

1-1-1998

Applying virtual reality to computational fluid dynamics post-processing

Sasikumar Sasikumar
Iowa State University

Follow this and additional works at: <https://lib.dr.iastate.edu/rtd>

 Part of the [Mechanical Engineering Commons](#)

Recommended Citation

Sasikumar, Sasikumar, "Applying virtual reality to computational fluid dynamics post-processing" (1998). *Retrospective Theses and Dissertations*. 17867.
<https://lib.dr.iastate.edu/rtd/17867>

This Thesis is brought to you for free and open access by the Iowa State University Capstones, Theses and Dissertations at Iowa State University Digital Repository. It has been accepted for inclusion in Retrospective Theses and Dissertations by an authorized administrator of Iowa State University Digital Repository. For more information, please contact digirep@iastate.edu.

Applying virtual reality to computational fluid dynamics post-processing



by

Sasikumar Kutti

A thesis submitted to the graduate faculty
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Major: Mechanical Engineering

Major Professor: Judy M. Vance

Iowa State University

Ames, Iowa

1998

Graduate College

Graduate College
Iowa State University

This is to certify that the Master's thesis of
Sasikumar Kutti
has met the thesis requirements of Iowa State University

Signatures have been redacted for privacy

“Ultimate pleasure is not obtained by gaining knowledge but by having the wisdom to share it”

-Anonymous

TABLE OF CONTENTS

LIST OF FIGURES	vi
ACKNOWLEDGEMENTS.....	vii
CHAPTER 1: INTRODUCTION	1
CHAPTER 2: METHODS	4
Computational Fluid Dynamics.....	4
Structured grid.....	5
Unstructured grid.....	6
Multiblock structured grid.....	7
Visualization	8
Virtual Reality.....	9
Description of VR systems.....	10
CHAPTER 3: MOTIVATION AND OBJECTIVE.....	16
Software Tools.....	16
Data storage.....	17
Ability to display more than one parameter	18
Ability to display vector data restricted to a cutting plane	19

Additional data generation.....	20
Viewing and Interacting with Data	21
Viewing tasks	21
Interaction tasks.....	21
Stereoscopic viewing.....	22
CHAPTER 4: VR-CFD SOFTWARE.....	24
General Features.....	24
User interface.....	24
Wand pointer	27
Navigation	28
Handling multiblocked data.....	28
Streamlines	30
Cutting planes	31
Iso-surfaces.....	34
Spatial sweeps.....	34
Legend	34
Changing the scalar and vector data.....	35
Application of VR-CFD.....	35
CHAPTER 5: CONCLUSIONS AND RECOMMENDATIONS	38
REFERENCES	41

LIST OF FIGURES

Figure 1 Structured grid and unstructured grid	6
Figure 2 Cyberglove [9]	11
Figure 3 Head-mounted display [18]	13
Figure 4 BOOM [2].....	13
Figure 5 C2.....	14
Figure 6 Streamtubes showing variation in diameter [29]	18
Figure 7 Vectors projected onto a cutting plane	20
Figure 8 Main menu	25
Figure 9 Menu structure	26
Figure 10 Rake	31
Figure 11 A person placing a cutting plane in a flow field	32
Figure 12 Cutting planes warped with velocity values [29]	33
Figure 13 Multiblock grid showing an iso-surface	35
Figure 14 Multiblock data with a cutting plane	36
Figure 15 Multiblock data with some streamlines	37

ACKNOWLEDGEMENTS

I cannot thank my major professor, Dr. Judy Vance, enough. Her patience, guidance, and support are very much appreciated. The way she stood as an example whom I could look up to made me a much better being. Thanks to Dr. Richard Hindman for making me become aware of the insatiable quests people have. Thanks to Dr. Richard Pletcher for his efforts in correcting this thesis and for having students like Dr. Todd Simons without whose invaluable guidance this work would not have been possible. Thanks to Abinesh Puthenpurackal and Rajni Chidambaram for taking the pains to proof read this thesis. Thanks to all my colleagues at Iowa Center for Emerging Manufacturing Technology and in the Department of Mechanical Engineering for their support. Thanks also to all other people who let me do at least few small things on my own, otherwise my debts will reach even higher levels.

CHAPTER 1: INTRODUCTION

Fluid mechanics is a physical science dealing with the action of fluids at rest or in motion, and with applications and devices in engineering using fluids [11]. Fluid mechanics can be subdivided into two major areas, fluid statics, which deals with fluids at rest, and fluid dynamics, concerned with fluids in motion. Fluid mechanics is basic to such diverse fields as aeronautics, chemical, civil and mechanical engineering, meteorology, naval architecture, oceanography, etc. For example, study of fluid flow around an airplane is vital in determining the stability of its structure in flight. Fuel efficiency in automobiles can be improved by reducing the drag created by airflow around them. Engineers also study the flow inside machinery such as turbines, compressors, and jet propellers in order to develop and improve the designs. In the process, engineers examine scalar parameters such as temperature, pressure, density etc. as well as vector parameters, such as velocity.

Fluid dynamic analysis can be performed using actual measurement of parameter values through experimentation or by theoretical computation. Computational fluid dynamics (CFD) is a specialized field involving the analysis of fluid flow using computers. Current improvements in the field of computers have facilitated a number of advancements in the computational fluid dynamics field. Almost all of the major work in

fluid flow analysis is currently performed by commercial computer applications, which provide a variety of analysis features.

Though numerous improvements have been made in this field with respect to computational capabilities, the benefits can be realized only if scientists are able to easily interpret the data produced by the applications. With the emergence of computer graphics, facilities for effectively visualizing the data have been developed. Adding interactive computer graphics to the fluid flow applications is helping scientists to understand the data better and giving them the power to reach engineering decisions faster.

Currently available software packages have incorporated many advanced features of computer graphics. The latest of such tools incorporates virtual reality (VR) techniques, which provide the users with the ability to view data using three-dimensional stereographic display devices and interact with the graphical entities using natural hand and head movements. One of the first VR applications to be developed to examine fluid flow, the "Virtual Windtunnel" was created by Bryson and Johan [3]. More recently Sterling Software Inc. built a VR interface to FAST [1], a CFD post processing application, primarily with the support of Ford Motor Co. [22]. These applications use VR devices like the head-mounted display (HMD) [18] and the BOOM [2] as visual display devices. Kohlmeyer [24] of Iowa State University created a VR application to examine CFD data in the C2, a CAVE-like [7] device. This was a significant improvement from the earlier implementations since the C2 has several advantages over other VR systems. Using all the above applications the users can place entities such as

streamlines and cutting planes inside a flow field and then interactively manipulate their positions in a three-dimensional space using natural hand movements. The research presented here extends Kohlmeyer's work by providing support for standard CFD analysis data formats and additional analysis features.

Chapter 2 presents an overview of the methods used in CFD data analysis and the need for implementation of VR in a CFD post processor. Chapter 3 presents a review of the currently available software and the motivation for this research. Chapter 4 presents discussions about the application developed. Chapter 5 presents conclusions and recommendations for future work.

CHAPTER 2: METHODS

This work combines the areas of computational fluid dynamics, visualization and virtual reality. This chapter presents short overviews of each area in order to provide necessary background information to facilitate understanding of the work presented in this thesis.

Computational Fluid Dynamics

Industrial fluid flow problems are often complex and finding their solutions using analytical methods is a challenging task. One approach is to use experimental results to construct a mathematical model. For instance, an engineer who is designing the exterior of a vehicle might want to find out whether the flow is laminar or turbulent. The vehicle model is placed in a wind tunnel and the velocity of the fluid is measured at discrete locations around the vehicle. The velocity data obtained are studied by the engineers to determine the type of flow. Velocities at locations other than the ones for which measurements were taken can be obtained by interpolation. In another situation, the goal could be to find out the temperature profile of the fluid around the vehicle. In this case temperature measurements are taken at discrete locations around the vehicle and studied.

Another approach is to use computational methods like finite-volume or finite-difference to predict parameter values in the flow fields. Pozrikidis [28] says “The central

goal of the finite-difference method is to generate the values of a function that relates unknown parameters to known parameters at the nodes of a coordinate grid that covers the domain of solution, at a sequence of discrete time levels separated by the constant or variable time step” (p. 559). Computers are used for carrying out the calculations involved. Once the discrete finite-difference solution has been computed, the values of the function between grid points and time levels are obtained by applying standard methods of interpolation, extrapolation, or approximation.

The coordinate grid defines the locations where solutions are obtained in the flow field. The main types of coordinate grids used in CFD problems include structured grid, unstructured grid and multiblock structured grid.

Structured grid

A structured grid is a dataset with regular topology and irregular geometry (Figure 1). The structured grid can be considered to have a corresponding parametric space in which the points are arranged regularly in three dimensions. For each point in the parametric space, there is a corresponding point in the real data set. The parametric space represents the topology of the points. Though the geometry is irregular, each grid point of a structured grid can be referenced using topological i-j-k coordinates. One or more grid points are said to form a cell when their combination results in a recognizable geometric shape. The composing cells of a structured grid are quadrilaterals (2D grids) or hexahedrons (3D grids). The structured grid may be of any configuration in which the cells do not overlap or self-intersect.

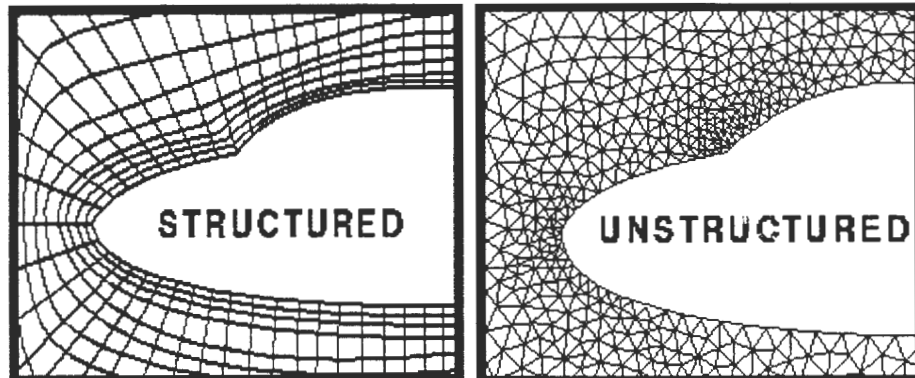


Figure 1. Structured grid and unstructured grid

The topology of the structured grid is represented by specifying the number of points in each dimension of the parameter space. The geometry is explicitly represented by maintaining an array of point coordinates. Structured grids are commonly found in finite-difference analysis. Typical applications include fluid flow, heat transfer and combustion.

Advantages of using structured grids are the simplicity of representation and their wide usage. All the commercial CFD analysis packages support this data set.

Unstructured grid

The most general form of dataset is the unstructured grid. The topology and geometry are completely unstructured. Cell types such as vertex, lines, hexahedrons, tetrahedrons etc. can be combined in arbitrary combinations in an unstructured grid and the layout is not uniform. Hence the topology of the cells ranges from 0D (vertex, polyvertex) to 3D (tetrahedron, hexahedron). The connectivity of the cells has to be specified in the data set and this increases the data storage requirement. Since this data type requires the most memory and computational resources, it is used only when necessary.

Unstructured grids are found in fields such as finite element analysis, computational geometry and CFD. The advantage of using unstructured grids is the ability to map complex flow fields using a variety of cells and the ability to position more cells in regions of interest as shown in Figure 1.

Multiblock structured grid

This type of grid combines the advantages of both the structured grid and unstructured grid. This is accomplished by dividing the flow field into smaller blocks, which are individually composed of structured grids. This allows the grid to be of different densities in different areas, yet remain structured. Using this method only the connectivity of the blocks is needed and not the connectivity of each grid point. This method was developed in order to take advantage of the faster computations involved in handling structured grids even if the flow field is complex. Algorithms are available to determine the connectivities of the blocks and therefore even this information need not be stored in the data file.

The CFD post-processing procedure can generally be divided into three steps:

- Pre-processing - Choosing the geometric locations (grid points) where the results are required and specifying the boundary conditions.
- Computation - Solving for the parameter values at each of the grid points.
- Post-processing - Examining the results.

This thesis explains the development of a program for post-processing CFD results in a virtual environment.

During post-processing, result data are examined in order to understand the problem. The aim of post-processing is to achieve comprehension and understanding. The process of studying and understanding any problem by examining the provided data is called cognition. Cognition includes attention, perception, memory, reasoning, judgment, imagining, thinking, and speech [11]. People gather data through the senses of sight, smell, touch, taste and hearing. The visual system, which provides the sense of sight, is one of the most advanced sensory systems in the body. More information is conveyed to the brain through the sense of sight than by any other means [11]. Hence converting the data into visual form is the one of the most effective ways of achieving comprehension.

Visualization

Transforming data into images in order to achieve comprehension of complex information is known as visualization. Creating a bar chart of a two-dimensional dataset is an example of visualization. Visualization of data is indispensable for business, demography, weather reporting, information management, etc. The process of visualizing scientific data is known as scientific visualization [29]. Visualization of data, obtained from problems involving CFD, finite element analysis, seismology, thermodynamics and many other branches of science and engineering, comes under definition of scientific visualization.

The ability to visualize complex data has greatly increased due to the advent of computer graphics. Computer graphics is the process of creating models of objects and their pictures using a computer [16]. The development of modern computer graphics began with Ivan Sutherland's Sketchpad [33] application. Computer graphics involves

creating line graphs, bar charts, pie charts, 3D surfaces, models, etc. from data stored in the computer. These images are immediately displayed in a standard display device. Due to the high speed of the process, data can be interactively modified and visualization results are instantly obtained. Combination of the computational and the graphical capabilities of the computer gives scientists the ability to analyze and study data much more effectively.

Current display techniques are two-dimensional and hence they present a challenge when analyzing three-dimensional data sets. Advanced three-dimensional stereographic display and interaction are provided by virtual reality systems.

Virtual Reality

“Virtual reality is a high-end user interface that involves real time simulation and interactions through multiple sensorial channels. These sensorial modalities are visual, auditory, tactile, smell, taste etc.” [11]

A typical virtual reality system has at least the following features

- Stereo optic three-dimensional graphical images - These are displayed in such a way that the users feel as if they are inside or immersed in that graphical environment. The images are changed, at a sufficiently high frequency, according to the change in the position and orientation of the viewer's head.
- Three-dimensional interaction - This is provided by some type of position tracker device. Users are able to navigate and ‘pick and place’ graphical objects using natural movement of their hands because of this feature.

Virtual reality was first demonstrated by the 'father' of computer graphics, Ivan Sutherland. He wrote about the 'Ultimate Display' [32] which included interactive graphics, force-feedback devices, audio, smell and taste. In 1968 [34] he built a head-mounted display that tracked the viewer and updated a graphics display to correctly reflect the new viewing position. Sutherland's system, called the Sword of Damocles, used two displays, one for each eye, which were visible from a pair of half-silvered mirrors. This provided the viewer with stereoscopic computer graphics images overlaid onto the real world.

Description of VR systems

Virtual reality systems are composed of input devices, which give information about the user to the computer, and output devices, which provide sensations to the user. Input devices include controllers and position trackers. Output devices include display and haptic devices used to provide forces in the virtual environment.

Controllers are devices which are used to send commands to the computer while the user is in the virtual environment. The equivalent desktop devices are the keyboards, mouse, etc. Some of the more common VR control devices include joysticks, wands and datagloves. The controllers are often coupled with the position trackers which indicate the position of the controller in the three-dimensional space.

Joysticks are hand-held devices often used in computer games. They contain switches, which send signals to the computer when pressed. These signals control actions in the VR simulation.

Datagloves, which are devices that are worn on the hand like gloves, send signals to the computer about gestures that the users make with their hands. There are two types of datagloves. Cybergloves [9] (Figure 2) contain strain gages at several locations on each finger and can sense the amount of bending in each finger. Using Cybergloves, full articulation of each finger can be sensed. Pinch gloves [27] send signals when contact is made between electrical contact patches, which are located at the tip of each finger and the palm. Datagloves allow natural gestures to be used to provide input, whereas the joysticks operate much like the desktop mouse except that 3D input is provided. Other devices such as foot pedals, steering wheels, etc. are also being used as control devices in many immersive simulation applications.

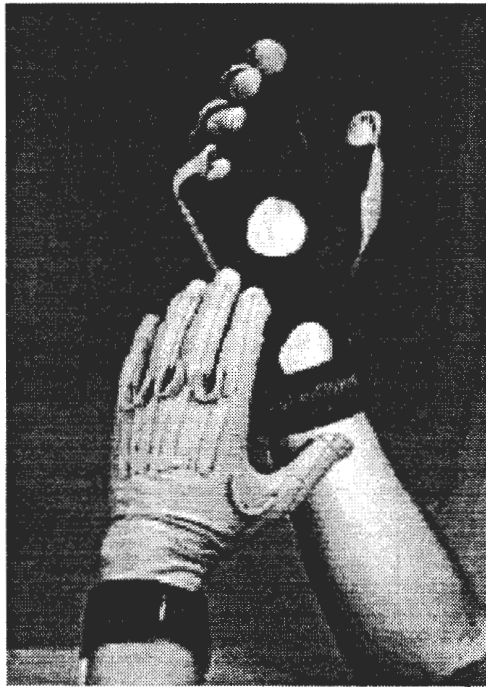


Figure 2. Cyberglove [9]

Position trackers are devices that are used to track the position and the orientation of the user's head, hand and other parts. Many technologies are available for position tracking. For example, the Ascension Flock of Birds [14] uses magnetic field tracking, the BOOM uses mechanical tracking and the Logitech 3D mouse [21] uses ultrasonic sound tracking. The position trackers in combination with the joysticks or datagloves are equivalent to the mouse and the trackball, but unlike their two-dimensional counterparts this combination has a three-dimensional field of operation. The mouse and the trackball provide both controller and position tracker functions in 2D applications.

The number of position trackers required can vary according to the requirements of the application. Typically, a VR system will require at least one tracker for head position and one for the input device position. The position and orientation information from the head tracker is used to compute the image that needs to be rendered based on the user's viewpoint. The information from the input device tracker is used to position a graphical representation of the device in the scene, which enables the user to point and pick objects.

Output devices include display and haptic devices. The VR display devices replace the monitors of the 2D systems. The Head Mounted Displays (HMD) (Figure 3), BOOM, and C2 are such display devices. The HMD is a device that has to be worn by the user over the head and the eyes. It has two small displays located in front of each eye. Each eye can see the image displayed by a corresponding display. The HMD sometimes has a coupled position-tracking device. Premier HMD's are manufactured by companies like n-Vision [18].



Figure 3. Head-mounted display [18]

One disadvantage of the HMD is that it is quite heavy and thus strains the user. The current technology has not allowed manufacture of a light-weight high resolution HMD. Further, the fixed focal length of the HMD produces eyestrain if worn over an extended period of time. Finally, since the user's view is confined to the HMD, communicating with others is restricted because only the person wearing the HMD can view the scene.

The BOOM is a head coupled display device with mechanical tracking. The BOOM consists of two small cathode ray tube projectors enclosed in a viewing device on one end of a rod, which is pivoted at the center and balanced by a counterweight at the other end (Figure 4). The BOOM can be easily held in one hand by the user and turned to view other parts of the virtual environment. Some disadvantages of the BOOM include the need to hold it with one hand leaving only one hand free for other inputs, and the restricted area available for movement.

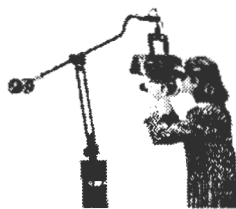


Figure 4. BOOM [2]

The C2 (Figure 5), a CAVE[7]-like virtual reality device, consists of a room that has four projection screens as walls. Stereo images computed for each eye are projected alternately on these walls at a frequency of 48 Hz per eye. Stereo vision is obtained by wearing CrystalEyes shutter glasses [8], which are synchronized with the display of the images using an infra-red transmitter, so that only one image can be seen by an eye at a time. Though the walls are perpendicular to each other, the perspective computation is done in such a way so as to ensure that the user will perceive a continuous image. These devices offer high resolution, full immersion, a large work area, and a wide field of view. Several people can interact in the C2 simultaneously and hence communication comes easily to users. Further, since the users can see the real world as well the computer generated virtual environment, real objects can be brought into the C2 and placed in the virtual environment.

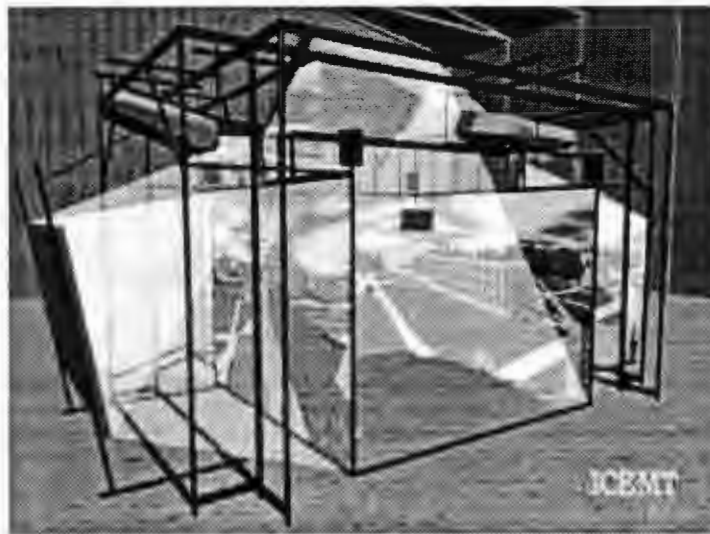


Figure 5. C2

The program developed in this research uses various combinations of the above mentioned devices. This is made possible by using the C2 library, which has built-in functions and configuration options to use any of these devices.

CHAPTER 3: MOTIVATION AND OBJECTIVE

Computer graphics and advanced visualization techniques have taken CFD visualization software to a high level of sophistication and ease of use. The graphics programming tools available today are providing programmers with high level functions to develop interaction with three-dimensional objects. Some such programming libraries are Open Inventor [36], IRIS Performer [26] from SGI and World Tool Kit [36] from Sense8. For example, Open Inventor has built-in routines which provide mouse controls to rotate rendered objects (Virtual Trackball), and functions to change the visuals from mono display to stereo display. The stereo visuals can then be viewed using CrystalEyes shutter glasses [8]. In addition to advanced programming tools, today's computers can be equipped with fast graphics hardware, which is optimized to display 3D geometry efficiently. All of these advancements in computer technology have contributed to the rapid growth of VR applications.

Software Tools

The tools used in building the application described here, called VR-CFD, are OpenGL [25], Visualization Tool Kit (VTK) [29], and the C2 library [7]. The Visualization Tool Kit is the main tool used in VR-CFD. This software library provides various functions to facilitate scientific visualization. It provides programmers with many

essential functions, such as streamline calculation routines, and produces the output as OpenGL graphics. These routines can be used in any OpenGL program.

The C2 library is used as the virtual reality platform for VR-CFD. The C2 library provides programmers with functions to facilitate viewing and interacting with the graphics using a number of VR devices. Integrating VTK with the C2 library is possible because both the C2 library and VTK support rendering of graphics using OpenGL. In order to use VTK functions in a C2 program the renderwindow of the VTK is replaced with the X11 window widget created by the C2 library. This procedure is simple enough to facilitate easy porting of applications built using VTK to the C2.

There are a number of commercial application software packages available for post-processing of CFD data. Fieldview [13], EnSight [12], FAST Inc., STAR-CD [30] and Fluent [15] are such commercial CFD post-processing applications. The goal of this research is to extend the functionality provided by the Fieldview software to a virtual environment. Fieldview was chosen because of the input from an industrial sponsor. Further, following an existing commercial application guarantees the usability of VR-CFD directly by industrial users. Fieldview has quite a few interesting and useful features. Some important features are discussed in the following section.

Data storage

Fieldview and EnSight provide the ability to analyze multiblock grid data. The sample data obtained for this research was of multiblock format. Earlier implementation of CFD post processor in VR by Kohlmeyer does not have support for multiblock grid

data. Since multiblock data sets are common in the industry, the approach presented here will include the ability to handle multiblock data.

Ability to display more than one parameter

Analysts are generally interested in examining several data attributes. For example, the sample data used in this research includes pressure, temperature, density, volume fraction, mass fraction and other such data provided for each grid point. Visualization of each of these data is important and in many situations, engineers analyzing the data need to visualize more than one scalar at a time. Sometimes two scalars can be effectively visualized by using color to represent the values of one scalar parameter and the diameter of a streamtube to represent the values of another scalar parameter (Figure 6). This requirement is adequately satisfied by the commercial packages available today. For example, features to display two scalars together by representing one as color and another as thickness of the lines are available in many of these packages including Fieldview.

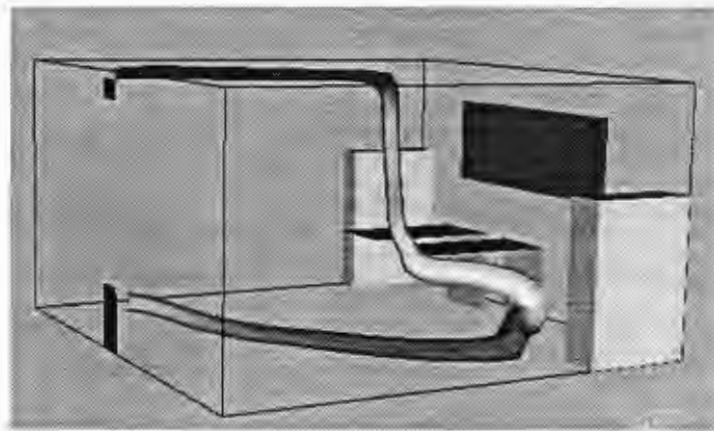


Figure 6. Streamtubes showing variation in diameter [29]

Ability to display vector data restricted to a cutting plane

CFD analysis generally involves investigating several types of vector data. Vector data of interest include velocity and momentum. Visualizing vector values using a 2D-display device is difficult if the flow field is 3D. In 3D flow fields, vector quantities are generally three-dimensional. One method of visualizing vectors is to draw arrows representing them in the flow field. However filling the entire flow field with lines or arrows gives very little information about the type of flow, especially when the visualization takes place using a two dimensional projection plane as in the case of a monitor. Hence, the engineers and scientists restrict the region of visualization by choosing a region of interest.

One method of selecting the region of interest is placing a plane, which is made of triangular pieces, in the flow field. This plane is known as the cutting plane since it cuts through the flow field. Schemes like displaying the vector as arrows restricted only to the points making up the cutting plane are very useful in visualizing flow (Figure 7). In this method, the values of vector parameter at the points placed on the cutting plane are obtained by interpolation. Three-dimensional arrows representing these vectors are placed on the plane starting from points on the plane. Sometimes it is useful to project the arrows onto the plane so that they are restricted to the 2D plane. Figure 7 shows an example of how the arrows on the cutting plane indicate flow rotations. This particular feature is not available in all the commercial applications.

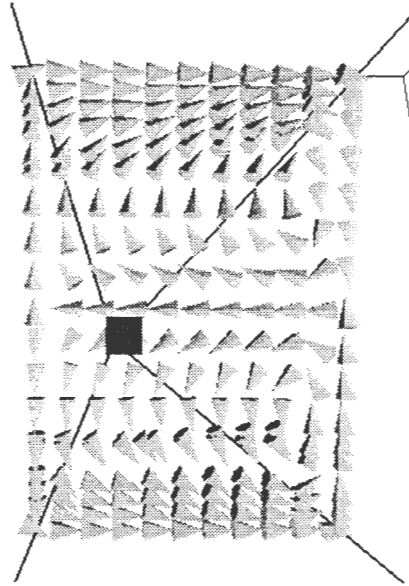


Figure 7. Vectors projected onto a cutting plane

Additional data generation

Generally, the users are interested in visualizing only the pre-computed parameters. However, in some situations, the users are interested in creating and viewing new scalar values based on already read scalar and vector data. For example, the vector data obtained by computing the product of the density scalar and the velocity vector provides more insight into the fluid flow problem. In Fieldview and EnSight, there are functions to create such ad-hoc data using a 'Calculator' feature. Operations such as dot, cross, curl, normalize, exp., ln, sin, etc. are available in Fieldview to act upon existing scalar and vector data in order to create new scalar and vector data. It also allows the users to output the resulting values as additional data so that they can be used for subsequent analyses.

Viewing and Interacting with Data

In all the commercial applications, viewing and interaction with the data are restricted to the 2D monitor and desktop mouse. This presents problems to the analysts when investigating three-dimensional flow. Although the flow exists in 3D, the interaction is confined to projection of the 3D data on the 2D screen. Some of the difficulties felt by users while carrying out these tasks in a 2D display and interaction system are discussed below.

Viewing tasks

In a typical desktop system, the view is changed by using the mouse buttons. The user chooses a menu button to get into the view manipulation mode and then uses mouse button operations to change the position, orientation, and zoom-factor. This way of changing the viewpoint is not natural and is difficult when compared with the ease with which people change their viewpoint in the real world. Why can't the user just walk around to view the models from different positions? This need is not satisfied by the commercial applications available today because of restrictions of the monitor as a display device and the 2D mouse as the input device.

Interaction tasks

In desktop systems users have the option to select the position and orientation of any graphical entity either by specifying the required parameters in a text field or by selecting a point simultaneously in two or three viewports. This is because 3D data are displayed in 2D monitors and interaction is with 2D mouse but users select points in 3D space. These

means of positioning the entities are totally unnatural and frustrate the first time users. Even for experienced users, this operation becomes a time consuming task and one which, in some situations, is limiting. Difficulties faced while carrying out interaction tasks limit the capabilities of the desktop 2D application. In most cases, these increase the amount of time and effort taken by the user to complete the task.

Sometimes these difficulties can be overcome by providing guided controls. For example, EnSight software provides controls by which the graphical entities are constrained to move only in the x-y plane by using one mouse button and only in y-z plane by using another. However, these solutions are not user-friendly.

Stereoscopic viewing

The traditional desktop system displays the graphical objects projected onto a single plane. It is incapable of providing depth information. Visual depth information can be used to convey to the user the exact location of the object in the three-dimensional field. It is difficult to perceive the depth in a desktop system although perspective view projection produces visuals good enough to satisfy most needs. In certain situations, stereo viewing becomes critical as in the case of viewing vector fields. Though hardware is available for carrying out stereoscopic projection, not all the commercial applications provide stereo projection. Increasing number of commercial applications have started providing support for stereoscopic viewing in desktop systems. EnSight and FAST already support stereo viewing on the monitor. Fieldview is providing this feature starting in 1999.

The program described in this thesis attempts to remedy many of the viewing and user-interaction difficulties that exist in current CFD software. Stereo viewing in a fully immersive environment and three-dimensional input is implemented allowing users to easily interact with the data.

This thesis has presented an overview of the field of computational fluid dynamics and its use in industry as well as a short description of virtual reality and the motivation behind the need to develop a VR interface to CFD data. The following chapter will describe the general features of the VR-CFD program and its application to an industrial dataset.

CHAPTER 4: VR-CFD SOFTWARE

The VR-CFD software was developed to provide a virtual reality environment to examine computational fluid dynamics data. The intent was to provide a VR interface that enables users to effectively investigate the data so that they can understand the fluid flow better. This VR-CFD program was designed to provide many of the utilities currently present in the Fieldview software.

This work is an extension of work previously done by Kohlmeyer of Iowa State University [24]. He developed a CFD post-processor by integrating VTK and the C2 library. This application has capabilities to place streamlines, cutting planes in flow fields. User interaction was through a menu placed in the front of the user. The current work is based upon Kohlmeyer's program.

General Features

Some of the features available in the current work are explained below.

User interface

In the application developed, the users choose the mode of operation (navigation, picking, etc.) using a menu which is positioned close to the front wall of the C2 device. The menu consists of a series of polygons, which are texture mapped with images, placed in rows and columns (Figure 8). Each image depicts a specific choice. The images are

made of text, pictures or combinations of both. To select a menu option the user points a ray from the wand and clicks a wand button while the ray is intersecting the chosen option. The ray is made of a line, which is colored brightly and contrastingly so that it stands out visually against the background.

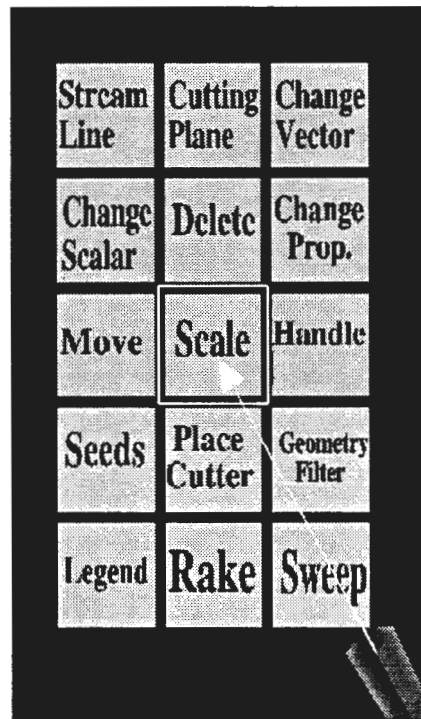


Figure 8. Main menu

A hierarchical menu is used in this program. The menu structure is shown in Figure 9. Choosing some of the menu options will bring up child menus, which take the place of the main menu. An important consideration when designing the menu structure was to minimize the number of times the user needs to select the menu items. Keeping the number low helps the user concentrate on the problem that is being analyzed instead of spending time and effort in the menu selection process.

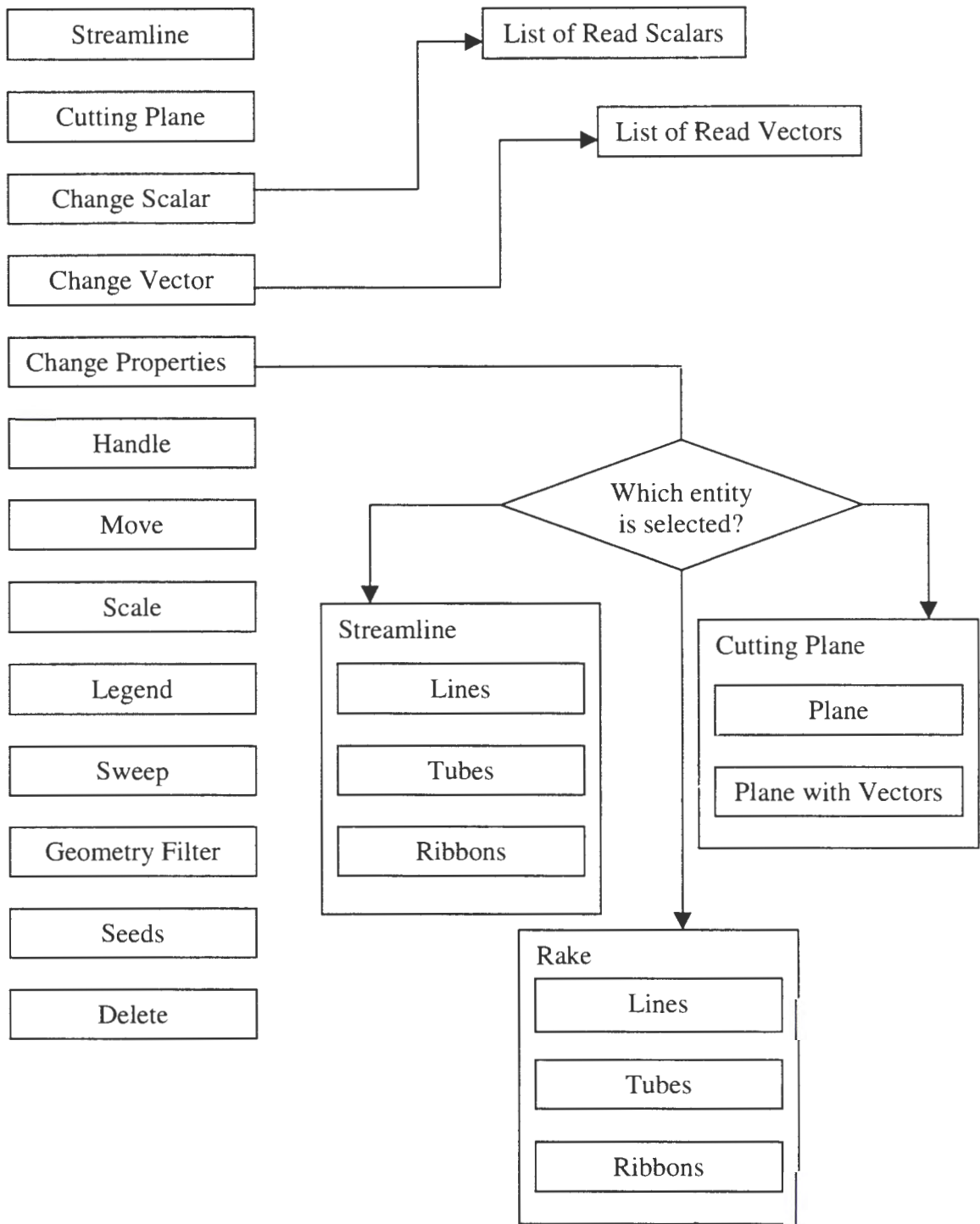


Figure 9. Menu structure

Another consideration was a means to provide shortcuts for certain menu choices for experienced users. For example, after creating a streamline, users can either select the menu option 'Delete' to remove it or just click the middle wand button. During experimental runs, inexperienced users have unintentionally clicked the wand button and deleted entities. Training the users or making the shortcuts more obscure are effective remedies to avoid these accidents. Creating such short cuts is similar to creating a 'Macro' in a word processing application.

Wand pointer

Position of the wand in the virtual environment is indicated by a graphical representation called the pointer. The pointer in VR-CFD is made up of three perpendicular lines with a sphere at their intersection. The presence of the sphere is useful when picking graphical objects. To pick an object the user must insert this sphere inside the object, click and hold a wand button. After being picked, the object moves along with the wand pointer.

Drawing the pointer is essential in applications using the head mounted display since the user's real hand is blocked from view. In applications using the C2, the real hand is not blocked from view, however a virtual pointer is still required. In the C2 when a user tries to pick an object, which looks as though it is closer to the eye than the hand, a problem occurs. The object gets hidden by the user's hand and this causes visual confusion since the user expects the object to hide the hand. This occurs due to the positioning of the projection plane. The projection plane or the screen in the C2 is not as close to the eye as in the case of the head-mounted display. Although virtual objects

might look to be very close to the eye due to the stereo effect, they are actually drawn upon a screen and hence can be obscured by real objects. This phenomenon is known as occlusion. This problem is one of the unsolvable problems of the C2. Drawing a pointer becomes necessary to overcome this difficulty. By drawing the pointer, the users can compare the position of the object they want to pick with the pointer position rather than the hand. The occlusion will not occur when using the wand pointer because both the object and the pointer are drawn as virtual objects on the same projection plane. In the HMD, the occlusion does not occur because the device itself will not let the hand come to a position closer to the eye than the projection plane.

Navigation

The users can navigate around the data using the wand. To move around the data users position the wand away from the body in the direction they want to move and then pull the wand towards them while holding a wand button. This action is similar to the zoom operations provided in the desktop systems. The users can also grab the objects just as they naturally would and position them anywhere at any orientation. Users can also walk within the confined space of the C2. This is possible due to the head tracking.

Handling multiblocked data

One means of handling the multiblock data is to take the blocks as individual structured grids while making analysis entities such as streamlines continue from one block to another. This is done by evaluating the end point of the streamline in one block

and computing the subsequent streamline using this point as the seed. This method brings in additional complexity in order to maintain the continuity of the streamlines.

The other means of handling multiblock data is to read in each block of data, append all of them together, and treat the resulting collection as an unstructured grid. This unstructured grid can then be loaded by VTK and used as the dataset for the computation of streamlines, cutting planes and other visualization components. Though this is theoretically simple, an error in the VTK library makes this task unrealistic at present.

The VTK functions for generating cutting planes work correctly with the unstructured grid formed by appending several structured grids together but streamline generation functions do not result in correct output. Debugging measures undertaken in this research led to the discovery of an error where, VTK fails to generate streamlines when there are coincidental points making up hexahedral cells.

For example, if two hexahedral structured grids each containing eight points are arranged in such a way that one of the faces is common for both grids, when they are appended together they will result in an unstructured grid containing three hexahedral cells where the middle cell will be flat and have coincidental points. In this dataset, the streamline will stop as soon as it crosses the first cell. This error has been reported to the authors of VTK and is expected to be fixed shortly. Another bug was present in a VTK function and initially did not allow even cutting plane generation with unstructured data obtained by appending several structured grids. The bug was in the way the data are written to the output file after the structured grids were appended together. For the

hexahedral cells, the point order was being written as 0,1,3,2,4,5,7,6 instead of 0,1,2,3,4,5,6,7. This bug was fixed in the course of this research.

Streamlines

The streamline is the path through which a massless particle will travel when placed in a flow field. The streamline is tangent everywhere to the velocity vectors. To obtain a streamline, numerical integration is used. VTK uses second order Runge-Kutta method of numerical integration.

The streamlines are generally the most interesting visualization components when visualizing a flow field. The generated streamline can be color mapped to any scalar value. Therefore, the user can examine the path of the particle through the flow and also see the temperature or pressure change along the flow path.

Modified forms of streamlines include streamtubes and streamribbons. The tubes and ribbons are more desirable than lines because their width or diameter can be varied according to a user selected scalar value while the color is varied according to another scalar value. Hence using colored tubes or ribbons, two scalars can be visualized at a time.

In the VR-CFD program, users can interactively pick points where they want to place the seed for generating the streamlines using the wand. One or more such streamlines are generated together to form a rake (Figure 10), which is a collection of streamlines originating from a single source like a line, quadrilateral etc. In VR-CFD the user can choose a line and create a rake using the several points in the line as seeds.

Cutting planes

In order to visualize scalar data it is easier to restrict the region of visualization to just a single plane which is placed inside the flow field. This plane is called the cutting plane since it cuts the flow field. The cutting plane is made of triangular polygons. Scalar values for each point of the triangles are found by interpolation and the color for the triangles are assigned by mapping the scalar values to a color table. Hence, the whole cutting plane shows a distribution of colors according to the scalar values at the individual points making up the plane.

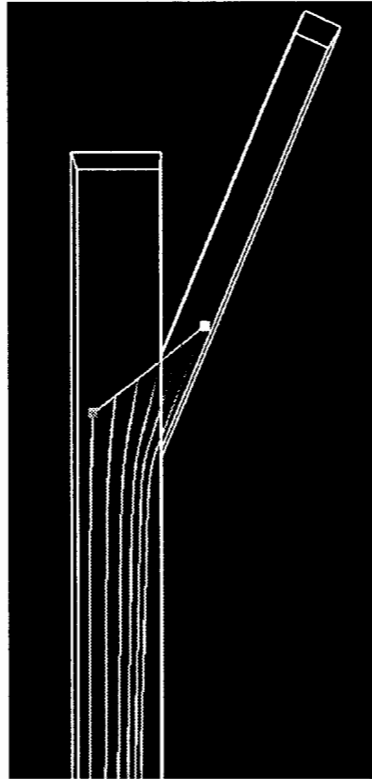


Figure 10 Rake

The user can specify the location and the orientation of the cutting plane using the wand. To move the cutting plane, the user inserts the wand pointer into a cube, which denotes a point on the plane and then orients the plane by modifying the wand orientation. The Figure 11 shows a user, standing inside the C2 device, placing a cutting plane in a flow field using the wand. The number of points on a cutting plane can be increased for accuracy but the default value is generally sufficient.

The vector values at the points making up the plane can be visualized by representing them as arrows. These arrows are positioned at the points on the plane and are oriented along the vector direction at those points. The colors of the arrows are related to the scalar values. The cutting planes with vectors projected onto them are useful while studying flow with rotations. The length of the arrows can indicate the vector magnitude. In this program three-dimensional cones are used in place of the two-dimensional arrows. This is to ensure that the user can see the vectors from all viewing directions.



Figure 11 A person placing a cutting plane in a flow field

Often one of the velocity components is much larger than the other velocity components of the flow at a specific location. For example when a large velocity flow goes down a duct and a side flow is introduced, the feature of interest to the analyst is the cross flow or mixing that is occurring. The magnitude of this cross flow is small relative to the main duct flow and therefore it is difficult to distinguish it. To investigate the mixing flow, the velocity component in the primary flow direction is set equal to zero. The other two components are then used to generate streamlines and cutting planes with vector projections. The streamlines thus generated will be aligned in a single plane. These streamlines show the vortex centers and the cross flow in the flow field clearly.

The cutting planes can also be warped according to user chosen scalar values along their normal (Figure 12). Warping means the points on the cutting plane will be displaced from their original location along the normal of the plane. The amount of displacement can be related to the scalar values. These warped planes generate a surface. Viewing the surfaces in 3D brings out the full potential of the VR system. The color of the warped cutting plane and the displacements of the points can together help to visualize two scalars at the same time.

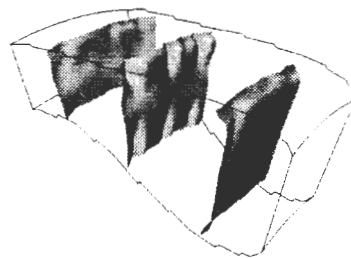


Figure 12 Cutting planes warped with velocity values [29]

Iso-surfaces

Creating polygons connecting points in the data that have the same user-specified scalar value generates an iso-surface. The resulting surface is colored according to user specified scalar value (Figure 13). Contour plots are obtained instead of a surface when a two-dimensional dataset is used instead of a three dimensional one. Thus, if a cutting plane is treated as the source for the iso-surface a contour plot is obtained. The contour plots have usage similar to the cutting planes color mapped according to scalar.

Spatial sweeps

The cutting planes can be swept along a user-specified axis. Though the users can move the planes themselves, the sweep operation automates the process. The gradual motion of the cutting planes gives a good understanding of the chosen scalar value through the entire flow field. An important consideration here is the time delay caused by the computation of the cutting planes. If the time delay is very long then the frame rate drastically falls down because the frames are not drawn during the computation. Thus, the user's immersive feeling is affected while sweeping cutting planes if large data sets are used for the cutting plane generations.

Legend

A color bar, which shows the color mapping for the chosen scalar value, is called the legend. The legend helps the users to see what approximate value the colors on cutting planes and streamlines represent.

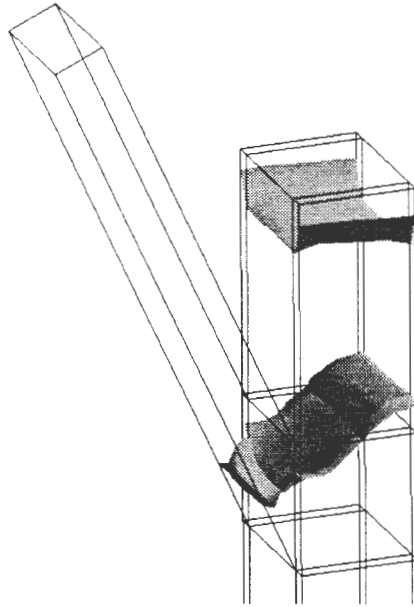


Figure 13 Multiblock grid showing an iso-surface

Changing the scalar and vector data

Users can interactively change the current scalar and vector values by selecting the corresponding menu items. This option is needed when more than one scalar and vector values are specified in the data.

Application of VR-CFD

VR-CFD was tested with the data provided by a major industrial manufacturer. The data provided were in PLOT3D data format. Ten blocks of structured grids with varying densities of grid points were present in the data. Scalar values including pressure, temperature, density, mass fraction etc. and velocity vector values were provided in the data. The geometry of the flow field consisted of a main duct and a side duct, which was

attached to the main duct. The fluid in the side duct is mixed with the fluid in the main duct. The features of interest in this flow are the mixing taking place at the junction of the two ducts. The VR-CFD program was used to investigate this dataset in the C2.

The engineers who used the program noted that being inside the data and the capability to interactively place the streamlines and cutting planes in 3D space was very useful in understanding the data. Instead of using the mouse to rotate and position the data, they were able to use a combination of naturally walking and three-dimensional navigation. Further, instead of using two projection planes to accurately position a point in the data set, they were able to reach out and place points in three-dimensional space. They moved easily around the data to investigate the CFD results. Figure 14 and Figure 15 show a sample data set containing 10 blocks of structured grids with cutting plane and streamlines.

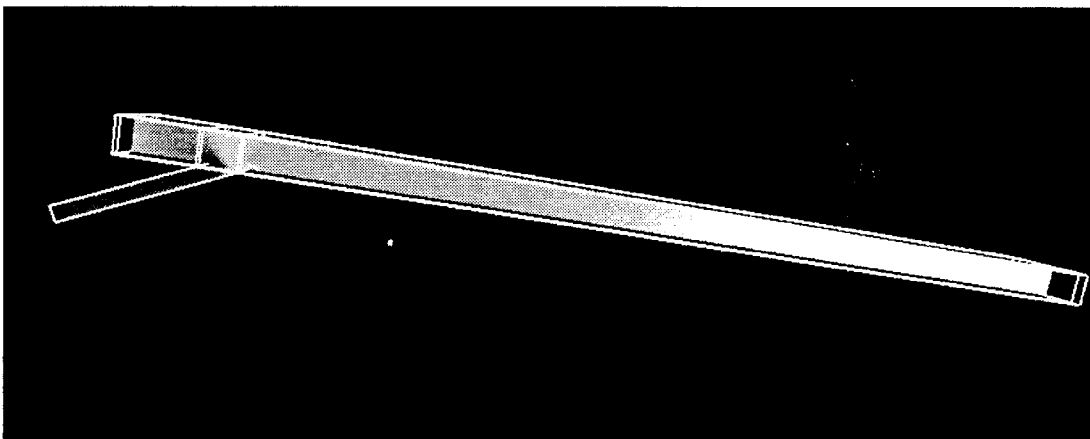


Figure 14 Multiblock data with a cutting plane

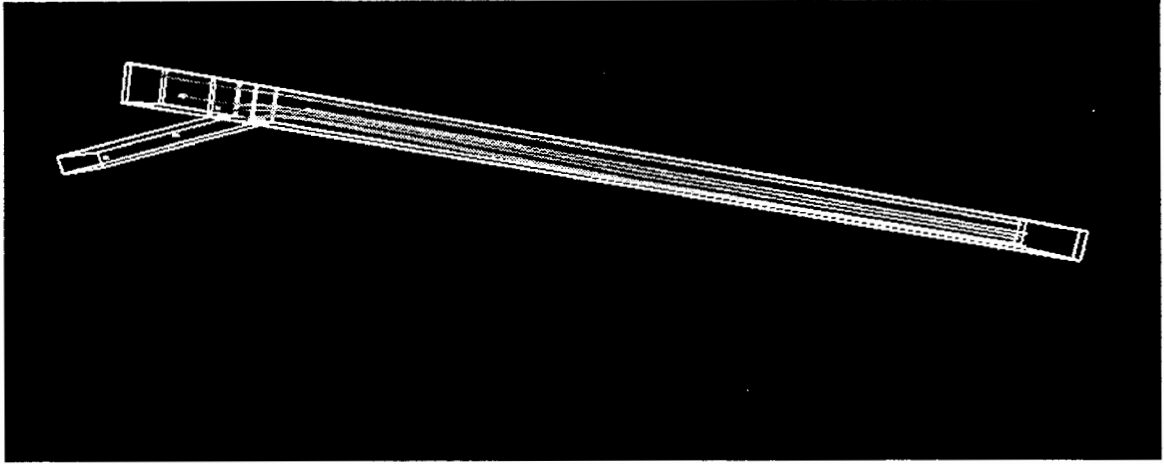


Figure 15 Multiblock data with some streamlines

CHAPTER 5: CONCLUSIONS AND RECOMMENDATIONS

This thesis has explained development of a software program, which combines CFD post-processing capabilities of VTK and virtual reality techniques. Application of VR techniques to CFD post-processing overcomes some of the barriers presented by the traditional 2D graphics and interface methods. In this program fluid flow visualization functions provided by VTK were integrated in a C2 program. The program was built as a general-purpose tool so that any CFD data in the supported format can be read in and analyzed. User interface through a menu was provided to allow the users to interactively pick and place seeds for streamlines, rakes, etc. The goal of extending the earlier work done in this field by Kohlmeyer was reached by providing additional features like

- Support for multiblock ASCII PLOT3D data.
- Support for unstructured grid data.
- Ability to interactively pick and place cutting planes and streamlines.
- Ability to create rakes.
- Ability to navigate through 3D representation of the data.
- Ability to restrict vector representations to cutting planes.
- Ability to read multiple scalars and vectors and interactively change any of the read scalar and vector values.

The project has many open avenues for improvement and development. The computational speed of VTK limits some aspects of real time display and interaction.

Hence, there is need to port the application to a faster application programming interface (API). Possible sources include Intelligent Light Inc. [13] and Computational Engineering Inc. [10] and Application Visualization System (AVS) [35].

The goal of this research was to build a general-purpose post-processing tool, which can be developed into a design tool. Work is currently being carried out in this regard. In the latest advancement, approximate solutions for several different design parameter values are calculated and interpolated values of the scalars and vectors are computed for the intermediate design values. Using this method, a design change can be made in the application and the approximate results can be obtained immediately. This type of design process does not need as much computation power as a full CFD analysis, and is expected to be fast enough to carry out from within the C2.

The approximation procedure can also be used to visualize time varying flow instead of just the steady state flow. Steady state flow data obtained at a few instances of time over a period can be read and the data for the intermediate times can be computed using approximation methods. This will let the users interactively specify the time at which they want visualization results. Similarly, particle trace, which is the actual path a particle placed in a time varying flow field takes, can be computed by using the approximation methods.

The application can be modified to run over a network of VR systems. This will enable users from two different sites to analyze a single flow field. The current menu system is essentially two-dimensional. It can be modified so that the user can change the location and orientation of the menu. Using a standard menu system such as the VUI [19]

will be ideal for future development. The current application supports only PLOT3D [4] data format. Support for other type of data will make this application more usable. Currently the C2 supports image computation for one user. Though more than one user can stand inside the C2 at the same time the displayed perspectives look distorted for all users except the one wearing the tracked glasses. Support for more than a single user will be helpful if more than one user is viewing the data.

Voice recognition is currently a stable and reliable technology. Using it in this application will lead to reducing a number of menu operations. By using the voice recognition software, available from companies like Dragon Systems [10], many of the menu operations can be done away with. Users can simply speak out the action that they need and the software will convert the voice into text and then carry out the required action based on the text.

REFERENCES

1. G. V. Bancroft, F. J. Merritt, T. C. Plessell, P. G. Kelaita, R. K. McCabe, and A. Glorus. "FAST: A Multi-processed Environment for Visualization". In proceedings of Visualization '90, p.14-27, IEEE Computer Society Press, Los Alamitos, CA, 1990.
2. BOOM. Fakespace Inc. homepage. <http://www.fakespace.com> (12 June 1998).
3. S. Bryson, and S. Johan, "The Virtual Windtunnel: An environment for the exploration of three-dimensional unsteady flows", Proceedings of Visualization '91, IEEE Computer Society Press, Los Alamitos, 1991.
4. P. Buning, "PLOT3D Graphics Software for CFD". <http://ad-www.larc.nasa.gov/~buning/plot3d.html>. (12 June 1998).
5. G. Burdea, "Virtual Reality Systems and Applications", Electro' 93 International Conference, Short Course, Edison, N.J. p. 164.
6. Computational Engineering Inc., EnSight User's guide, 1997, Morrisville, NC.
7. C. Cruz-Neira, D. J. Sandin and T. A. DeFanti. "Surround screen projection based virtual reality: The design and implementation of the CAVE", ACM SIGGRAPH 1993 proceedings, p. 135-142.
8. Crystal Eyes. Stereographics Inc. homepage. <http://www.stereographics.com>(12 June 1998).

9. Cyberglove. <http://www.virtualrealitysource.com/virttech/vtmain.htm> (12 June 1998).
10. Dragon Systems Inc. homepage <http://www.dragonsys.com> (12 June 1998).
11. Encyclopedia Britannica, <http://www.eb.com/> (12 June 1998).
12. EnSight. Computational Engineering Inc. <http://www.ceintl.com/> (12 June 1998).
13. Fieldview. Intelligent Light Inc. <http://www.ilight.com> (12 June 1998).
14. Flock of Birds. Ascension Technologies homepage. <http://www.ascension-tech.com/> (12 June 1998).
15. Fluent homepage. <http://www.fluent.com/software/software.htm> (12 June 1998).
16. J. D. Foley and A. Van Dam. Fundamentals of Interactive Computer Graphics, Addison-Wesley Publishing Co., Philippines, 1982.
17. N. Goldfarb, "Virtual Reality: The State of the Art", MicroTimes, October 14, 1991, p.62.
18. Head-mounted display. n-Vision homepage. <http://www.nvis.com> (12 June 1998).
19. D. J. Heath. "Virtual User Interface (VUI) A windowing System for VR", Second International Immersive Projection Technology Workshop proceedings, Iowa State University, Ames IA, 1998.
20. B. Lingard, "A Tutorial on Time-Multiplexed Stereoscopic Computer Graphics". <http://www.cs.wpi.edu/~matt/courses/cs563/talks/stereohtml/stereo.html>. (12 June 1998).

21. Logitech 3D mouse. Logitech Inc homepage. <http://www.logitech.com> (12 June 1998).
22. D. P. Mahoney, "Driving VR", Computer Graphics World, May 1995, p. 22- 35.
23. Microsoft Encarta, Microsoft Inc, Redmond, Washington, 1997.
24. J. Oliver, J. Vance, G. Luecke, C. Cruz-Neira, "Virtual Prototyping for Concurrent Engineering", International Immersive Projection Technology Workshop 1997, p. 51-57.
25. OpenGL, Silicon Graphics Inc. homepage. <http://www.sgi.com/Technology/OpenGL/> (12 June 1998).
26. IRIS Performer, Silicon Graphics Inc. homepage <http://www.sgi.com/Technology/Performer/>. (12 June 1998).
27. Pinchgloves. Fakespace Inc. homepage. <http://www.fakespace.com> (12 June 1998).
28. C. Pozrikidis, Introduction to Theoretical and Computational Fluid Dynamics, Oxford University Press, Inc., New York, N.Y, 1997, p. 559
29. W. Schroeder, K. Martin, B. Lorensen. The Visualization Toolkit: An Object-Oriented Approach to 3D Graphics, Prentice-Hall, Inc., New York, 1998.
30. STAR-CD. Computational Dynamics homepage. <http://www.cd.co.uk/> (12 June 1998).
31. Sterling Software Inc. homepage. <http://www.sterling.com/> (12 June 1998)
32. I. Sutherland, "The Ultimate Display", Proceedings of IFIP Congress, 1965, p. 506.

33. I. Sutherland, "Sketchpad: A Man-Machine Graphical Communication System", Massachusetts Institute of Technology Lincoln Laboratory Technical Report No. 296, Jan 30, 1966.
34. I. Sutherland, "A Head-mounted Three-Dimensional Display", Proceedings of Fall Joint Computer Conference, 1968, p. 757.
35. C. Upson, T. Faulhaber Jr., D. Kamins, and others. "The Application Visualization System: A Computational Environment for Scientific Visualization", IEEE Computer Graphics and Applications, 9(4), p. 30-42, July 1989.
36. J. Wernecke. The Inventor Mentor. Addison-Wesley Publishing Company, Reading, MA, 1996.
37. World Tool Kit. Sense8 homepage. <http://www.sense8.com> (12 June 1998).7