a decision support tool and most likely will be used by a group or a team of analysts. Within the CAVE any view at any time is instantly available to all for examination and discussion (Figure 4).
- Within the walls of the CAVE one is able to use natural means to communicate with others.
- It is easier to increase realism in a CAVE environment since both the left view and the right view are already rendered (on the left and right walls) giving the user ready access to different views by simply turning their head. With the HMD, the head orientation is tracked (T1 in Figure 1) and used to trigger image rotation in correspondence with the user's head rotation.
- From previous experiments, hygienic factors were a big issue for some users. After all, wearing a helmet with an inside display while walking around will definitely make most of us sweat.

Next we will present a definition of a data warehouse, the three phases that make up what is known as "Knowledge Discovery in Databases" or (the **KDD**process), and the operations needed for data mining.

Data: Warehousing and Mining

Corporations worldwide are mining their data to learn about fraud, client purchasing patterns, fleet utilization, credit applications and health care outcome analysis. As discovered by recent research conducted by a leading eResearch company [11], the worldwide business intelligence and data warehousing (BI/DW) market had a 62% year-over-year growth in 1999. This translates to a market share that exceeds $28 billion. As one expects, there has been a surge in the number of applications that provide data warehouse creation, management and mining. All this is made possible by simply knowing how to read what has been collecting in database volumes for many years.

Significant research efforts have been devoted towards facilitating access to pertinent information hiding beneath massive data volumes. Typical relational database systems are well optimized for query processing and online transaction processing (**OLTP**), which minimizes the time needed for systematic daily operations of an organization. These operations consist of a well-structured and repetitive set of atomic transactions that occur in short bursts. However, we seek a data model designed for the use of *knowledge workers* [1] (upper management and analysts) for online analytic processing (**OLAP**), where historical data can be quickly presented in various views and degrees of abstraction. Data warehouses have been designed with that purpose in mind.

Data Warehousing

A **data warehouse** is "*a subject-oriented, integrated, time-variant and non-volatile collection of data in support of management's decision making process"* [5]. This definition is comprehensive and distinguishes data warehouses from all other data repositories. It is built *not* to facilitate the day-to-day operations of an organization (OLTP) but to provide predictions, patterns, anomalies or evidence upon which certain corporate decisions can be made. Constructing the data warehouse requires the transformation of traditional data models (usually the ER model) that exist on DBMS systems into a multidimensional subject-oriented data model [6] (example follows). Data warehouses are built with a **central theme** in mind. The notion of **dimension** can be thought of as the perspective from which an organization wants to view their data. For example, a company may want to construct a data warehouse for the purpose of budget analysis. In this case, the central theme could be "dollars_budgeted" while some of the possible dimensions may be "location", "product" and "time." With such a warehouse, we are able to instantly obtain budgeting information regarding a given "product" in a "location" for a specific "time". To be able to better support OLAP operations, a data warehouse is often implemented as a hierarchical N-dimensional data model that is called **data cube** [1, 4]. The data warehouses generated by DIVE-ON are indeed N-dimensional data cubes. The need for a hierarchical data cube model becomes clearer when we discuss the data cube constructor module (DCC).

Data Mining

The entire process of nontrivial extraction of implicit, potentially useful and previously unknown information from a database is called the Knowledge Discovery in Databases process (KDD process). This process consists of three main phases (each phase can be further subdivided). First is the preprocessing phase where irrelevant and incomplete data is removed. Preprocessing transforms the raw data found in a DBMS into a collection of complete data items that is relevant to the main theme. For example, to analyze sales patterns for an international company one would construct a data cube that focuses only on sale figures disregarding irrelevant information that may exist on the flat files or DBMS of a given local branch or location. The second phase is data integration and consolidation that combines several, possibly heterogeneous, preprocessed sources into a homogeneous source that is suitable for mining. The data warehouse (N-dimensional cube) is usually built in this phase. The final step in the KDD process is the iterative data mining phase where mining algorithms are "fine tuned" and reapplied after evaluating their results.

The question we pose here is "why experiment with an IVE tool for such applications?" Most commercial data exploration application has a visualization component. Using visual cues we are able to comprehend more in less time and with less instructional help. The benefit gained by performing such visualizations in an interactive IVE is now two fold as it combines the use of our visual abilities in conjunction with our **sensorimotor** (visual processing for the control of movement) capabilities.

DIVE-ON: System Components

DIVE-ON can be abstracted in terms of three task-specific subsystems, which are tightly coupled to provide the services required. Figure 2 shows the various layers composing the complete system from the server-side (DBMS) to the client-side (The CAVE). The first subsystem is the Data Cube Constructor (**DCC**), which is responsible for creating and managing the data warehouse over the distributed DBMS. The DCC also fulfills incoming data transportation requests (Figure 2: 1 and 2). The second subsystem is the Visualization Control Unit (**VCU**), which is responsible for the creation and handling of the IVE in a manner that maximizes the frame rate to insure that the "reality" in virtual reality is not compromised (Figure 2: 3). The usability of the whole system is the task of the third subsystem, the User Interface Manager (**UIM**) (Figure 2: 4). A communication layer passes requests and their corresponding replies between the subsystems as well as between the constructed federated data warehouse and its DBMS source. This communication is implemented either with CORBA [12] over TCP/IP or with SOAP [1] over HTTP. In later stages of our research, we will evaluate and compare the two implementations. Messages between subsystems are transmitted as XML [10] documents, which contain the requests and the corresponding responses. The VCU and the UIM locally exist in the graphics research facilities at the University of Alberta, while the DCC exists remotely at the data source.

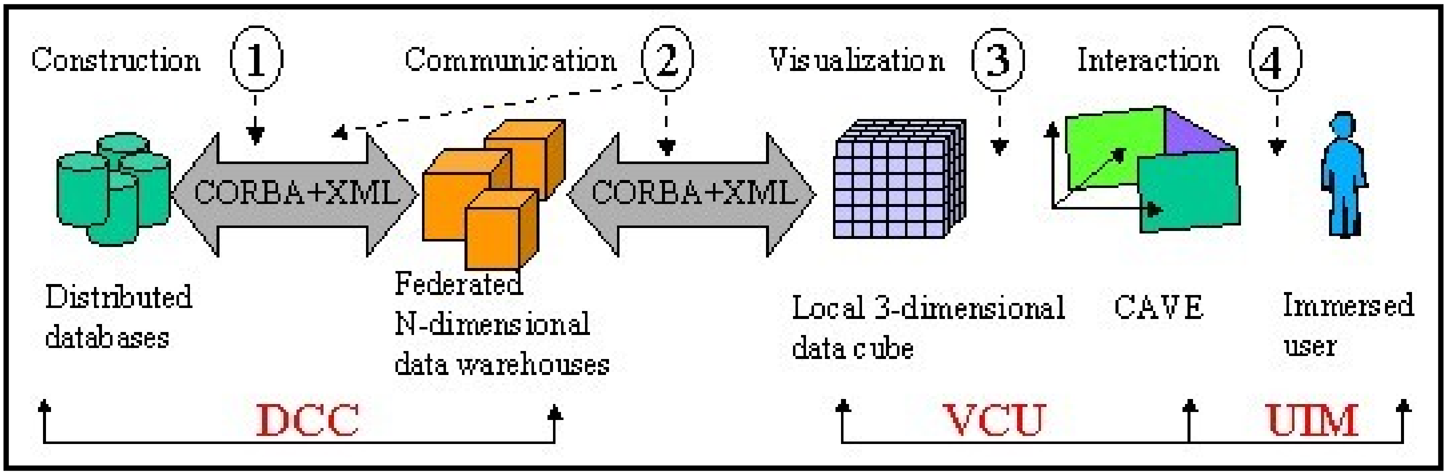

Figure 2: DIVE-ON components from the data source to the virtual environment

Next we will examine these subsystems in the broader context of what they aim to accomplish. First we introduce the DCC side, which includes the construction of a data warehouse appropriate for data mining.

DCC: Data Cube Constructor

The DCC is a DIVE-ON module responsible for completing preprocessing and consolidation (data cube creation). Installed on a possibly remote system, the DCC is the *server* in a distributed client-server model. Raw data from a single or multiple sources is first queried according to a given criteria to isolate incomplete and irrelevant data items (preprocessing). In the case of multiple DBMS sources, the querying process is run on a "main" server that uses CORBA/SOAP to invoke remote methods capable of executing the appropriate SQL queries. This information is then gathered to form the homogeneous, relevant, and consolidated data needed in the data cube creation. The DCC can also be instructed to create more than one data cube in instances where more than one central theme is to be considered. These related, n-dimensional data cubes are called a federated N-dimensional data warehouse (Figure 2).

For the DCC to create the data cube it must first extract structural information about the data sources; this information is used to define each of the N dimensions of the cube. Every dimension can be thought of as a perspective or a logical organization of entities according to which an organization wants to view its data. A dimension definition must also include a **concept hierarchy** which further describes the dimension in terms of a sequence of mappings between low-level concepts and higher-level concepts. Each level in this concept hierarchy defines a **level of abstraction**. For example, a typical data cube dimension is "time", which is usually associated with the concept hierarchy {year, quarter, month, day}. Hierarchical information presentation in the form of different levels of granularity helps support aggregation and summarization for the purpose of informed decision-making. Data viewed at the "day" level (decreased summary) is said to be at a **low abstraction level**. As one moves up the hierarchy, from "day" to "month," the data can be viewed at a **higher abstraction level** providing less detail (increased summary).

It is important to point out that this process cannot possibly be fully automated for the general case because the concept of dimension is user defined and relies heavily on the central theme being considered. For example, for a typical dimension, like "location", described by a concept hierarchy such as {continent, country, region, city}, the raw data within a DBMS may describe locations using only store ID numbers. In such cases, a human expert is needed to define which "city" falls within which "region," the regions making up a "country" and so on.

VCU: Visualization Control Unit

The VCU is the module responsible for generating and managing the Immersed Virtual Environment (IVE) for data visualization and exploration and should be viewed only as such. This means that the specifics of the DCC should be of no concern to the VCU developer and vice versa. To implement this abstract view, each of the VCU and DCC are constructed within a wrapper that provides the only means of relaying messages between the two subsystems. A simple communication protocol that defines a set of requests (VCU to DCC) and their corresponding replays (DCC to VCU) is implemented in DIVE-ON. After the DCC completes the creation of the N-dimensional data cube it signals the VCU. Since we are generating a 3D virtual world, only three dimensions can be viewed at any given time. The three dimensions that are chosen by the user are extracted from the N-dimensional data cube and a 3D data cube is passed to the VCU for rendering. You may wonder, in light of the above discussion, what level of abstraction does that the 3D cube represent? In other words, is the 3D cube highly summarized or highly abstract? Since the user will be placed within an IVE, it is imperative that the delays to the user's actions are kept at a minimal. For this reason DIVE-ON relies on the VCU to perform the data aggregation required (generating less detailed data). This effectively reduces network dependence to a minimum.

OLAP Operations Implemented

Mining a hierarchical multidimensional wealth of specific data requires the implementation of a set of operations; these operations are called OLAP operations. The **roll-up** operation performs aggregation on the specified dimension of the data cube effectively moving the view to a higher level of abstraction. The opposite operation is called **drill-down**. For example, if you are currently viewing the warehouse on the "month" level you can roll-up to the higher level of abstraction "year" or drill-down to the lower level day. Other typical OLAP operations that are very important in data mining include **slice** and **dice**. Slicing the data cube involves the selection of a specific value along one dimension. For example, by imposing the restriction (Z = t) on data points in 3D the result obtained would be a 2D plane or a "slice" at Z = t. The dice operation allows the view to be restricted along more than one dimension; thus creating a sub cube of the original data. DIVE-ON does support drill-down, roll-up, slice and dice operations within the IVE created by the VCU (Figure 5). The user is capable of inputting the parameters needed for each operation via a set of task-specific interaction techniques that are managed by the UIM. In the following section, we introduce how the VCU uses the data obtained from the DCC to generate the IVE in a way that facilitates the final phase of the KDD process, data mining.

Visual Cues and Measures

The data presented to the user in the IVE is encoded using graphical objects; these are the objects that actually make up the rendered virtual world. The VCU views the three-dimensional cube it receives from the DCC as a three variable function. Each of the three data dimensions becomes associated with one of the three physical dimensions, namely X, Y and Z. Since each entry in the data cube is a structure containing two *measures* M_1 *and* M_2, the VCU simply plots the two functions M_1 (x, y, z) and M_2 (x, y, z) in ($\mathbf{R}^3$). Next we will present the meaning of these measures.

Assume that the data warehouse under analysis is built around the theme "dollars sold" (Figure 4). An OLAP decision support person is not primarily interested in the fact that during the year **t** the total sale of product **p** at store **s** was $100,000.00; it is the *context* that this measure occurs within is what is important. The VCU expresses this context to the user in VR by associating these measures with visual cues. The first cue we use is **size**, which is associated with the measure M_1 (dollars sold). After normalization, M_1 (x_t, y_p, z_s) is used to render a cube (or a sphere) of that length (or radius) centered at position (x_t, y_p, z_s), for some **t**, **p**, and **s** within the data range. Using this criteria, one can instantly conclude from Figure (3b) that the total sales for location A, cube (t, p, A), are double that of location B, cube (t, p, B) without the need to view any numeric data.

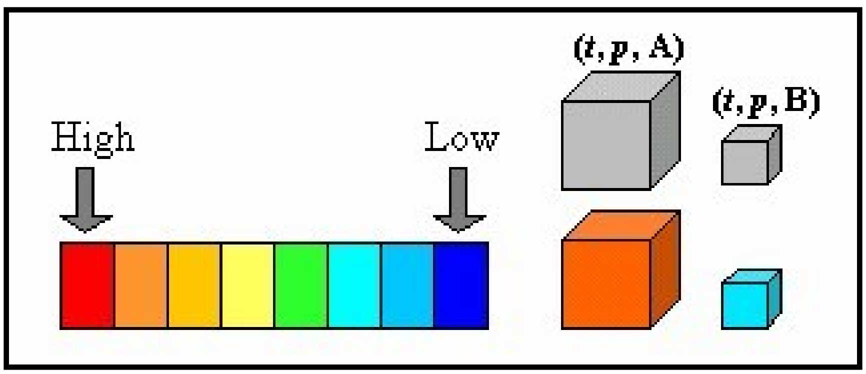

Figure 3: (a) Color pallet (b) One cue (c) Two cues

The second cue used is the object's **color**. An 8-color palette, Figure 3a, is chosen and the range of normalized values of the measure are discretized and mapped to the pallet. The red side is used to indicate "high" values of the associated concept and that value decreases from left to right where the blue is made to represent the "low" end. Using color to encode data mining results had shown significant results. As discussed earlier, each dimension is associated with a concept hierarchy that further describes that dimension. If, according to some *interestingness measure*, an anomaly does occur at a low level of abstraction, DIVE-ON can quickly and effectively pinpoint that result which may be "buried" down on a low aggregation level. This can be accomplished by employing the color cue. For example, at the lowest level of aggregation (high granularity) color can be used to represent the deviation from the mean along one of the dimensions. This is particularly useful for market fluctuation analysis. Formally, this is presented by the following equation:

$$M2\ (xi, yj, zk) = (M1\ (xi, yj, zk) - U_t)$$

Where U_t is the population mean along the time dimension.

When the roll-up operation along the time dimension from "day" to "month" is activated, the second measure (M_2) assumes a different role. In the new "rolled-up" view, the M_2 value for a month object is the maximum M_2 found in all the days it aggregates. Just to demonstrate the effectiveness of this approach, consider the example in Figure 3b where the objects represent annual sales for a given product in a given location. After adding color as a second visual cue to Figure 3b, a quick inspection of the two resulting objects (Figure 3c) reveals important and possibly hidden information. Using the pallet in Figure 3a and measure M_2 as above, although the annual sales for location **A** are double that for location **B** one of the months has a great deviation from the rest of the year. This will entice the analyst to *pick* that (t, p, A) object in an attempt to understand the reason. Alternatively, the user may be interested in knowing that a great stability dominates that product category at **B** (interaction is discussed in the next section).

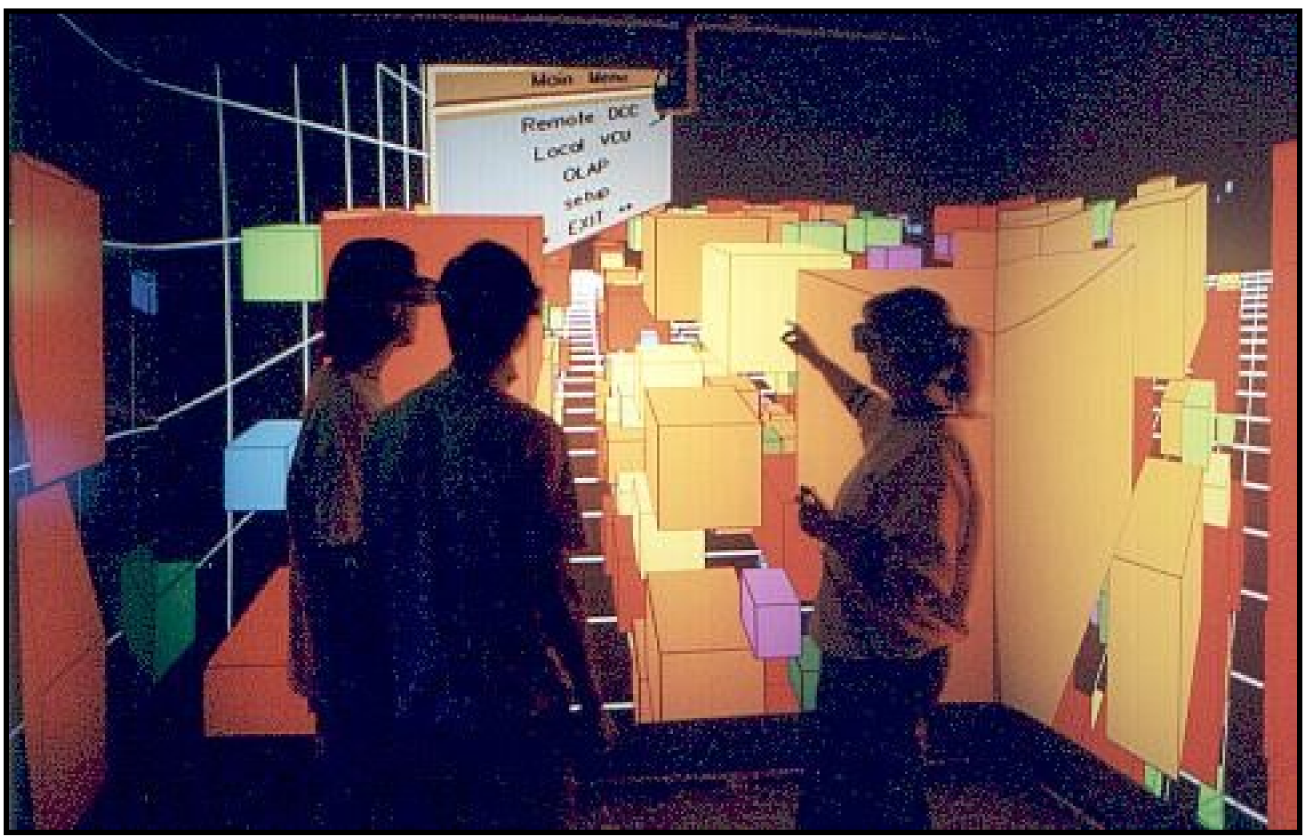

Figure 4: A team of immersed users discussing the "dollars sold" data cube. X-axis: "Product" Y-axis (front): "Time" Z-axis (up): "location"

Figure 4 presents the IVE created by rendering cubes that embody the above-described use of visual cues. The X-axis (left to right) is made to represent the "product" dimension. The axis pointing in the direction perpendicular to the picture is the "time" dimension (Y) while Z represents "location". The floating 3D menu appearing in the picture is discussed at a later section.

So far our discussion has been focused on cubes as the geometric objects that embody the presented information. DIVE-ON is also capable of creating an IVE based on a *spherical presentation* of the data (Figure 6). In this mode, each point is presented with a sphere with the size and color as described above. The reason for providing this mode is due to the fact that spheres, while capable of presenting the same amount of information as cubes, occlude fewer objects. It is possible to "see" a smaller sphere behind a larger one; this is not possible when dealing with cubes; however, it is important to point out that rendering spheres is computationally much more expensive than rendering cubes. To create a 3D sphere the system must perform light source simulations, normal vector calculations, material specification and shade rendering. None of these calculations is required dealing with cubes since rendering polygons is one of the very basic operations in graphics hardware.

View Point Manipulation

Pauline Baker [2] provides a generic framework for developing a VR application for the purpose of data exploration. Her work describes the characteristics needed in such applications to maximize the sense of reality and was used to create appropriate views of the generated data in our system. DIVE-ON manipulates the viewpoint to produce two distinct natural views of the IVE representing the data. To simulate normal everyday experiences, the IVE is constructed from the point of view of the user as if they were at the center of the virtual world. This effectively creates an **egocentric** frame of reference (Figure 4). The second frame of reference is **exocentric**, which is made available to provide the user with means of extracting themselves out of the virtual world for an out-of-world viewpoint (Figure 5). Egocentric views are essential for local data exploration where the user can examine the relationships that exist between consecutive data items and can access all available attributes that pertain to a specific object. Conversely, the exocentric approach enables the user to examine and detect global patterns by looking at the data from an "outside" viewpoint.

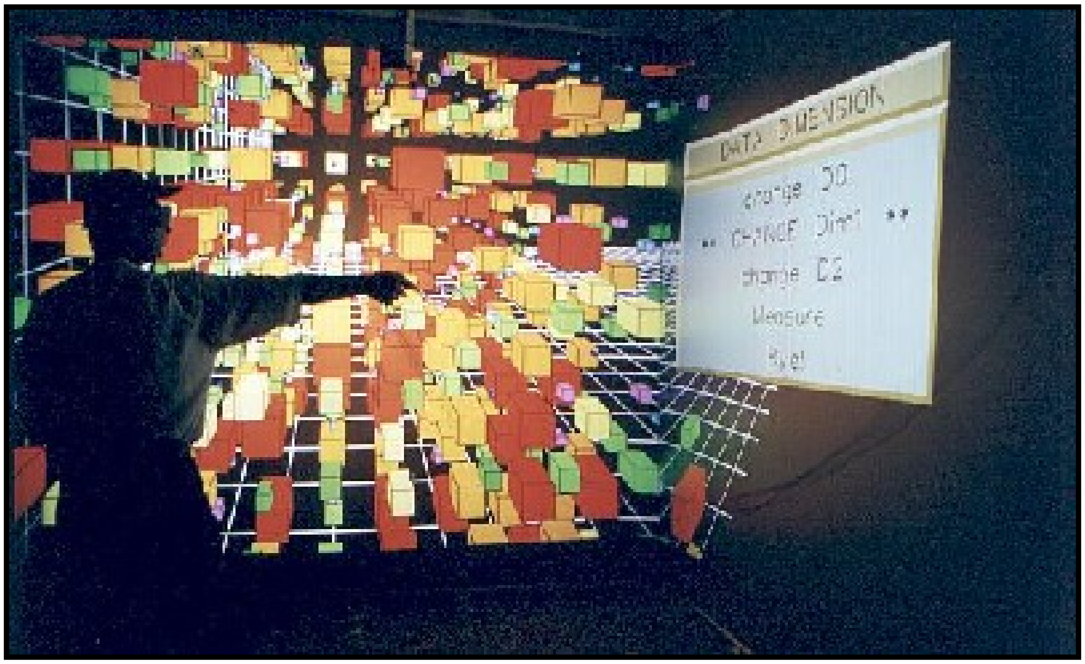

Figure 5: Exocentric viewpoint (User performing OLAP)

UIM: User Interface Manager

The UIM is the DIVE-ON component that handles all aspects of human-computer reaction. All tracker signals feed into the VCU to maintain the IVE and then passed to the UIM for inspection. Constant update of the user's location is necessary to determine the initial location of the floating menu, the active menu number and the current menu choices. Due to the lack of similar applications that can be studied while designing our system, volunteers helped asses the effectiveness of the UIM. Presentation, navigation, OLAP operations and the over all spatial knowledge acquisition were evaluated by experimentation. The initial design of the UIM facilitated the system interaction through a set of widgets that appeared at zero parallax (flat with the screen) so that they are always easily accessible in a familiar location. After the initial testing this was abandoned for reasons including:

- These widgets *occluded* a great deal of valuable screen space effectively reducing the amount of data visible.
- Having these menus (or widgets) appear in such fashion instantly changes the frame of reference from egocentric to exocentric effectively breaking the user's sense of presence in the IVE.

In almost every IVE application the essence of computer-human interaction can be categorized as *object manipulation*, *viewpoint manipulation*, or *application control* [8]. The reason for this taxonomy is the fact that simulating reality is all about simulating the changing views around us along with the ability to interact with the objects that make up these views. As a result, an interface that is highly transparent must use the human sensorimotor system to its advantage by providing the proper visual feedback to the users' motion. Understanding these issues helped us provide an interface that requires little or no instructions; however, it should be clear that the user is required to have an understanding of the data source and OLAP operations. All the interactive capabilities of DIVE-ON have been grouped in adherence with the above categorization. The users chosen in our experiments were people that are familiar with the architecture of the data warehouse being viewed and with the terminology and methodologies of data mining. The three interaction categories that are managed by the UIM are discussed next.

Figure 6: A user pointing the direction of flight (Spherical presentation) Some small data items would have been totally occluded using cubes.

System-Based Interaction

The system-based interaction refers to the application control needed to instruct the DCC and the VCU regarding what and how data is viewed. During initiation, the system sends a message to the DCC requesting the set of all N available dimensions in the data warehouse. The corresponding XML document is received by the VCU and the user is presented with a list from which they specify the three data dimensions to be visualized (one for each of X, Y and Z). At that point the DCC constructs the corresponding 3D data cube at the lowest level of abstraction and sends it to the VCU along with another XML document that contains the corresponding three concept hierarchies. Our primary concern in the above design is that OLAP operations be performed locally, within the VCU, to minimize the use of a possibly congested network. All system-based interaction is provided through the use of 3D floating menu hierarchy (Figures 4, 5), which is initiated by pressing the first button on the hand-held tracker. The UIM implements these menus with a 6 degrees of freedom (**6-DOF**) to allow them to flow freely in the VR in total sync with the user's hand. The orientation data structure (quaternion) stream transmitted from the hand held tracker (T2 in Figure 1) was used to rotate the menu surface so that its normal is constantly facing the user.

Aggregate-Based Interaction

At the very top of the general user goals during the visual exploration of data is to identify what they are looking at and to locate what they are after [9]. Based on this result the aggregate-based interaction has focused primarily on instantly providing the *lineage* associated with each visual cue for any given object. As discussed earlier, the size and color of every object is the result of normalized measures in the data warehouse. Since normalization is irreversible, it is important that the system maintain the original data used before rendering. The location of the hand-held tracker is obtained and used to draw a pointer in the IVE. When the pointer is placed "close" to an object, the second button will activate a small text panel that displays all information pertinent to that particular aggregate. This information is sufficient to identify all aspects of that particular aggregate, which include the current level of abstraction along each of the three dimensions and the actual value of the first and second measure. The text panel pops as a 3D box (Figure 5) that is not occluded by any other object and made to point to the particular object being selected.

Environment-Based Interaction

This type of interaction is responsible for the viewpoint manipulation and aids in building and using a cognitive map of the IVE. As seen above, the ability to locate a given aggregate within the environment is the second most sought after operation in VR data exploration. Providing effective navigational means is essential since, depending on the aggregation level, only a fraction of the data may fall within the clipping planes. The third button of the pointer is used for navigation and location control. Pressing this button provides the user with two movement options, *specified coordinate* movement and *specified trajectory* movement. The first mode provides the user with a planar cellular map of the X, Y or Z planes indicating the cell that includes the user's current location. The pointer can be then used to point at a new destination cell. With the second mode, the user points to the direction that they want to "**fly**" and based on the pointers trajectory, the image is transformed to simulate the sensation of flying through the environment. The *flight speed* is controlled by the distance between the user's hand and their head; thus, stretching the arm will result in a faster transition and bringing it close to the body will slow down the transition (Figure 6).

DISCUSSION AND FUTURE WORK

For a virtual reality system to be effective and well received by the user a great deal of emphasis has to be placed on the "reality" factor in virtual reality. As the user walks around within the walls of the CAVE the degree of realism of the projected image translations is directly proportional to the system's overall scene rendering speed. As the amount of data present in a particular view increase so does the number of polygons needed to render a scene, which in turn significantly cripples the system's ability to produce smooth, realistic image transformations. We are hence faced with *scalability vs. reality* trade-off. For data warehouse and data mining visualization systems, handling large volumes of aggregates in single views should be expected and not avoided. The way that DIVE-ON handles this problem is by creating a unique spatial data structure that is suited for *both* hierarchical volume decomposition and hierarchical data aggregation.

DIVE-ON creates a virtual world inhabited with geometric objects (spheres or cubes) which use color and size to tell something. We would like to examine the possibility of increasing the number of data mining measures presented by introducing more than one type of geometric objects. For example, a pyramid that points upwards could be used to indicate the existence of monotonic increase somewhere at a lower level, which is particularly useful in market analysis studies. It also important to experiment with audible cues in a similar fashion that we have used the visual encoding of information.

REFERENCES

1

Agrawal, S. et al "On the Computation of Multidimensional Aggregates," Proc. of VLDB Conference, 1996.Chaudhuri, S., and Umeshwar, D., "An Overview of Data Warehousing and OLAP Technology," Proc. ACM SIGMOD Record, Mar.1997.

2

Baker, M. P., "Human Factors in Virtual Environments for the Visual Analysis of Scientific Data," NCSA Publications: National Center for Supercomputer Applications

3

DeFanti, T. A., Cruz-Neira, C., and Sandin, D. J., "Surround-Screen projection-Based Virtual Reality: The Design and Implementation of the CAVE," Proceedings of SIGGRAPH, 1993/ ACM. http://www.evl.uic.edu/EVL/VR/systems.shtml.

4

Gary, J., Chaudhuri, S., Bosworth, A., Layman, A., Reichart, D., and Venkatrao, M., "Data Cube: A Relational Aggregation Operator Generalizing Group-by, Cross-Tab, and Sub-Totals," Proc. of the Twelfth IEEE International Conference on Data Engineering, Feb. 1996: 152-159

5

Han, J., and Kamber, M., "Data Mining: Concepts and Techniques," Morgan Kaufmann Publishers, 2000.

6

Inmon, W. H., "DATA WAREHOUSE - A PERSPECTIVE OF DATA OVER TIME," 370/390 Data Base Management 'Feb 1992

7

Green, M. and Shaw, C. develop MR-Toolkit at the University of Alberta: http://www.cs.ualberta.ca/~graphics/MRToolkit.html

8

Hand, C., "A Survey of 3D Interaction Techniques," Computer Graphics Forum, Dec97, 16(5): 269-281.

9

Wehrend, S., and Lewis, C., "A Problem Oriented Classification of Visual Techniques," proc. of IEEE Visualization '90: 139-143.

10

Extensible Markup Language (XML): http://www.w3.org/XML/

11

Survey.com is an eResearch company: http://www.survey.com/

12

Common Request Object Broker Architecture (CORBA): http://www.corba.org/

13

Simple Object Access Protocol (SOAP): http://www.w3.org/TR/SOAP/

Biography and Acknowledgments

Ayman Ammoura (ayman@cs.ualberta.ca) is currently a graduate student at the University of Alberta. His research interests include mining databases, visualization of large data sets and computer vision. The DIVE-ON project is conducted under the supervision of Dr. Osmar Zaiane (zaiane@cs.ualberta.ca).

The main idea behind DIVE-ON came from Dr. Zaiane. The author would like to thank Marc Perron (perron@cs.ualberta.ca) for his roll in implementing the DCC/VCU protocol simulation using CORBA. Dr. Marc Green (mark@cs.ualberta.ca) and Lloyd White (lloyd@cs.ualberta.ca) provided me with a great deal of documentation, examples and technical support to implement the VCU within the CAVE environment.