# A Framework for the Intelligent Visualization of Large Time-Dependent Flow Datasets in Medical VR Systems

Damon S.M. Liu and Walter J. Karplus
Computer Science Department
University of California,
Los Angeles
{damon, karplus}@cs.ucla.edu

Daniel J. Valentino
Dept. of Radiological Sciences
University of California,
Los Angeles
djv@ucla.edu

*May 28, 1997*

Abstract

Unsteady (time-varying) blood flow visualization motivates much of the research effort recently undertaken in Virtual Reality (VR) technology for medical applications. With continuously increasing resources for computing, simulation, and data acquisition, medical scientists are able to produce extremely large multidimensional and multivariate datasets with improving speed and accuracy. To ensure numerically stability, most simulations are computed using small time stepsize such that only a very small fraction of the total data change their values from one step to the next. The large quantities of simulated data are often stored in archives or databases on disk and moved to main memory during visualization.

We are designing and developing a framework that supports an interactive exploration environment tailored to the particular needs of brain aneurysm specialists and directly assists them in accomplishing high-level tasks. We study a scheme that can exploit the spatial-temporal properties in simulated dataset to control the reconstruction fidelity based on the areas or features of interest. The scheme may automate and highlight the extraction of fluid topology where there are abrupt changes or local extremes, offering the possibility of investigating the position and illumination of emergency or disaster situations in an environment under a variety of conditions. Besides, this scheme should provide adequate support to offer substantial data reduction to permit local storage, and achieve considerable speedup in graphics rendering.

The proposed framework defines a software architecture that needs to support computing and access to time-dependent data resident on a hierarchical storage devices, including a large-scale RAID (Redundant Arrays of Inexpensive Disks) type storage server. The software architecture should scale and make transparent the location of data while satisfying run-time performance constraints. This architecture should also support exploration across heterogeneous hardware platforms. The UCLA's Virtual Aneurysm and Virtual World Data Server projects serve as the front-end and back-end application testbeds, respectively.

# 1. Introduction

## 1.1 Virtual Reality in Medicine

In recent years, increased computer usage in clinical medicine and biomedical research has changed the way health care is delivered. In particular, the use of VR technology impacts a number of distinct areas such as [15]:

- Teaching students human anatomy and pathology.
- Surgical procedure training for new surgeons.
- Surgical planning of complex treatments.
- Navigational and information aids during surgery.
- Predicting operational outcomes.

VR-based systems in medicine represents a paradigm shift in the way that will allow surgeons to evaluate and practice new technologies in a simulator before using them on patients [2].

At UCLA, researchers are currently building a VR simulation environment to help the neurosurgeon, radiologist, or vascular system specialist plan treatment of brain aneurysms, which are a weakening of the blood vessel walls that can expand like a balloon. Treating aneurysms with implantable coils is designed to seal this expansion to prevent them from bursting [8] (see Figure 1).

## 1.2 Virtual Reality for Aneurysms

### 1.2.1 Motivation

Aneurysm surgery remains dangerous because surgeons have limited knowledge of blood flow patterns and complex anatomy of aneurysms. Part of the problem physicians face with is to determine if the aneurysm is suitable for a certain surgical technique. For example, if it is too large, in a different location, or oddly shaped, there may be difficulties. Techniques are often learned in a mentoring environment with its inherent risks to patients. Accordingly, scientific simulation/visualization of time-varying aneurysm anatomy, pressure, and flow at any point in the vascular system is one example of how VR can help medical specialists roam the complex dataset, evaluate the effects on sealing, blood flow, and pressure of various surgical techniques.

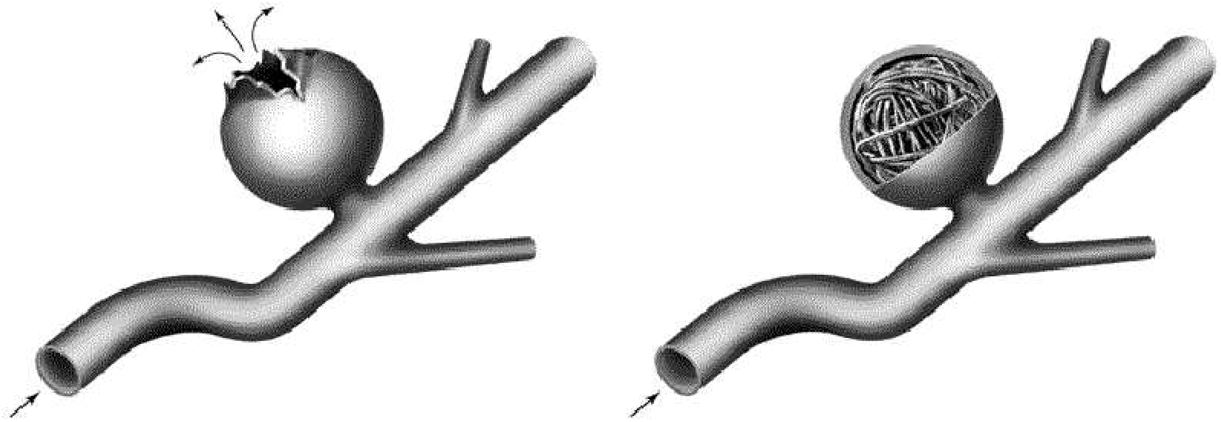

Figure 1. Aneurysm ruptured (left) and treatment of implantable coils (right). (Illustrations provided courtesy of Target Therapeutics (Fremont, CA) and Dr. Gary Duckwiler, Dept. of Radiological Sciences, UCLA.)

### 1.2.2 The prototype

The challenging effort for UCLA's Virtual Aneurysm research is to develop a virtual environment which permits a physician to move around and into an aneurysm, interactively navigate to explore its complex computer simulated fluid dynamics within the vascular system using VR and scientific visualization techniques.

In the prototype, output from blood flow simulation is velocity and pressure values at many nodal points within each element and at discrete times as the heart goes through its pumping cycle. The Nekton package [12] uses a spectral elements method which returns a high order solution over each large element within the domain. This flow information is entered into an object-oriented database for access by the run-time visualization software. The flow database is organized into tetrahedral cells which contain linear fields with the raw data, providing rapid indexing and interpolation of flow parameters within the computational mesh at multiple timesteps.

However, it requires that the flow database is main memory resident in order to guarantee delivery of information with low-latency response times [6]. Typically, one frame may require computations at over a thousand flow points.

## 1.3 Trends in Visualization

The increased capabilities of workstations have been used to dramatically increase the size and complexity of numerical simulations of fluid flow [9]. As the size of the simulations increases, the size of the flow solution datasets also increases and can result in immense datasets representing the physical characteristics of a flow field. The greatest impediment to developing systems for visualization of 3D fluid flow variables is the size of the data that constrains interactive response time. We expect that with a dramatically increasing rate in the amount of data produced by different sources and more intensive computations required in evaluating those large datasets on the fly, the data management problem in interactive flow visualizations may be severe.

When faced with data quantity problems, it is tempting to wait for hardware improvements. True, processor and disk will be faster, and we will see workstations with more and more memory. It is clear, however, that the response to higher performance will be to make even greater demands on the system, as we wish the frame rate to be as high as possible and wish to have as many visualization tools as possible in a single virtual environment. Apparently larger physical memory alone is not sufficient to solve this sort of data management problem.

The tactics is to organize the storage devices (including cache, physical memory, magnetic disk, disk farm, and network of distributed or shared memory) as a hierarchy of memory structures. Store and access to those memories the data in a progressive fashion to limit the visualization problem in volume, allowing more timesteps or a smaller volume to be loaded at a time. A more ambitious scheme is to load volume of data only when required by a computation.

The solution is further complicated in a collaborative visualization environment where multiple users share large datasets, simultaneously view visualizations of the data from widely scattered locations [19].

## 2. Project Overview

This project is concerned with research and development of a software architecture that integrates the prototype Virtual Aneurysm simulator to support computing and visualization of time-varying datasets resident on a large-scale RAID type storage server [14] in a highly interactive environment. Figure 2 illustrates this flow visualization environment.

In the system we present an approach, based on user and task characteristics, that supports intelligent visualizations. The system will take into account the goals and capabilities of

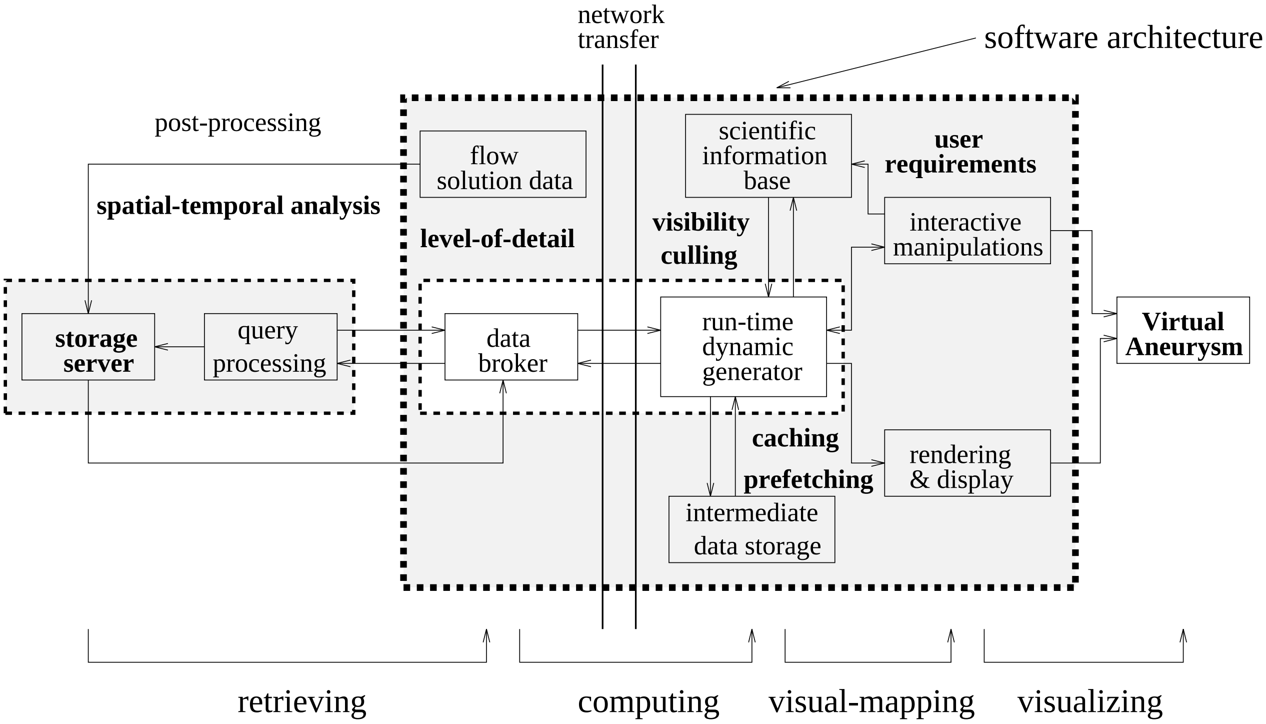


Figure 2. Schematic of the blood flow visualization environment.

the user, the requirements of the task, the type of data to be displayed, and other software and hardware considerations. Thereafter, the system will guide the user in the visualization process, assisting user with an effective environment to present only the relevant information to user in a timely manner. This will improve the quality of the visualization and decrease the time user is required to spend for exploratory data analysis.

## 2.1 The Scope

The kind of VR applications this proposed architecture applies to will focus on large-scale simulation and scientific visualization problems that arise in radiological sciences. To some extent, those VR applications share salient characteristics that differ from physics and urban VR systems:

- System dedicated to physicians that need to visualize blood flow dynamics within the vascular system for planning of brain aneurysm surgeries.
- Complex mathematical models and data types.
- Time-dependent multidimensional datasets.

- Exploiting a variety of visualization techniques.

While some of these characteristics may not be wholly new to systems in other application domains, they apply more strongly to the targeted medical speciality. This differentiates this research endeavors from many other approaches.

In order to fully exploit the advantages of applying the increased power of computing, storage, and VR to computational medicine, this software architecture will be designed and developed to achieve two primary objectives. They are detailed as follows.

1. **Visualizing large time-varying datasets**

   The kind of fluid flow simulations this project is mainly concerned with start with the numerical solutions of Navier-Stokes equations for the case of three dimensional time-varying flows. Usually a marching procedure in discretized time is used to solve these equations, resulting in flow solutions that consist of a series of single-time solutions, called timesteps. To ensure numerically stability, simulations are computed using small time stepsize such that only a very small fraction of the total data change their values from one step to the next. Adding the time dimension dramatically increases the dataset size, increasing storage requirements and computational complexity.

   In order to get maximum system performance, various aspects like data parallelism, network throughput, disk space, memory size, size of the data needs to be analyzed, typical operations to be performed on that data, all these strongly influence the proposed system performance and therefore should be taken into account.

   1) <u>Scalability:</u> The architecture should scale, support visualizations running on workstation clusters and massively parallel computers, and access to datasets resident on a hierarchical storage devices (disk, RAM, etc.).

   2) <u>Load data only as needed:</u> The architecture should minimize data storage, data movement, and network communication requirements. It should store only that part of the flow field that makes up the display objects which, in general, could be orders of magnitudes smaller than the entire flow field data. The architecture should transparently manage migration of data between memories while satisfying the interactive exploration constraints.

2. **Interactive environment tailored to user requirements**

   The proposed VR application differs from most other systems in that it is intended to be used by a small class of users for very specific purposes. In particular, radiologists will be primarily interested in visualizing MRI, CT or angiogram images in order to carry out clinical procedures to alleviate malfunctions in the brain, the heart or other bodily organs. These highly-trained physicians will seek optimal ways to adapt established procedures to the anatomy and pathology of a specific patient. The kind of interactions that such users will require from the VR system are therefore predictable to a considerable extent. There are several ways in which this predictability can be utilized to maximize the effectiveness of the VR system. The following are examples that apply specifically to the brain aneurysm problem.

   1) <u>Modeling the generic user:</u> All of the radiologists using the VR system can be expected to have similar objectives. For example, most radiologists intending to employ the endovascular procedure (described above), will be principally concerned with the danger of placing excessive stresses on the walls of the aneurysm while inserting the platinum coil. To this end, they will seek to find those points in time and space at which the sheer stresses reach a maximum. Data moved to the graphics engine can be sequenced appropriately so that the user is quickly led to those results.

   2) <u>Modeling a specific user:</u> Assume that a very small number of radiologists will use the VR system at a given installation. The manner and sequence in which they require data can be predicted in one of two ways (or in combination):

      a. Ask the user to describe his approach to the simulation, possibly with the aid of a questionnaire or a personal interview.

      b. Keep a record of the user's actions while using the system and develop an automatic working set strategy, this to make each successive use of the system by a specific user efficient than proceeding one.

### 2.2 The Requirements

The design and implementation of such a software architecture is a means of combining elements to produce a prototype of UCLA Medical School's Virtual Aneurysm simulator. However, because it is also part of the Computer Science Department's Virtual World

Data Server project development, this leads to another clear set of requirements for this architecture.

- The architecture should form well into the body of Virtual World Data Server and collaborate with other software and hardware subsystems in the same project.
- The architecture should support multiple levels of interaction so that it can easily adapt itself to offer useful but limited functionality to support other virtual world applications such as plasma physics and urban simulation systems.
- The architecture should allow work in a heterogeneous environment.
- The architecture should allow other virtual world applications to use their existing code and import or export data which they need to use in their visualizations.

### 2.3 The Approaches

1. **Octree construction**

   Octrees, like quadtrees, are hierarchical data structures based on decomposition of space. In quadtrees, space is recursively subdivided into four subregions [16]. Octrees are three-dimensional extensions of quadtrees, where space is recursively subdivided into eight subvolumes [11].

   1) Basics: We can spatially decompose the entire 3D volume into hexahedral and non-overlapping regions, called the voxel. Each voxel is atomic and represents the information with extent of variations about one single flow parameter which pierces that voxel. Voxels can be aggregated into super-voxels in an octree-like hierarchy. Use of octrees for controlling volume traversal is appropriate for sample data volumes common to scientific visualization. However, the storage and traversal of octrees themselves may add to the resource consumption of the algorithm, both in terms of time and space.

   2) Performance gains: The octree is a linked-list structure, so nonexistent branches do not need any storage. Information stored in voxels is often used to prune portions of computational space by marking the branch empty and not recursing down into further levels, thus avoiding useless data references and test for feature of interest. Encoding

flow physics into hierarchical data structures can often trade storage space for on-the-fly data computations which pose major obstacles for interactive investigation.

2. **Intelligent visualization tools**

Scientific visualization system is primarily for computational specialists who have datasets, particularly large and time-depending ones, to evaluate and analyze. Specialists' job is to concentrate on the science, not become visualization experts. They allocate very little time to learning a new system and demand actual results within minutes of exploring datasets on that system [3]. Furthermore, specialists often want to explore their data and simulation in unique ways. These characteristics present a good chance to make visualization tools tailored to specialist's tasks and optimized for specific visualization goals.

There are four major aspects of requirements for the design and development of these intelligent visualization tools:

- The tools require interfaces for specifying the goal.
- The tools require task analysis that decomposes the goal into a sequence of tasks. The visualization system should not violate task sequencing constrains.
- The tools require knowledge of the actions that users perform in the course of accomplishing the tasks. The tools should provide support for the appropriate user actions at the relevant times during task performance.
- The tools should alert the users to specific parameters that convey maximum understanding.

3. **Spatial-temporal analysis**

Flow simulations for radiology systems are often computed over time as the heart goes through its pumping cycle. To ensure numerically stability, simulations are computed using small time stepsize such that only a very small fraction of the total data change their values from one step to the next. The chief goal of such visualizations is to map datasets onto comprehensive visualization parameters such that radiologists can quickly find and comprehend the interesting features.

1) Data coherence: Raw data is generated from a simulation which might typically consists of hundreds of timesteps worth of information. Due to the spatial and temporal coherence, the faction of variation between data is often very small.

2) Data analysis: We can employ a mechanism to exploit the spatial-temporal properties in the simulated data of consecutive timesteps or neighboring locations, obtaining the positions, range of variations, and corresponding timestep values of changed data elements. It is hoped that exploitation of this time-frequency technique can impose a close approximation to unsteady flow topology. Instead of simply storing and retrieving the whole sequence of data, we can housekeep and explore this performance relevant information as auxiliary data to make time vs. space vs. presented quality tradeoffs.

4. **Data broker**

It is possible to analyze and predict in advance the volume of data that is required in the course of an interactive visualization session in order to perform the task actions. The data broker is an intelligent program that can be created to rearrange data so that the most likely requested data is prefetched and moved to local memory that makes it easier to navigate to.

Caching data into memory reduces the latency by eliminating the need to access a disk. However, because cache sizes are limited, data requested may not be present in the cache (a miss) and may still need to be fetched from disk. Prefetching data before it is required is one means for masking the remaining latency by increasing the probability that run-time visualization will find the needed data in memory rather than being compelled to go to disk.

In short, the function of the data broker is:

- To tailor the form and content of information according to its assessment of what the users need or wish to know.
- To retrieve appropriate data from backup memory and offer the information necessary to support and enhance the user actions.

## 3. Experiments

In this phase of project development, we are exploring the advantages and disadvantages of hierarchical data structures for scientific visualizations. In particular, we have explored the use of octrees in conjunction with the cutting plane visualization tools that are used to investigate the speed of blood flow within the aneurysm in any orientation (Figure 3).

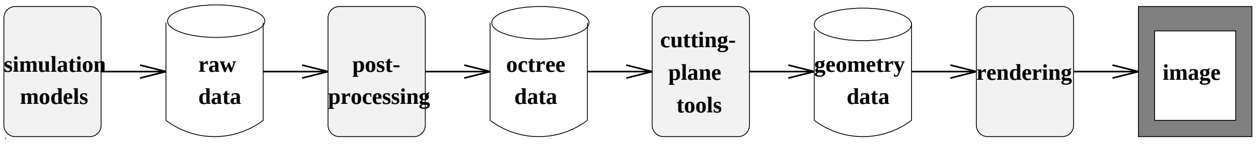


Figure 3. Data flows through the visualization pipeline.

The octrees were implemented in C++ on a Sun UltraSPARC workstation with 256 megabytes of main memory. The rendering software is Sense8's 3D/VR WorldToolKit which was used to control devices and manage graphical objects. UNIX's `gettimeofday()` utility was used to measure the time to build the trees and compute the flow values.

### 3.1 Description of Implementation

CFD (computational fluid dynamics) data are frequently visualized as scalar fields defined over grid points. Grid points are represented as a three dimensional array of real numbers representing the physical $x, y$, and $z$ locations. The spacing between grid points may be distinct. The integer space of indexes into the array is known as computational space, whereas the real space of the $x, y$, and $z$ locations is known as physical space [13]. Computational space can be easily divided by cutting each of the three dimensions in half, it is therefore much faster to build trees that divide the computational space as opposed to physical space. Typically, the number of grid points in our studies is several hundred or more.

The octree was built top down. The root node represents all of computational space and a procedure was applied recursively to generate the whole tree. Motivation for this development is a desire to represent homogeneous parts of a flow field. To transform the flow data into an octree, a criterion must be chosen for deciding that a field is homogeneous (i.e., uniform). Using this criterion, the flow field is successively subdivided into octants, suboctants, and so on until homogeneous volumes are obtained. The homogeneity criterion ultimately chosen to guide the subdivision process depends on the type of data represented.

- We consider the criterion as being the standard deviation of its flow values that is below a given threshold.
- If one associates with each leaf node the mean flow value of its volume, the resulting octree will then completely specify a piecewise approximation to the flow field where each homogeneous volume is represented by its mean.

Search using octree is then walked repeatedly with different parameters, the parameter is tested against the characterizing information stored in each node.

- If the test fails, the rest of the branch need not be examined. If it succeeds the tree walk continues.
- If a leaf node test succeeds, then the data represented is examined for feature of interest.

### 3.2 Space Requirements

The space requirement of an octree can be a serious issue in the design of a visualization system that will process large datasets. A full octree is one in which each node has exactly eight children. However, this is possible only if the dataset's resolution is the same power of two in each dimension, e.g., 4 x 4 x 4 (64) samples, 8 x 8 x 8 (512) samples, 32 x 32 x 32 (32,768) samples, etc. Full octrees offer the best ratio of the number of nodes to data sample points. For a dataset with a resolution of $n$ in each direction where $n$ is a power of 2 (see Figure 4),

$$\text{total number of nodes} = 1 + 8^1 + 8^2 + \cdots + 8^{log_2 n} = \frac{8^{log_2 n+1}-1}{8-1} \cong 8^{log_2 n}$$

Let us assume that each data value requires the same space as a pointer, and call this a "word". Usually a "word" is 4 bytes or 32 bits.

Consider a data volume whose $x, y$, and $z$ resolutions are 128 x 128 x 128, for a total of 2 million data points. An octree for this volume will have seven levels. If node requires 18 words each, a full octree will require 36 megawords or 144 megabytes. If there are 14 simulation timesteps, the storage need for the entire octrees is approximately 2 gigabytes. Most existing visualization systems assume much smaller quantities of data than this, i.e., they should load only part of the octrees to fit in RAM where a user gets all of data in one place for analysis. Usually the order and size of fetching component octrees are decided with respect to the user's viewpoint, view direction and speed of movement.

For our running experiments, we set up octrees over a 32 x 32 x 32 volume, which yields nearly 13 megabytes of space and this fits into physical memory. In comparison with octrees, the prototype tetrahedra database, which has resulted from a flow calculation on a 2,000 hexahedral mesh points, contains at least 16 megabytes dataset describing the fluid state at any particular instant.

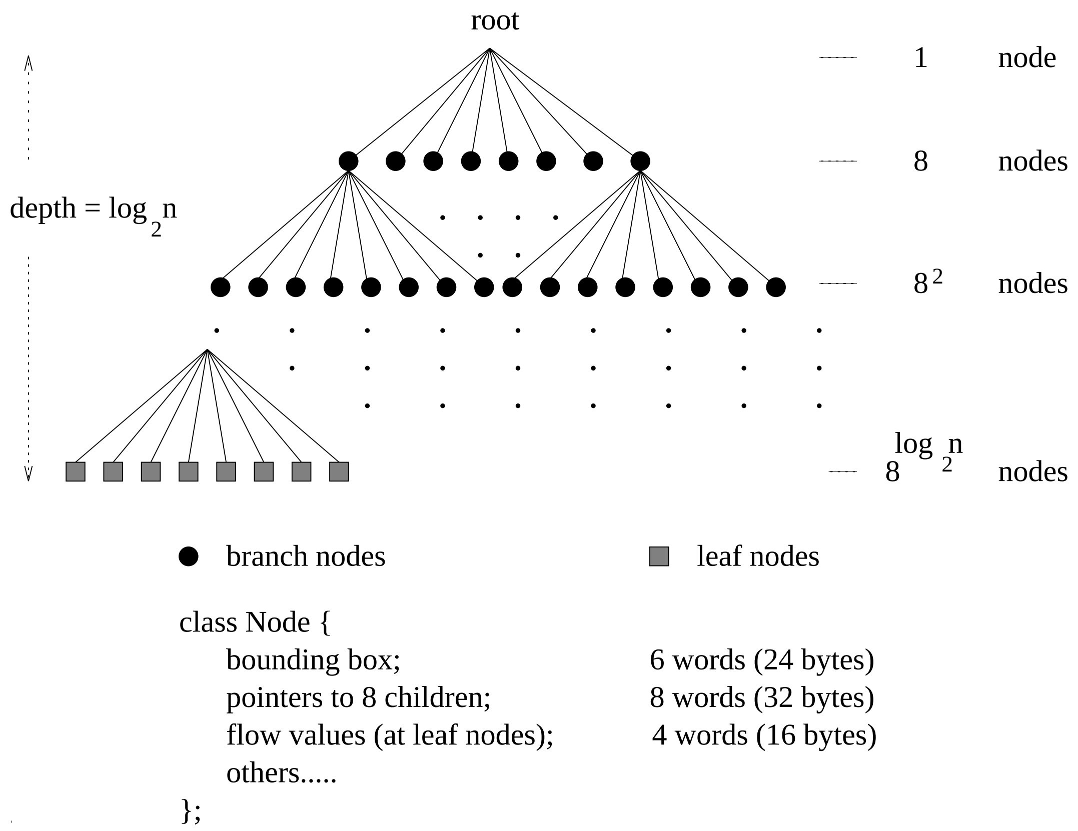


Figure 4. A full octree and the records it represents.

### 3.3 Timing Results

Octrees can be precomputed and stored for reuse, thus the octree creation time (which includes reading in the data and doing minor preliminary initialization, allocating memory for the octree, recursively traversing it downwards to establish pointers, and accumulating information on the return to the root node) is not of great concern. On the contrary, the calculation time (which involves traversal of regions of the octree and volume as dictated by characterizing information in the nodes) is worth a serious look.

If the dataset has a resolution of $n$ in each direction, then a careful analysis shows that a flow value can be computed in $O(log_2 n)$ time using octrees. In the case $n = 32$, we measured time and found it was approximately 35 microseconds. Compared with a spatial/temporal interpolation which almost always took 70 microseconds or more, our preliminary results indicate speedups in the range of 2 to 2.5 while still maintaining the accuracy. This factor will become even more significant on visualization techniques that involve even greater computational demands.

### 3.4 Miscellaneous Remarks

In order to use octrees to achieve speedup in calculation times, there must be:

- A fast, space efficient test to prune the search tree.
- Features of interest are somewhat sparse. They must be sparse so that much of the space can be pruned away.
- They need to apply the algorithm repeatedly over the same datasets with different parameters. Otherwise, the overhead of tree structure is not worthwhile.

Cutting planes, which always pass through a small subset of 3D space, fit these descriptions. Our studies show that use of octrees can yield substantial improvements in performance for cutting planes on flow datasets. These improvements will be even more significant if a careful caching algorithm is applied to reuse the results of earlier computations, then discard those results when they would not be referenced again, replacing the storage for others.

## 4. Related Work

Exploratory visualization systems have evolved over a period of time. However, most systems offer a fixed functionality and limited ability to process large volumes of data. Besides, many of them are machine-dependent and bundled in with their hardware platforms.

- Hibbard [7] describes the VIS-5D system designed for interactive visualization of unsteady 3D earth's atmosphere and ocean datasets. VIS-5D uses a five-dimensional array to store data in RAM. Data is organized as 2D arrays of 3D spatial grids. The 2D arrays are indexed by time and by model field (e.g., pressure or temperature).

- Bryson [1] uses virtual environment technology to examine time-dependent vector fields, although datasets must fit in RAM. They improved the system by distributing processing between a workstation and a mini-supercomputer to increase RAM, and thereby increasing program size.

- Haimes [5] describes the Visual3 software package used to visualize steady and unsteady flow data. A unique aspect of Visual3 is the dimensional windowing approach coupled with cursor mapping which allows efficient pointing in 3D space. Visual3 can run on a distributed architecture using multiple concurrent processors across the network. Server generates graphical objects as polygonal data and sends the data to the client. Rendering of the data is performed by the client on a graphics workstation.

- Grinstein [4] visualizes unsteady jet flows by animating the results computed from a number of instantaneous visualization techniques. The simulations are performed in a supercomputer and the visualization is performed on a workstation. However, animations are made only for two-dimensional calculations, not 3D data.

- Plot4D [18] visualizes 3D unsteady flows by using variables in 2D computational space with time as the third coordinate. Therefore several time slices of a computational plane from the 3D flow data can be visualized. However, they require a preprocessing of the input data such that a number of timesteps of flow data are stored in one input file.

- Lucas [10] describes the IBM Data Explore, an implementation of dataflow model that offers modules which perform filtering, mapping, and rendering tasks. Modules can be combined to achieve a desired visualization goal. Memory requirements become prohibitive and cause performance degradation when datasets are large.

- Senay [17] describes the Vista, a knowledge-based visualization system, that provides design support and guidance for visualization mapping process. Vista can automatically generate an effective visualization technique design for a given dataset. However, it does not consider the role of the user tasks in the design process.

## Acknowledgments

This work was performed for the research project "The Virtual World Data Server", which has been conducted by the UCLA's Computer Science Department and has been supported by a High Performance Computing and Communications (HPCC) Multidisciplinary Challenge Program Award from the National Science Foundation under grant IRI-9527178.

## References

[1] S. Bryson and C. Levit, "The virtual windtunnel: an environment for the exploration of three-dimensional unsteady flows," *Proc. Visualization '91,* pp. 17-24, San Diego, California, Oct. 1991.

[2] C. Chinnock, "Virtual reality in surgery and medicine," *Hospital Technology Feature Report,* vol. 13, no. 18, pp. 38-41, American Hospital Association, Dec. 1994.

[3] G.J. Edwards, "The design of a second generation visualization environment," *Visualization and Intelligent Design in Engineering and Architecture,* J.J. Connor, S. Hernandez, T.K.S. Murthy, and H. Power (Eds.), pp. 3-16, Computational Mechanics Publications, 1993.

[4] F. Grinstein, U. Obeysekare, and G. Patnaik, "Flow visualization as a basic tool to investigate the dynamics and topology of jets," *Proc. of Visualization '92,* pp. 164-170, Boston, Massachusetts, Oct. 1992.

[5] R. Haimes and M. Giles, "Visual3: interactive unsteady unstructured 3D visualization," *29th AIAA Aerospace Sciences Meeting and Exhibit,* AIAA 91-0794, Reno, Nevada, Jan. 1991.

[6] M.R. Harreld, D.J. Valentino, and W.J. Karplus, "The virtual aneurysm: VR in endovascular therapy," *Medicine Meets Virtual Reality: 4,* Jan. 1996.

[7] B. Hibbard and D. Santek, "The VIS-5D system for easy interactive visualization," *Proc. of Visualization '90,* pp. 28-35, San Francisco, California, Oct. 1990.

[8] W.J. Karplus and M.R. Harreld, "The role of virtual environments in clinical medicine: scientific visualization," *Proc. CISS First Joint Conference of International Simulation Societies,* pp. 13-17, Zurich, Switzerland, Aug. 1994.

[9] T. Lasinski, P. Buning, D. Choi, S. Rogers, G. Bancroft, and F. Merritt, "Flow visualization of CFD using graphics workstations," *Proc. American Institute of Aeronautics and Astronautics 8th Computational Fluid Dynamics Conf.,* AIAA Paper 87-1180, pp. 814-820, Honolulu, Hawaii, June 1987.

[10] B. Lucas, G. Abram, N. Collins, D. Epstein, D. Gresh, and K. McAuliffe, "An architecture for a scientific visualization system," *Proc. Visualization '92,* pp. 107-114, Boston, Massachusetts, Oct. 1992.

[11] D.J. Meagher, "Geometric modeling using octree encoding," *Computer Graphics and Image Processing,* vol. 19, no. 2, pp. 129-147, June 1982.

[12] "Nekton: spectral elements based computational fluid dynamics software," URL: http://www.cray.com/PUBLIC/APPS/DAS/CODES/NEKTON.html.

[13] G.M. Nielson, T.A. Foley, B. Hamann, and D. Lane, "Visualizing and modeling scattered multivariate data," *IEEE Computer Graphics and Applications,* vol. 11, no. 3, pp. 47-55, May 1991.

[14] D.A. Patterson, P.M. Chen, G. Gibson, and R.H. Katz, "Introduction to Redundant Arrays of Inexpensive Disks (RAID)," *Proc. IEEE COMPCON Spring '89,* pp. 112-117, IEEE Computer Society Press, 1989.

[15] J.M. Rosen, H. Soltanian, R.J. Redett, and D.R. Laub, "Evolution of virtual reality," *IEEE Engineering in Medicine and Biology Magazine,* vol. 15, no. 2, pp. 16-22, Mar.-Apr. 1996.

[16] H. Samet, "The quadtree and related hierarchical data structures," *Computing Surveys,* vol. 16, no. 2, pp. 186-260, June 1984.

[17] H. Senay and E. Ignatius, "A knowledge-based system for visualization design," *IEEE Computer Graphics and Applications,* vol. 14, no. 6, pp. 36-47, Nov. 1994.

[18] M. Smith, W. van Dalsem, F. Dougherty, and P. Buning, "Analysis and visualization of complex unsteady three-dimensional flows," *27th AIAA Aerospace Sciences Meeting and Exhibit,* AIAA 89-0139, Reno, Nevada, Jan. 1989.

[19] M.J. Gerald-Yamasaki, "Interactive and cooperative visualization of unsteady fluid flow," NASA Ames Research Center, NAS System Division, Applied Research Branch technical report RNR-92-018, Mar. 1992.