VICON Volume Visualization

being a dissertation submitted in partial fulfilment of the requirements for the Degree of Master of Science

in the University of Hull

by

V Rama Aravind
First degree and where if not Hull

August 2004

Mạhấ Sạmudramlo neeti binduvanti nấ Ee Yokka
Tapana
Bấbấ Pấdạ Pạdmamula ki Ankitham.

This Small Piece of work is
to Adorn the feet of
My LORD
Shri. SHIRIDI SAIBABA
for granting me all that I wished.

Acknowledgements

At first the project has been started with perplexing mind and with unclear cerebrations. The only mentation that's crystal clear at the rudimentary stage of project is that there has been the necessity to learn lots of new ideas in order for both to accomplish the project and to become a successful Master of Science graduate.

My whole-hearted veneration to my Almighty who showered vivid brightness in my way to accomplish this project and for helping me to persuade all sorts of hurdles successfully.

In no way it would be a metaphor if the supervision of Mr. Warren J. Viant ought has been described as Grandeur Supervision. His act of cultivating ideas in ones intellect is undefinable. I would like to thank Mr. James W. Ward wholeheartedly for his timely help and for his prompting jots in many aspects. Tingeing advices from Mr. Derek P. Wills helped in resolving entangled logics. I thank all of them for their help in my way of accomplishing this project.

My respectful prayers to my parents for their moral support and my gratitude to all who implicitly helped me under the name of the University of Hull. Also thanks to the innumerable resources on the WWW which contributed their part of unforgettable guidance while accomplishing this project.

V. Rama Aravind,
25 August, 2004..

Brief Contents

Contents

List of Figures

Figure A3.8: Truncated Capture volume for at least one camera within the selected resolution range 0.71mm and 0.73mm, inclusive.

Figure A3.9: Orthographic view of capture volume common to 20 cameras

Figure A3.10: a) Scene with room.
b) Scene with rendering of room disabled.

Figure A3.11: Volume not being covered by any of the cameras

List of Tables

Trademarks

The following are registered trademarks or trademarks with registration pending.

Windows XP / 2000	Microsoft Corporation.
Acclaim	Acclaim Technologies Inc.
VICON	OMG Plc.
BodyBuilder. BodyLanguage. DynaCal	OMG Plc.
VICON IQ	OMG Plc.
IRIS Explorer	NAG Ltd.
TGS Open Inventor	TGS Inc.

1 *Introduction to VICON Volume Visualization tool*

1.1 Introduction

The goal of the project, *VICON Volume Visualization*, is to build a visualization tool to simulate VICON motion tracking cameras, help visualize and analyse the capture volume of cameras and thereby help setup cameras at optimal positions in a room (studio or laboratory) under effect. The paragraphs below elucidate the terms VICON motion tracking cameras and motion capture and will give an outline of the *VICON Volume Visualization* tool, developed to accomplish the above said goal.

VICON motion tracking cameras are the cameras which are used for live motion capture of subjects with reflective markers[1] placed at well defined positions on the subject. (Tebbutt, Wood and King, 2002)

Motion capture is the recording of movements of subjects by an array of video cameras in order to reproduce it in a digital environment. This three-dimensional reproduction of movements has the advantages like, (Tebbutt, Wood and King, 2002)

- Generating lifelike character animation for movies, video games, broadcast, web cast, etc.
- Incorporating motion into Virtual Environments (VEs) for engineering design
- Understanding of athletic techniques and gait
- Medical assessment and analysis of movement disorders

Applications are countless. The system can be used for gait analysis of physically disabled, for analytic study of Virtual Environment (VE) navigation and etc. Accuracy of the motion capture should be optimal, depending upon the importance of application. More accurate motion capture is desired for a scientific study rather than in the motion capture for entertainment purposes.

Accuracy of the VICON motion tracking system depends upon the number of cameras that can track a particular reflective marker. In turn, the number of cameras that can track a particular reflective marker depends upon the total capture volume *common* for maximum number of cameras in a room. Hence, it is of utmost importance to know in advance, the common volume that will be captured by maximum number of cameras placed at certain positions in a room under effect. And it is also important to decide optimal camera positions so that maximum volume of the room under effect can be captured or tracked.

[1] Markers are reflective spheres attached to the body of subject (the person, animal or whatever else) whose movement is to be recorded. Light from the strobe (placed around lens of a camera) is reflected by markers back into the camera. A filter ensures that only light with the colour of that of strobe light (red or infra-red) is allowed to strike the light sensitive plate behind the lens (Tebbutt, Wood and King 2002).

This report elucidates the development of *VICON Volume Visualization*, a tool that helps in viewing and analyzing the volume captured by VICON cameras with certain specifications and placed at certain positions in a room under effect.

1.2 Objectives of the project

As introduced, the VICON Volume Visualization tool helps in viewing and analyzing the *capture volume* of VICON cameras placed at certain positions in a room under effect. The primary goal of this tool is to help user in deciding positions of cameras so that the maximum volume of the room (studio or laboratory) can be captured or tracked.

Primary objectives that were laid out for developing this tool are,

- Modelling the coverage of each camera and in turn identifying regions that have only partial camera coverage.
- Precise positioning of the cameras
- Modelling field of view for alternative camera lenses

These were the objectives that were laid out originally for the development of the application, VICON Volume Visualization tool. After achieving these objectives, a cerebration to further improve the application by adding more functionalities and building a better user interface had lead to lay down the following secondary objectives.

Secondary objectives that were laid out for refining the tool,

- Modelling the resolution of a point in space for a particular camera
- Visualizing the capture volume of certain number of cameras and volume not being captured by any of the cameras
- An intuitive user interface to manipulate parameters of cameras and view their effect in real-time

The chapters that follow include in them a detailed score of the work done, in developing this application and in accomplishing the above said objectives, viz, description of the application and its developmental stages, algorithms implemented, critical appraisal, technical details pertaining to the VICON motion tracking cameras and etc.

2 *Background*

This chapter provides a background study of VICON Motion Tracking system, its major components and existing volume visualization tools, which are necessary to infer the suitability of the application *VICON Volume Visualization* that has been developed to assist the existing VICON system.

2.1 The VICON motion tracking system

The VICON motion tracking system includes both hardware and software applications for complete control and analysis of motion capture. The hardware includes Datastation, cameras, tripods, reflective markers, gaffer tapes, etc. The general software applications include Workstation, BodyBuilder, Polygon, Plug-in Gait, Plug-in Modeller and Mobius. (Tebbutt, Wood and King, 2002)

A brief introduction to all of the components stated above is appropriate in understanding the working of VICON system.

2.2 VICON Hardware

A short score of general VICON hardware is essential to understand the basic working of the system. The paragraphs below introduce all of the general VICON hardware and try to explain the working of the system.

2.2.1 *Cameras and Strobes*

The camera and strobe unit is that part of VICON system which *collects* light from the reflective markers placed at well defined positions on subjects. The strobe emits light in time with the camera, illuminating the retro-reflective markers attached to the subject. The markers reflect light straight back into the camera where they pass through a filter that allows light of only one particular wavelength to be focussed on the camera’s sensitive plate. Filter controls that, light only with wavelengths same as that of light from strobe are passed through. (Tebbutt, Wood and King, 2002)

Figure 2.1 below shows the front and rear panel of MCam. Types of VICON cameras are introduced later in the section 2.2.5.

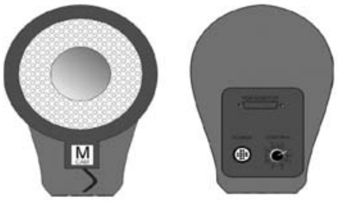


Figure 2.1: The MCam2 style of camera. Front and rear panel. Image Courtesy of VICON Motion Systems, OMG Plc. Source Tebbutt, Wood and King (2002).

2.2.2 *Datastation*

The Datastation controls cameras and analogue devices used to capture data. It is the essential link between these (cameras and analogue devices) and the computer used to process the data. The Datastation resembles a personal computer and is labelled with the system version number. The Entertainment and Animation series are black; the Science & Engineering series (612, 460) are white. The connectors may be on front or on the rear panels. (Tebbutt, Wood and King, 2002)

Figure 2.2 below shows 612 and V8i versions of Datastation. The paragraph before infers that the V8i (black coloured) system is for Entertainment purposes and 612 (white coloured) system is for Science and Engineering purposes. (Tebbutt, Wood and King, 2002)

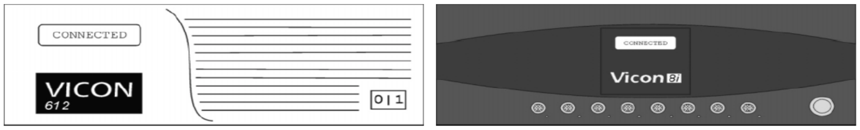


Figure 2.2: Schematic of 612 and V8i Datastation front panels. Image Courtesy of VICON Motion Systems, OMG Plc. Source Tebbutt, Wood and King (2002).

Below is a table which shows some of the VICON systems with their probable applications and maximum number of cameras in that system.

System	Primary Use	Max. no. of Cameras
460	Science and engineering	6
612	Science and engineering	12
624	Science and engineering	24
V6	Entertainment / Animation	6
V8i	Entertainment / Animation	24

Table 2.1: Typical System Hardware specifications. Data courtesy of VICON Motion Systems, OMG Plc. Source Tebbutt, Wood and King (2002).

2.2.3 *WorkStation PC*

Workstation is the software application for controlling the Datastation and the motion capture process. It is also a common name used for the PC on which it is installed. The Workstation PC should be linked through a network to the Datastation. Workstation software shall be installed on the computer that is connected to Datastation. (Tebbutt, Wood and King, 2002)

2.2.4 *Dynacal Calibration Objects*

Calibration of the VICON system is one of the most important steps in capturing motion and recording data. Calibration allows the system to define the capture volume and the relative positions and orientations of cameras. This information is then used with the data from each camera to create the three dimensional motion – a process called reconstruction. (Tebbutt, Wood and King, 2002)

There are two types of calibration: static and dynamic.

Static calibration is used to set the origin and the direction of the axes. The LFrame and FlatCal, as shown in figure 2.3, are used for static calibration, the former being used for full body capture, the latter for small volumes. They both have an L shaped arrangement of reflective material, three on one arm and one on the other. (Tebbutt, Wood and King, 2002)

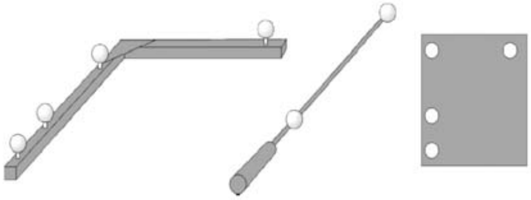


Figure 2.3: Types of Calibration object. L-Frame (left), Wand (centre) and FlatCal (right). Image Courtesy of VICON Motion Systems, OMG Plc. Source Tebbutt, Wood and King (2002).

Dynamic calibration is used to calculate the relative positions and orientation of the cameras. The dynamic calibration object is called a *wand*, shown in figure 2.3. For medium and large volumes this is a rod with two 50mm reflective balls placed 500mm apart. For small volumes the rod should be used with 14mm spheres, 100mm apart. The act of dynamic calibration involves moving throughout the capture volume, waving the wand so that it passes through as much of the Capture Volume as possible allowing each camera to record the wand in several orientations. (Tebbutt, Wood and King, 2002)

2.2.5 *VICON Cameras*

VICON supplies a number of camera types for motion capture. Typical are the MCam series and LCam series cameras. Some of the most common camera types along are briefed in the paragraphs below. (VICON, “nd”a)

M2 Camera (MCam2)
MCam was the first one million-pixel camera uniquely designed and manufactured for motion analysis by VICON motion systems. Now in its second generation, the M2 Camera brings new levels of performance to motion capture. The convenience and accuracy of the system has been considerably increased by larger capture volumes and smaller markers (VICON, “nd”b).

V Camera
Built on the pedigree of the MCam, the V Camera uses a lower resolution CMOS digital sensor. V Camera has the same rugged housing as the M2 Camera but its lower cost offering is a good compromise for applications in which good quality reliable data is preferred to ultimate performance (VICON, “nd”b).

SV Camera
Utilizing the same sensor as the V camera, the SV camera allows capture in small and awkward environments. Its' small format and wide-angle lens allows easy mounting of cameras close to subjects.

LCam
LCam cameras are compact cameras that are appropriate for the applications where motion data has to be collected in small environments. These cameras are less obtrusive and they are appropriate, if the application requires the subjects not to notice them.

2.3 VICON Software

VICON Motion Systems provides a software suite for complete control and analysis of motion capture. These help in collection and processing of raw video data (Tebbutt, Wood and King, 2002). The important of them is the WorkStation which is explained in section 2.3.1. A detailed study of all the software components of VICON system can be had from the CD accompanying this report.

2.3.1 *Workstation*

Workstation is the central application of the VICON software suite for motion capture. It reads in the two-dimensional data from each camera, combines it with calibration data to reconstruct the equivalent digital motion in three dimensions. This can be viewed in Workstation as a virtual three-dimensional motion. After this reconstruction, the data may be passed to other Vicon applications for analysis and manipulation or to third party applications such as those used for digital animation or virtual environments. In addition to providing the control interface for the whole system, it includes tools for editing and clean-up of motion capture data. (Tebbutt, Wood and King, 2002)

In general, a minimum VICON system consists of the following major components. (Tebbutt, Wood and King, 2002)

- Vicon DATASTATION
- A host WORKSTATION PC
- A 100 base-T TCP/IP Ethernet network connection
- Camera units with interfacing cables
- System and analysis software
- Dynacal calibration object
- Accessories kit
- Tripods or other camera mounting devices

This is a very brief background of the VICON motion tracking system and shall give a general idea of the working system, which aids to infer the suitability of the application *VICON Volume Visualization* tool.

2.4 *Existing Volume Visualization tools and techniques*

Volume Visualization is an active area of research and there exist some tools like IRIS Explorer and TGS Open Inventor, which aid in visualizing three dimensional data. The most popular of them is the IRIS Explorer provided by NAG Ltd®. It provides a Modular Visualization Environment to implement custom visualization techniques for various types of applications. Section 3.8 describes various methods of visualization of volume that were maneuvered before the visualization of volume through voxels (Webopedia, 2004) had been chosen, eventually. Figure 3.14 in section 3.8 depicts visualization of volume through the technique *Volume Slicing*, accomplished using IRIS Explorer V5.0 and does a considerable critical appraisal over this technique.

Principally, there are two fundamental methods of volume visualization; they are Direct Volume Rendering and Surface-Fitting (Brodlie, 1992). Direct Volume Rendering implies that data elements are directly mapped on to the screen with out any representation through geometric primitives and Surface-Fitting implies that a particular data set is parsed to *fit* a surface depending upon a threshold value and the resulted data is represented through the means of geometric primitives. (Brodlie, 1992)

The application VICON Volume Visualization uses a variation of the second method, Surface-Fitting. The data set (building of voxels out of volume has been explained in the section 3.9) will be parsed through, for a specific value, for example, number of cameras (refer to section 3.3) and geometric primitives, quadrilaterals, are used to build small cubes, which are termed as voxels

(Webopedia, 2004). An *Iso-Volume* is built out of the total volume, using voxels, which is based upon the specific value that the data set has been parsed for. Figure 3.17 shows application with volume visualized in this technique using voxels. Advantages of this type of visualization are elucidated in section 3.8.

The chapter that follows explains the topics like, simulating VICON cameras, building capture volume, visualizing the volume and etc., which are an important part of the application VICON Volume Visualization.

3 *Design*

The tool VICON Volume Visualization operates through the aid of ad hoc mathematics principles that use basic algebra. These principles have been elucidated in this chapter. Some of the topics that are described in this chapter are,

- Volume Capture of a Camera,
- Simulating Field Of View (FOV) of a camera,
- Orientation of camera FOV vectors and
- Calculating resolution of a camera.

In brief, the section *Volume Capture of a Camera* explains the method of testing whether a point in 3D space falls into the field of view of a camera or not, the section *simulating Field Of View of a camera* explains the calculation of horizontal and vertical FOVs of a generic camera based upon the cameras' technical specifications, the section *orientation of camera FOV Vectors* explains the order of rotations required for aligning the camera in a particular direction in 3D space and the section *calculating resolution of a camera* explains the method of finding the resolution of a point in 3D space at a particular distance for a generic camera with certain specifications.

The sections below explain the above skimmed topics in detail.

3.1 *Field of View (FOV) of a camera*

The important part of VICON Volume Visualization is to simulate practical cameras, programmatically. The horizontal and vertical FoVs of camera are the parameters that specify how much volume a camera can span. The FOV of a camera is based upon its technical parameters, focal length of the lens and dimensions of the photo sensitive film frame at the back of the lens. In an opto-mechanical camera the photo sensitive film frame at the back of the lens is an emulsified photographic film which, in general terms is called as a *negative film* and in a digital camera the photo sensitive film frame at the back of the lens is a *light sensitive plate* (VICON, "nd"a) called in general terms as CCD (Charge Coupled Device) sensor.

Given the focal length of lens (f) and dimensions of the film frame (width * height) at the back of the lens, horizontal and vertical FOVs of camera can be calculated using the formulae given below, (Rigg, 2002)

- Horizontal FOV = $2 * \tan^{-1}$ (width / 2 * f)
- Vertical FOV = $2 * \tan^{-1}$ (height / 2 * f)

As indicated, 'f' stands for focal-length of the lens, 'width' implies the horizontal size of the film frame and 'height' implies the vertical size of the film frame. Generally used 35mm negative films in opto-mechanical cameras have the dimensions of 36mm X 24mm (Rigg, 2002) and 70mm films have the dimensions of 72mm X 24mm (Fitzgerald, 2000). Similarly, the dimensions of photo sensitive film frames or CCD Sensors in digital cameras vary among different types of cameras. The VICON cameras' specifications provided by VICON are as below which give the technical parameters of general VICON cameras.

Lens Focal Length:	8.5
Pixel Size (mm):	0.0099

Frequency	H Pixels	V Pixels	H Sensor Dim in mm.	V Sensor Dim in mm.	H FoV	V FoV
100	659	493	6.5241	4.8807	41.99	32.04
120	659	493	6.5241	4.8807	41.99	32.04
150	659	493	6.5241	4.8807	41.99	32.04
180	659	493	6.5241	4.8807	41.99	32.04
200	659	493	6.5241	4.8807	41.99	32.04

Table 3.1: Technical parameters of general VICON cameras provided by VICON Motion Systems, OMG Plc.

The parameters of interest in the table above are the lens focal length, horizontal sensor dimension in mm (H Sensor Dim in mm), vertical sensor dimension in mm (V Senor Dim in mm), horizontal field of view (H FoV) and vertical field of view (V FoV).

The horizontal sensor dimension is the *width* of the film frame and vertical sensor dimension is the *height* of the film frame, which is located at the back of lens.

Test

A general test, between the actual parameters provided by VICON Motion Systems and results of the formulae given above gave satisfactory results.

Lens focal length = 8.5mm
Film frame width = 6.5241mm
Film frame height = 4.8807mm

Substituting the above values in the formulae,

Horizontal FOV = $2 * \tan^{-1} (6.5241 / 2 * 8.5) = 41.99 \sim 42.00$
Vertical FOV = $2 * \tan^{-1} (4.8807 / 2 * 8.5) = 32.04$

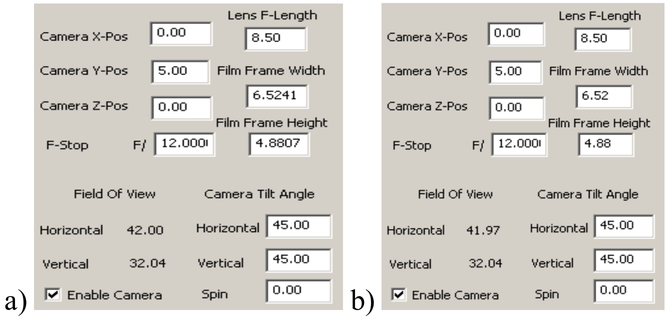
Figure 3.1 shows the screen shot of the application showing the FOVs in respective Static Text Box controls. The calculated FOV values vary with a slight precision of about 0.001 (when considering numbers with precision of two digits after decimal point) to actual values provided by VICON Motion Systems Ltd., and this is shown in figure 3.1-a. This difference is due to round off of the numbers after calculation. The application uses double precision floating point variables and the results are quite close to the actual values.

3.2 *Orientation of camera FOV vectors*

FOV Vectors

Section 3.1 describes the calculation of horizontal and vertical FoVs of camera and did a critical test of the calculations with the actual parameters provided by VICON Motion Systems Ltd.

Figure 3.2 depicts the actual representation of the horizontal and vertical field of view values by the camera models visualized in the application VICON Volume Visualization tool.


a)

Camera X-Pos 0.00
Lens F-Length 8.50
Camera Y-Pos 5.00
Film Frame Width 6.5241
Camera Z-Pos 0.00
Film Frame Height 4.8807
F-Stop F/ 12.000
Field Of View
Horizontal 42.00
Vertical 32.04
Camera Tilt Angle
Horizontal 45.00
Vertical 45.00
Spin 0.00
☑ Enable Camera

b)

Camera X-Pos 0.00
Lens F-Length 8.50
Camera Y-Pos 5.00
Film Frame Width 6.52
Camera Z-Pos 0.00
Film Frame Height 4.88
F-Stop F/ 12.000
Field Of View
Horizontal 41.97
Vertical 32.04
Camera Tilt Angle
Horizontal 45.00
Vertical 45.00
Spin 0.00
☑ Enable Camera

Figure 3.1: a) Part of a screen shot of application window showing the horizontal and vertical FOVs for the given film frame dimensions. b) Screen shot of application showing results for values with two precision floating numbers (6.52 and 4.88 instead of 6.5241 and 4.8807 as provided by VICON Motion Systems Ltd.)

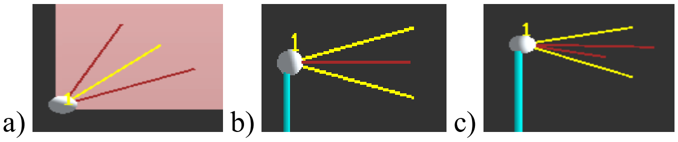


Figure 3.2: a) Camera model in the application with two red lines indicating horizontal field of view. b) Camera model in the application window with two yellow lines indicating vertical field of view. c) 3D view of the camera model in the application window with both horizontal and vertical FOV indicators.

In figure 3.2-a horizontal field of view is represented by the lines rendered in red hue, in figure 3.2-b the vertical field of view is represented by the lines rendered in yellow hue and figure 3.2-c shows the 3D view of the camera positioned at the top front left corner of the room. The camera is modelled as a sphere as shown in figure 3.2 and the number above the camera represents the *number* of the camera as listed in the Combo Box control under title 'Select Camera' shown in figure 2.16.

Note: In the VICON Volume Visualization tool all the cameras in the application are numbered in the order they were specified. By default first four cameras are positioned at the top four corners of the room CCW starting from the top left front corner of the room. All other cameras are initially assigned the centre position of the room (origin in world coordinates) and user has to explicitly define the positions of the cameras individually. In contrast, a camera data file (Windows text file) with number of cameras and with their positions and their technical parameters can be uploaded into the system, which is described in the section 4.2.18.

The paragraphs that follow explain the procedure of orienting the camera FOV vectors which are depicted in figure 3.2.

Orientation

Programmatically, a *camera* is represented by four vectors indicating its FOV vectors and a point indicating its location in 3D space. Out of the four vectors two vectors are parallel to *x*-axis and point in positive direction of *x*-axis and the other two vectors are parallel to *y*-axis and point in negative direction of *y*-axis, with reference to the coordinate system shown in figure 4.11.

Two vectors parallel to *x*-axis and pointing in +ve direction are used to represent the vertical top and vertical bottom vectors and the other two vectors parallel to *y*-axis and pointing in –ve direction are used to represent the horizontal left and horizontal right vectors which are explained later in this section.

In addition a point indicates the location of camera in 3D space. In order to position a camera at a certain location with certain tilt angles and certain FOVs, the following procedure is followed which includes a series of rotations that orient the camera in accordance with the parameters specified.

As described in the section 4.2.11 and as shown in figure 4.16, a camera has four parameters that govern its location and orientation and three parameters that govern its FOV. They are,

- Horizontal tilt angle,
- Vertical tilt angle,
- Camera spin,
- Lens focal length,
- Film frame width and
- Film frame height.

The first three parameters directly imply the orientation of the camera with respect to the coordinate system shown in figure 4.11. The last three parameters infer horizontal and vertical FOV values of the camera which are used in positioning horizontal left, horizontal right, vertical left and vertical right vectors of a camera.

The phrase *orientation of camera* implies the orientation of the four vectors (horizontal left and right, vertical bottom and top) that infer the FOV and orientation of camera.

Consider the values α, β and γ for camera spin, vertical tilt angle and horizontal tilt angle, respectively. Similarly consider the values HFOV and VFOV for horizontal and vertical field of view values of a camera, respectively. The procedure for orienting the duo pairs of vectors (horizontal left and right, vertical bottom and top) is elucidated below with the help of these denotations. The coordinate system used is as shown in the figure 2.12.

In order to orient vectors they have been rotated using matrices in homogenous coordinates. Operations in homogenous coordinates aids in identifying the distinction between vectors and points.

Horizontal Left Vector
The horizontal left vector bears an initial value of (0, -1, 0) which represents a vector pointing in the negative *y*-axis direction. The vector, after converting into homogenous coordinates morphs as (0, -1, 0, 0), the last zero indicates that the set (0, -1, 0, 0) is a vector. A one (1) as the last value indicates that the set is a point.

- Rotate the vector HFOV/2 degrees around *x* axis,
- Then rotate –α degrees around *y*-axis,
- Then rotate 90 – β degrees around *z*-axis,
- And then again rotate γ degrees around *y*-axis, in order.

These rotations will align the horizontal left vector in accordance with the specified parameters.

Horizontal Right Vector
The horizontal right vector also bears an initial value of (0, -1, 0) which represents a vector pointing in the negative *y*-axis direction. The vector, after converting into homogenous coordinates morphs as (0, -1, 0, 0).

- Rotate the vector –HFOV/2 degrees around *x*-axis,
- Then rotate –α degrees around *y*-axis,
- Then rotate 90 – β degrees around *z*-axis,
- And then rotate γ degrees around *y*-axis, in order.

These rotations will align the horizontal right vector in accordance to the specified parameters.

Vertical Top Vector
The vertical top vector bears an initial value of (1, 0, 0) which represents a vector pointing in the negative *y*-axis direction. The vector, after converting into homogenous coordinates morphs as (1, 0, 0, 0).

- Rotate the vector VFOV/2 degrees around *z*-axis,
- Then rotate α degrees around *x*-axis,
- Then rotate –β degrees around *z*-axis,
- And then rotate γ degrees around *y*-axis, in order.

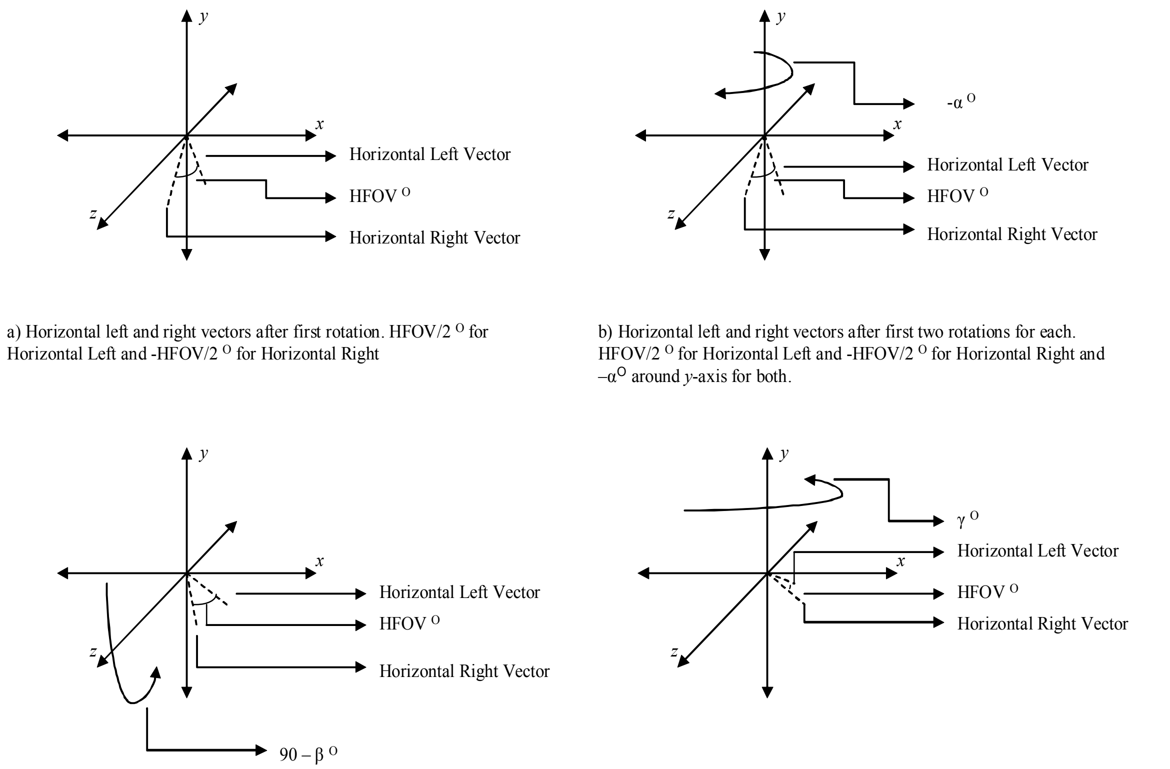
Vertical Bottom Vector
The vertical bottom vector bears an initial value of (1, 0, 0) which represents a vector pointing in the negative *y*-axis direction. The vector, after converting into homogenous coordinates morphs as (1, 0, 0, 0).

- Rotate the vector –VFOV/2 degrees around *z*-axis,
- Then rotate α degrees around *x*-axis,
- Then rotate –β degrees around *z*-axis,
- And then rotate γ degrees around *y*-axis, in order.

The above set of rotations for the four vectors described, align the horizontal left, horizontal right, vertical top and vertical bottom vectors which imply the camera FOV and its orientation.

A careful observation of the above set of rotations for the two pairs of vectors gives a meaningful insight. That is, rotations for the duo pairs of vectors vary only in one term for each. That is, the rotations for horizontal left and right vectors vary only when rotating around *x*-axis for first time. That is horizontal left vector rotates about HFOV/2 degrees around *x*-axis and horizontal right vector rotates about –HFOV/2 degrees around *x*-axis. All the remaining rotations are similar for both of the vectors. This variation makes the two vectors align as horizontal left and right vectors as shown in figure 3.2-a spanning a horizontal FOV of HFOV degrees.

Similarly, the rotations for vertical top and bottom vectors vary only when rotating around *z*-axis for the first time. That is, vertical top vector rotates about VFOV/2 degrees around *z*-axis and vertical bottom vector rotates about –VFOV/2 degrees around *z*-axis. Again all the rest of the rotations are similar for both.



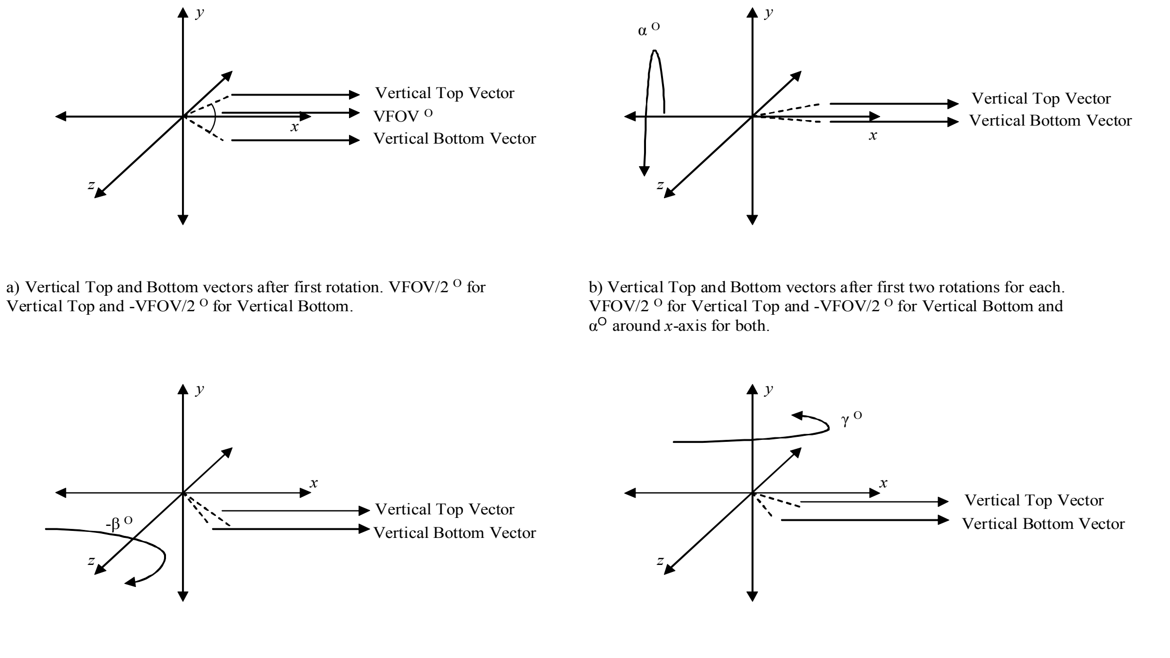
a) Horizontal left and right vectors after first rotation. HFOV/2 ^{O} for Horizontal Left and -HFOV/2 ^{O} for Horizontal Right

b) Horizontal left and right vectors after first two rotations for each. HFOV/2 ^{O} for Horizontal Left and -HFOV/2 ^{O} for Horizontal Right and $-\alpha^{O}$ around y-axis for both.

c) Horizontal left and right vectors after three rotations. HFOV/2 ^{O} for Horizontal Left and -HFOV/2 ^{O} for Horizontal Right, $-\alpha^{O}$ around y-axis for both and 90 – β ^{O} around z-axis for both.

d) Horizontal left and right vectors after four rotations. HFOV/2 ^{O} for Horizontal Left and -HFOV/2 ^{O} for Horizontal Right, $-\alpha^{O}$ around y-axis for both, 90 – β ^{O} around z-axis for both and γ^{O} around y-axis for both.

Figure 3.3: Order of rotations for horizontal left and right vectors



a) Vertical Top and Bottom vectors after first rotation. VFOV/2 ^{O} for Vertical Top and -VFOV/2 ^{O} for Vertical Bottom.

b) Vertical Top and Bottom vectors after first two rotations for each. VFOV/2 ^{O} for Vertical Top and -VFOV/2 ^{O} for Vertical Bottom and α^{O} around x-axis for both.

c) Vertical Top and Bottom vectors after three rotations. VFOV/2 ^{O} for Vertical Top and -VFOV/2 ^{O} for Vertical Bottom, α^{O} around x-axis for both and –β ^{O} around z-axis for both.

d) Vertical Top and Bottom vectors after four rotations. VFOV/2 ^{O} for Vertical Top and -VFOV/2 ^{O} for Vertical Bottom, α^{O} around x-axis for both, –β ^{O} around z-axis for both and γ^{O} around y-axis for both.

Figure 3.4: Order of rotations for vertical top and bottom vectors

This variation makes the two vectors align as vertical top and bottom vectors as shown in figure 3.2-b spanning a vertical FOV of VFOV degrees.

Figures 3.3 and 3.4 depict the order of rotations for pairs of horizontal and vertical vectors, respectively, in the order explained.

3.3 *Volume Capture of a Camera*

This is the most important part of the program which tests whether a point in 3D space falls into the FOV of a camera or not by using simple vector mathematics.

Testing a Point in 3D space

After the horizontal and vertical field of view vectors have been established and oriented correctly, the calculations for testing proceeds as described below.

The procedure is explained by considering the point **P** located at (Px, Py, Pz).

Step 1: Consider that the camera is positioned at a point **C** = (Cx, Cy, Cz). The operation **P-C** results in a vector that represents the direction of the point **P** with respect to the camera position **C**. Consider the resulting vector as **PC**.

Step 2: Consider the horizontal left vector and horizontal right vector (two red lines as shown in figure 3.5-a) as **HL** and **HR**, respectively. Similarly, consider vertical bottom vector and vertical top vector (two yellow lines as shown in the figure 3.5-b) as **VB** and **VT**, respectively.

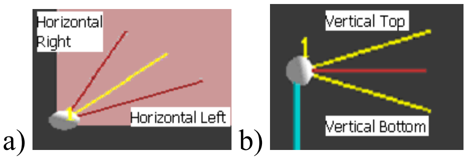


Figure 3.5: Horizontal Left and Horizontal Right vectors representing horizontal field of view of a camera.

Step 3: Find the angle between the vectors **HL** and **PC**. If the resulted angle is greater than the horizontal field of view, then, obviously point **P** lies out of the horizontal field of view of camera. Calculation of angle between vectors is explained in the appendix 1.

Step 4: If the condition in step 3 has been satisfied, that is, the angle between **HL** and **PC** is less than horizontal field of view; perform another check for confirmation by calculating angle between the vectors **PC** and **HR**. If the angle between the two vectors is again greater than the horizontal field of view, point **P** lies out of the horizontal field of view. This step is necessary and avoids improper volume detection. Even if the condition in step 3 has been satisfied then the condition in step 4 will detect if the point actually lies in the field of view of camera or not.

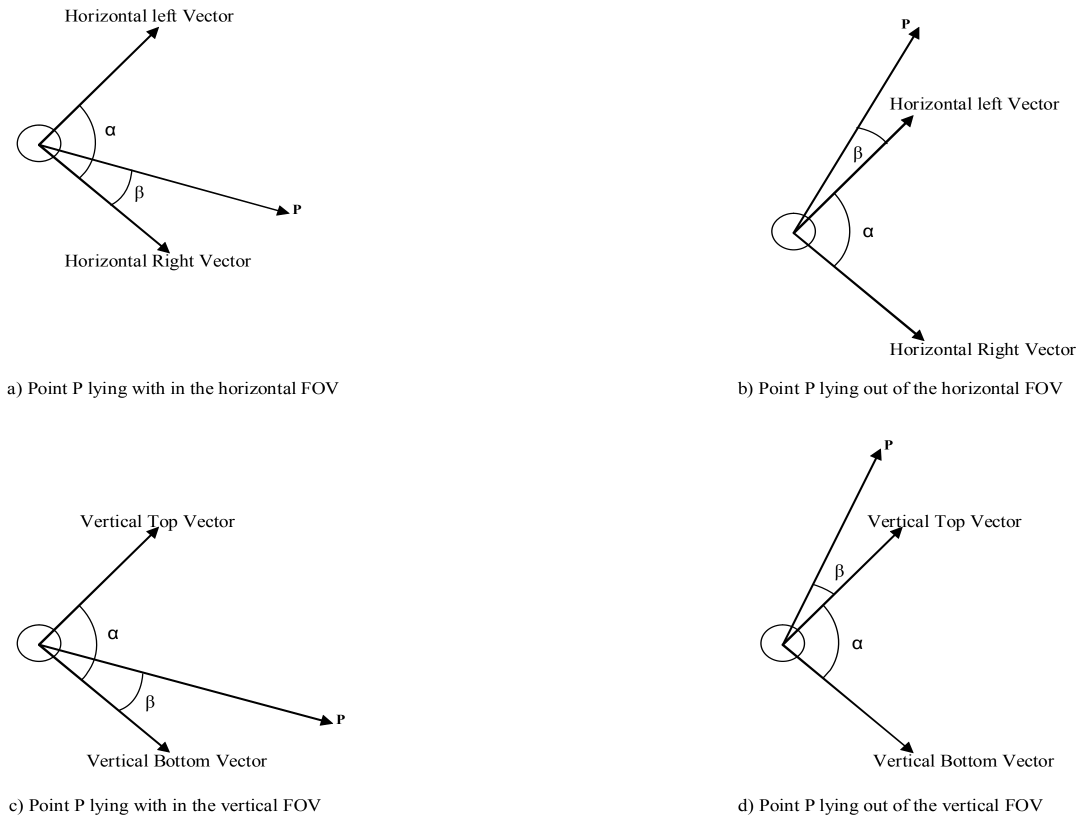
This can be explained through the aid of figure 3.6. As shown in the figure 3.6-b β is less than α (horizontal field of view) but the point **P** lies out of the field of view. Hence this step is necessary to avoid improper volume detection. The angle between **HR** and **PC**, in the case of figure 3.4-b is greater than horizontal field of view and hence point **P** lies out of the horizontal field of view of camera.

Step 5: If the condition in steps 3 and 4 have been satisfied, find the angle between **PC** and **VB**. If the angle is greater than the vertical field of view, point lies out of the vertical field of view of camera.

Step 6: Perform another cross check by finding the angle between **PC** and **VT** to avoid improper volume detection as explained in step 4.

Step 7: If all the conditions from steps 3 through 6 are satisfied then the point P lies with in the field of view of camera, if any of the conditions from steps 3 through 6 is not satisfied then it is not necessary to perform remaining steps as the point P lies out of the field of view of camera.

Figure 3.6 depicts various locations of point P with respect to horizontal and vertical FOVs of a camera. The steps 3 through 6 detect the volume for various situations as depicted in figure 3.6.



a) Point P lying with in the horizontal FOV

b) Point P lying out of the horizontal FOV

c) Point P lying with in the vertical FOV

d) Point P lying out of the vertical FOV

Figure 3.6: Point P depicted at various locations with respect to horizontal and vertical FOVs of a camera.

3.4 *Calculating Resolution of a point in Space for a Camera*

As an added functionality, the VICON Volume Visualization tool helps in analysing the geometric spread of a single abstract point, on the film-frame of a camera. This geometric spread of an *abstract* physical point on the film-frame of a camera is indicated in this report as *resolution*. This resolution has been visualized with the aid of a colour map overlaid on the capture volume and is explained in the section 3.5. A more robust way of analysing capture volume is described in section 6.2

The resolution of a camera can be calculated with the help of a simple formula as explained here under.

The underlying mathematics for the calculation of image resolution is based upon the geometry of a simple Single Lens Reflex (SLR) camera. At first a simple SLR camera has been considered to simulate the FOV of cameras. The geometry of a basic SLR camera has been used in simulating actual VICON cameras.

In addition, the physical observation of VICON cameras, their construction consisting of a photo sensitive frame, at the back of a lens, to capture images proves that the basic geometric principles underlying VICON cameras are that of an SLR camera.

The features of an SLR camera like image resolution, image intensity, field of view, image sharpness and image size are related by simple mathematics. Physical features of an SLR camera are depicted through the aid of figure 3.7.

The resolution, or sharpness, of an image is governed by the size of aperture of camera. An aperture of relatively large size in geometric optics decreases the sharpness of image. Similarly, decreasing the size of aperture in geometric optics increases the sharpness of an image. An expression for determining image resolution that holds in the realm of geometric optics can be derived from relations using the geometry of the triangles formed by light rays traced from the object to the projected image. The figure shown in figure 3.8 and formula given in figure 3.9 demonstrate the deviation, or spread, of a single point on the physical object to the projected points on the resulting image using geometric relations. (Parente, 2002)

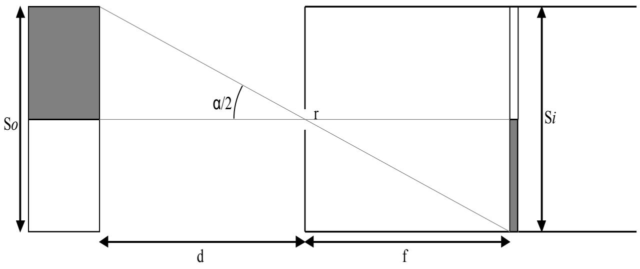


Figure 3.7: Physical features of SLR camera. Focal length (f), viewing distance (d), aperture diameter (r), image size (s_i), object size (s_o) and field of view (α). Source Parente, 2002.

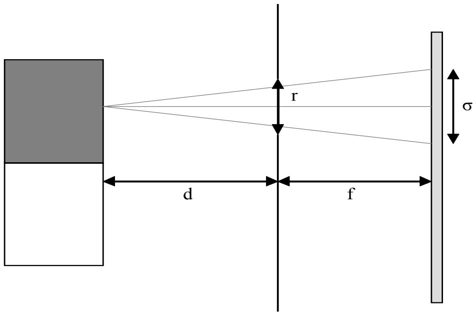


Figure 3.8: The geometry of triangles formed by light rays. Source Parente, 2002.

$$\frac{\frac{\sigma}{2}}{d+f} = \frac{\frac{r}{2}}{d}$$

$$\frac{\sigma d}{2} = \frac{r(d+f)}{2}$$

$$\sigma = r(1+\frac{f}{d})$$

Figure 3.9: Derivation of formula for calculating geometric spread of a point from physical object on the projected image. Figure courtesy of Parente (2002).

The formula given in figure 3.9 elucidates that the spread of points with in the image are directly proportional to both aperture diameter and focal length (Parente, 2002). That is, if focal length increases the FOV decreases and image resolution increases.

Lemma: *It is a general known concept that if FOV of a camera increases the resolution of image formed reduces and vice-versa. The paragraph above describes that the spread of a physical point on an image is directly proportional to focal length in addition to the diameter of aperture. As the focal length of a lens decreases the FOV increases and as the FOV increases the spread of a single physical point on an image decreases and hence the image resolution. Hence the logic that 'spread of points with in the image is directly proportional to focal length' can be declared as* ***true*** *by taking into consideration, the definition of resolution of an image. The more the pixels that a point occupies in an image, the higher the resolution of the image. And the spread of a single point of physical object on image directly corresponds to the number of pixels. The greater the spread, the larger the number of pixels.*

The formula σ = r * (1 + f/d) which is given in figure 3.9 has been used to calculate the resolution (spread of a point) of a point in space for a camera. The denotations r, f and d in the formula imply aperture diameter, focal length of lens and distance between point in space and lens of camera, respectively.

Thus, as the distance between camera (lens of camera) and point in space increases, the resolution (spread of point) decreases giving a diminishing detail of the picture and hence both are inversely proportional.

Hypothetically, when considering VICON cameras the resolution of a camera is not of much importance as the VICON cameras are specifically designed with filters to detect the gait of retro-reflective markers and as long as the cameras are able to detect the markers gait, the resolution does not play an important role. Instead the same analysis has to be done in a different way. That is, analysis of scan lines, which is explained in the section 6.2.

The sections that follow describe the colour map used for visualizing the resolution of a camera which is overlaid on the capture volume and critiques that the colour map used in the VICON Volume Visualization tool is better of all the colour maps that are used and tested in prior.

3.5 *The rainbow colour map*

The visualization technique that has been used to visualize the resolution overlaying on the capture volume is *discrete contour shading* (Brodlie, 1992). This technique implies that if a particular data value falls in a particular range, it shall be visualized with a distinct colour and hence the whole data range will be visualized as a combination of discrete colour bands where each colour specifies data in a specific range. The colour map that has been chosen eventually is the rainbow colour map which is a natural colour map that indicates linear variation of colours from less stimulating violet through highly stimulating red. Rainbow colour map is used for discrete shading of resolution values in the capture volume. That is, the capture volume with resolution in a particular range is shaded in a particular colour. This gives an insight of variation of resolution from low to high over the whole volume. As the resolution varies from low to high the colour map varies from violet through red. And naturally the colours give an insight of variation of data from low to high.

The resolution range of camera will be divided into seven equal ranges. It can be calculated simply by the relation,

Value of a range = (maximum resolution of camera – minimum resolution of camera) / number of ranges

Maximum resolution of camera – resolution of camera at the nearest wanted point, that is, at a distance equal to the half of voxel size.

Minimum resolution of camera – resolution of camera at the farthest wanted point, that is, at a distance equal to the diagonal length[2] of room.

The *Value of range* thus calculated is used to find the seven data ranges or eight data parts.

The capture volume of a single camera overlaid with resolution colour map is shown in figure 3.10.

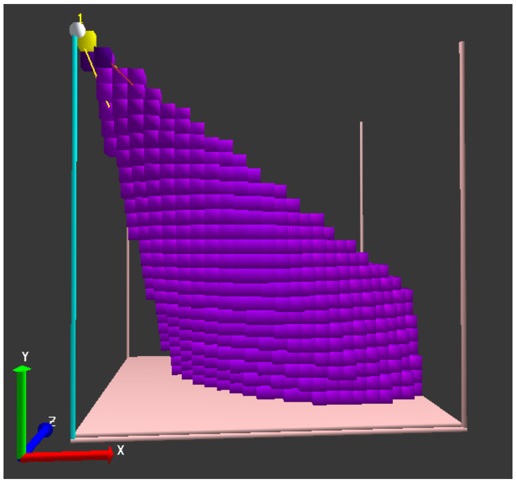


Figure 3.10: Capture volume of a single camera, as indicated, with resolution colour map of that camera overlaid on the volume.

The legend of the colour map is as shown in figure 3.11.

[2] Diagonal Length of a cube with a side of length *s* is equal to $\sqrt{3} * s$ and diagonal length of rectangular solid with dimensions as *l*, *w* and *h* which imply length, width and height, respectively, is equal to $\sqrt{(l^2 + w^2 + h^2)}$ (SparkNotes, 2004). Hence the diagonal length of room is equal to $\sqrt{(l^2 + w^2 + h^2)}$ where *l*, *w* and *h* imply length, width and height, respectively.

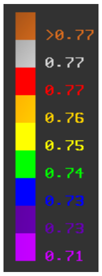


Figure 3.11: Legend of colour map depicting various ranges of resolution and their respective colours.

Figure 3.10 gives an insight that most of the capture volume falls into a *safer region*. It can be considered as safer region because there is not a drastic variation between the ranges of resolution which are depicted in figure 3.11. The same effect is observed with ranges of different resolution values as depicted in figure 3.12. Colours white and saffron on the top of colour red, shown in figure 3.11, indicate volume with resolution out of the specified range but it wasn't encountered in any of the simulations. The colour map depicted in figures 3.10 and 3.12 with various focal lengths gives same insight. Much of the capture volume is falling under the *safer region* as described above.

In figures 3.10 and 3.12 the camera considered for resolution is the one shown in those figures located at top left front corner of the room.

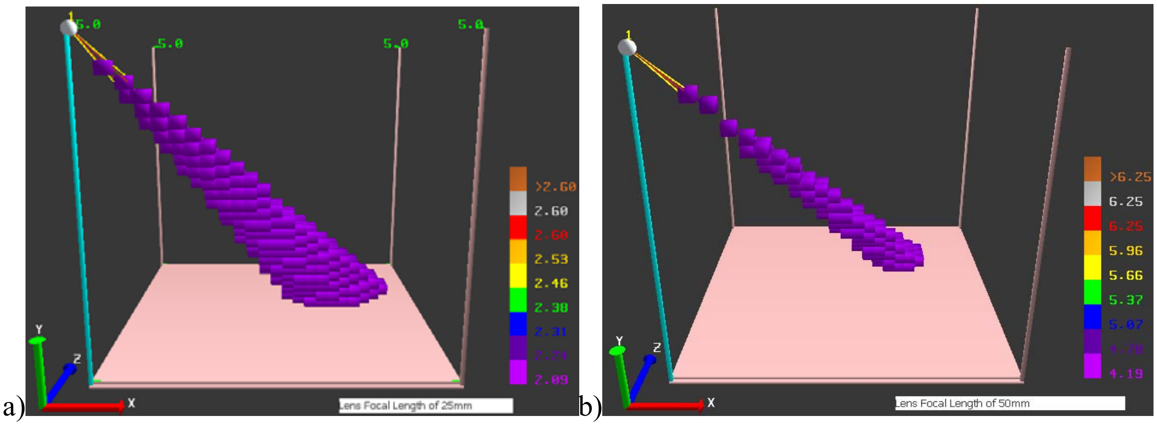


Figure 3.12: Resolution range for various focal lengths of lenses (various FOVs). a) lens of 25mm focal length. b) lens of 50mm focal length.

The functionality to analyse the resolution of a particular camera over the capture volume has not been implemented in the rudimentary stage of application development and it has been added later on. Refer to section 4.1 for the two developmental stages of the application.

3.6 *Other Colour maps*

Different colour maps were tried, prior to the implementation of the *rainbow colour map* to visualize the resolution over the capture volume. A couple of them are as shown in figure 3.14. All others are non-archived as they were modified then and there programmatically and they are prior to the implementation of one shown in figure 3.13-a. Figure 3.13-a shows an absolutely indefinite colour map. This colour map helps in identifying different ranges of data by the means of different colours but this colour map does not give any insight into the data. That is, the colour map does not help in gaining knowledge about data like, a particular volume under considerable resolution, etc., which the *rainbow colour map* helps in identifying. Figure 3.13-b shows a better colour map which gives a good insight into the data but the colours used are not natural to indicate the variation of data range. This colour map is also mapped by dividing the data range into seven parts but the colours chosen were not appropriate. Instead of colours violet and indigo maroon and dark maroon were chosen just on trial and error basis. This colour map gives the same insight that actual *rainbow colour map* gives, that is, much of the region falls into a particular range called as safer region as described in section 3.5.

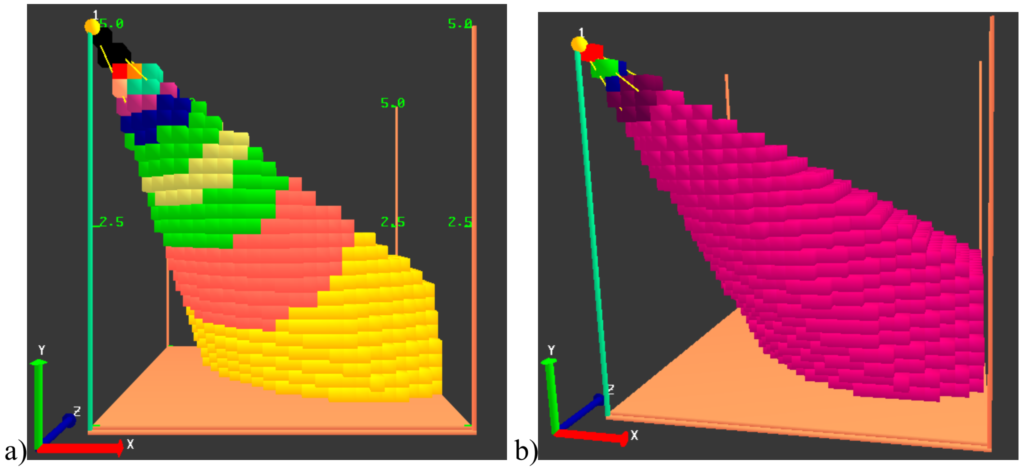


Figure 3.13: a) An indefinite colour map. b) A better alternative colour map.

But the colours violet and indigo as shown in figure 3.10 helps understand, that much of the resolution is in the *lower range* and the colour violet which is naturally less stimulating to eyes helps understand that the volume mapped with this colour is of resolution in lower range of the legend shown in figure 3.11.

3.7 *F-Number and aperture size*

The terms F-Number and aperture diameter of camera imply a single technical specification of camera which governs the amount of light entering a camera in addition to the *depth of field* of image (WorldHistory.com, 2004b). Technically it is called as F-Number and generally F-Number refers to the aperture diameter of the shutter opening when camera is clicked or when the shutters open themselves for the light to enter camera for recording image.

The F-Number expresses the diameter of the diaphragm aperture in terms of the effective focal length of lens. For example, an F-Number of f/16 represents a diaphragm aperture diameter that is one-sixteenth of the focal length of the lens. The higher the F-Number, the less the light that is admitted through the lens. (WorldHistory.com, 2004b)

Besides, F-Stops are a way of representing convenient sequence of F-Numbers in a geometric progression. Each F-Stop marked with its corresponding F-Number represents halving of the light intensity from the one before, corresponding to a decrease in diaphragm aperture diameter by √2 and hence halving the area of aperture. F-Stops make a logarithmic scale of exposure intensity. (WorldHistory.com, 2004b)

In the terms of VICON CCD cameras, the aperture controls the amount of light that reaches CCD sensor. It acts like a pupil of eye. Pupil opens wider as light decreases letting in more of the available light and it becomes smaller when the amount of light increases. (Digicamhelp.com, 2004)

Interesting description of various technical specifications can be had from WorldHistory.com (2004b).

3.8 Visualization of Capture Volume

Another most important part of the application VICON Volume Visualization is to visualize the three dimensional Capture Volume of cameras. Through the visualization user should be able to perceive the exact shape of the capture volume. The capture volume as depicted in a number of figures till now is of various shapes and sizes. In order to perceive the exact shape of this irregular volume it is of utmost importance that the volume be visualized in a way that gives perception of shape of the volume.

The capture volume has been visualized in a number of different ways before choosing *voxel visualization* eventually. Different methods used are explained below.

Volume Slicing
Volume slicing is a technique through which the whole volume is divided into slices parallel to XY or XZ or YZ planes. This technique has been thought but has not been implemented due to its naïve nature. This technique does not help in perceiving shape of the volume in an effective way. The number of slices that make up the volume can be increased but that brings up the problem of occlusion. A Slice can occlude another slice below it. Transparency provides a solution but when the numbers of slices are more transparency for all of those slices would result in a confusing image. The images on multiple slices get mixed up and provide no useful insight of the information visualized. Figure 3.14 below shows an image which provides an example for this technique of volume visualization. The image is the visualization of behaviour of air between hot and cold extremes in a piece of double glazed window panel. The two images shown in figure 3.14 are not correlated and are totally independent from each other with separate coordinate systems for each.

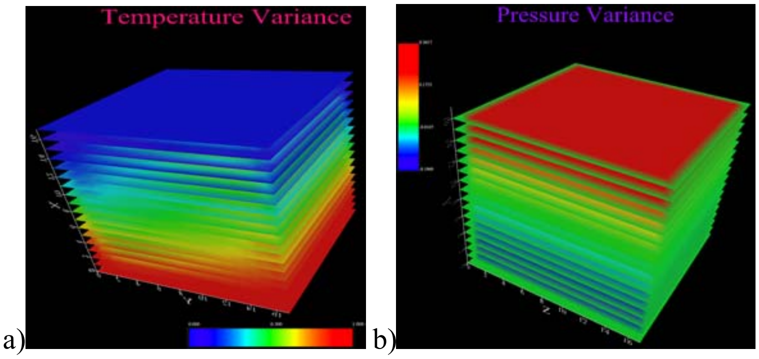


Figure 3.14: a) Volume sliced with planes parallel to YZ plane and indicating the variation of temperature from low to high from top to bottom. b) Volume sliced with planes parallel to XZ plane and indicating the variation of pressure from high to low from top to bottom.

From the above figure it is quite obvious that Volume Slicing is not a suitable way for visualizing solid volumes like the one that *VICON Volume Visualization* visualizes, because it does not provide a clear cut perception of shape of volume, which is of utmost importance for this application.

Visualization through points

Figure 3.15 below shows the whole volume of room visualized through points. This is the very first implementation of the application and the white sphere at the top right hand corner indicates a light placed in a room. As the visualization of light provides no useful insight it has been discarded in the latter implementations of the application.

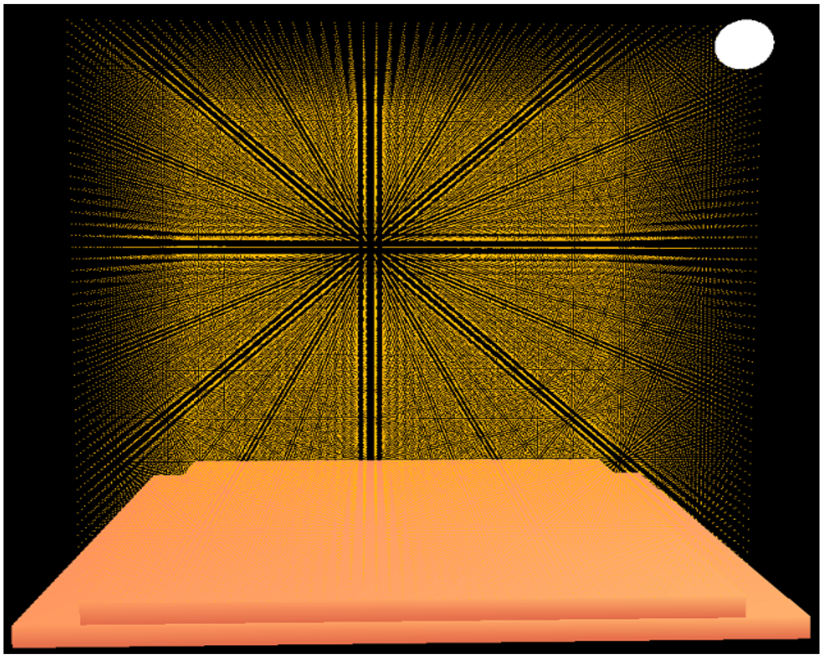


Figure 3.15: Whole volume of room visualized through the means of points. Allows user to see through the volume but difficult to perceive the shape of irregular volumes.

All of the points above are displaced with each other to the length that is equal to the length of Voxel Size described in the section 4.2.4. The very first implementation did not have any interface for manipulating this parameter. The distance between every two adjacent points depends upon the number of points desired per a metre of length. If the numbers of points that are to be visualized are more the volume looked better giving a greater accuracy but reducing the rendering speed drastically. And if the points are not densely packed the volume did not provide any information regarding its shape.

Later the point size (pixel size) has been increased in this method. This modification provided a means to make points denser but provided no means to perceive the shape of volume as colour of pixels are not shaded due to light.

Visualization through wire frame cubes

Figure 3.16 shows the volume of whole room rendered using wire-framed polygons. This implementation is also the very first of all implementations. In this method volume is visualized through the means of wire framed quadrilaterals drawn as cubes. This implementation is just prior to the final implementation of voxels. This method has been tried with an idea that it might help in seeing through the volume and providing perception of shape at the same time. But, the representation did not provide any of the above things in the better way. Neither provides seeing through the volume clearly nor provides the perception of shape.

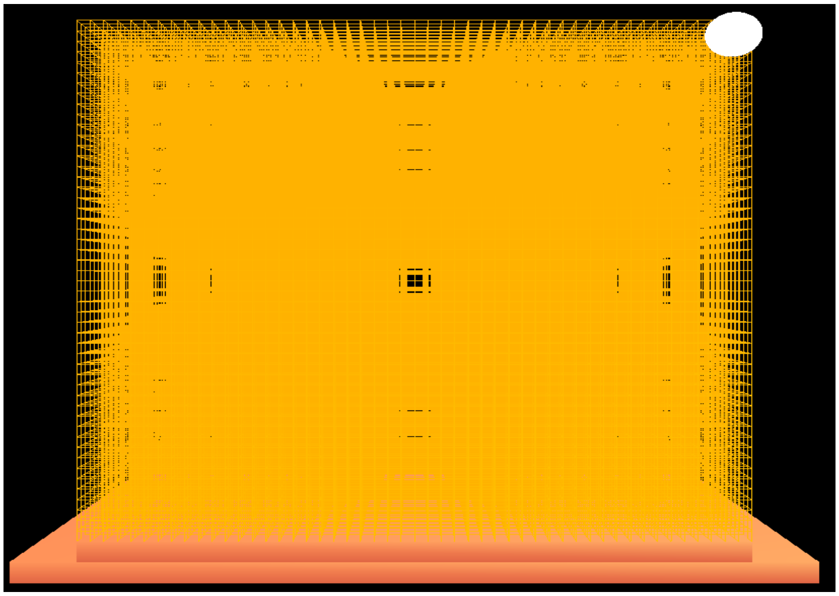


Figure 3.16: Whole volume of room visualized through the means of wire-framed cubes. Neither aids user to see through the volume nor to perceive the exact shape of irregular volumes.

After having tried the possibilities of visualizing the volume using points and wire framed cubes, filled cubes were used to represent the volume which is the final form of representations tried as it provided an exact perception of shape of volume. Although this does not provide a straight way to see through the volume, the truncation of the volume provides a means to perceive the inner details of solid volume which is described in the section 3.2.16.

3.9 Building Voxels out of Volume

The explanation till this point does not describe the way volume has been divided to form into voxels. The steps below describe the method of dividing the whole volume of room into equally sized parts which are visualized as voxels.

Variables Used: HUNDRED_CM, VOXEL_SIZE, noOfVoxels
Variable Description: HUNDRED_CM is a constant variable indicating the value 100, VOXEL_SIZE is a variable indicating the value of current size of voxel and noOfVoxels is a variable that indicates number of voxels required per a metre.

Step 1: calculate number of voxels required per one metre by the equation HUNDRED_CM / VOXEL_SIZE. Let the value be stored in noOfVoxels.

Step 2: Get the bottom left front corner point (X, Y, Z) of the room. This is the point indicated by visualizing coordinate system of the application. Consider it as **P**.

Step 3: Calculate the number of partitions that has to be made to the room length wise, width wise and height wise by the three formulae shown below,

roomXParts= floor(RoomLength * noOfVoxels),
roomYParts= floor(RoomHeight * noOfVoxels),
roomZParts= floor(RoomWidth * noOfVoxels).

Where variables roomXParts, roomYParts and roomZParts store unsigned integer forms of results obtained and function floor(…) returns a maximum least integer below the value that is passed to it.

Step 4: Create a huge one dimensional array of size equal to (roomXParts * roomYParts * roomZParts) to store floating point numbers to represent center points of voxels. Consider the array as VOXEL_CENTER_POINTS

Step 5: Compute voxel size in world coordinates by the formula VOXEL_SIZE/HUNDRED_CM. Consider it as *VoxelSizeInWrldCoords*.

Step 6: Computer half the length of side of voxel by halving voxel length in world coordinates. Consider it as *VoxelMedian*.

Step 7: Compute the center points of voxels for the whole volume of room using the loop given below

```
For(i=0; i<roomYParts; i++)
{
        For(j=0; j<roomXParts; j++)
        {
                For(k=0; k<roomZParts; k++)
                {
                        arraySubscript = i+(j*roomYParts)+(k*roomYParts*roomXParts);

                        VOXEL_CENTER_POINTS[arraysubscript].X=P.X + VoxelMedian
                        VOXEL_CENTER_POINTS[arraysubscript].Y=P.Y + VoxelMedian
                        VOXEL_CENTER_POINTS[arraysubscript].Z=P. Z + VoxelMedian

                        P.Z-= VoxelSizeInWrldCoords
                }
                P.X+= VoxelSizeInWrldCoords
                P.Z= VoxelSizeInWrldCoords
        }
        P.Y+= VoxelSizeInWrldCoords
        P.X= VoxelSizeInWrldCoords
        P.Z= VoxelSizeInWrldCoords
}
```

***Note:** The indentation of roomXParts under the loop of roomYParts builds the volume as layers of slices, which is a key for real-time volume truncation.*

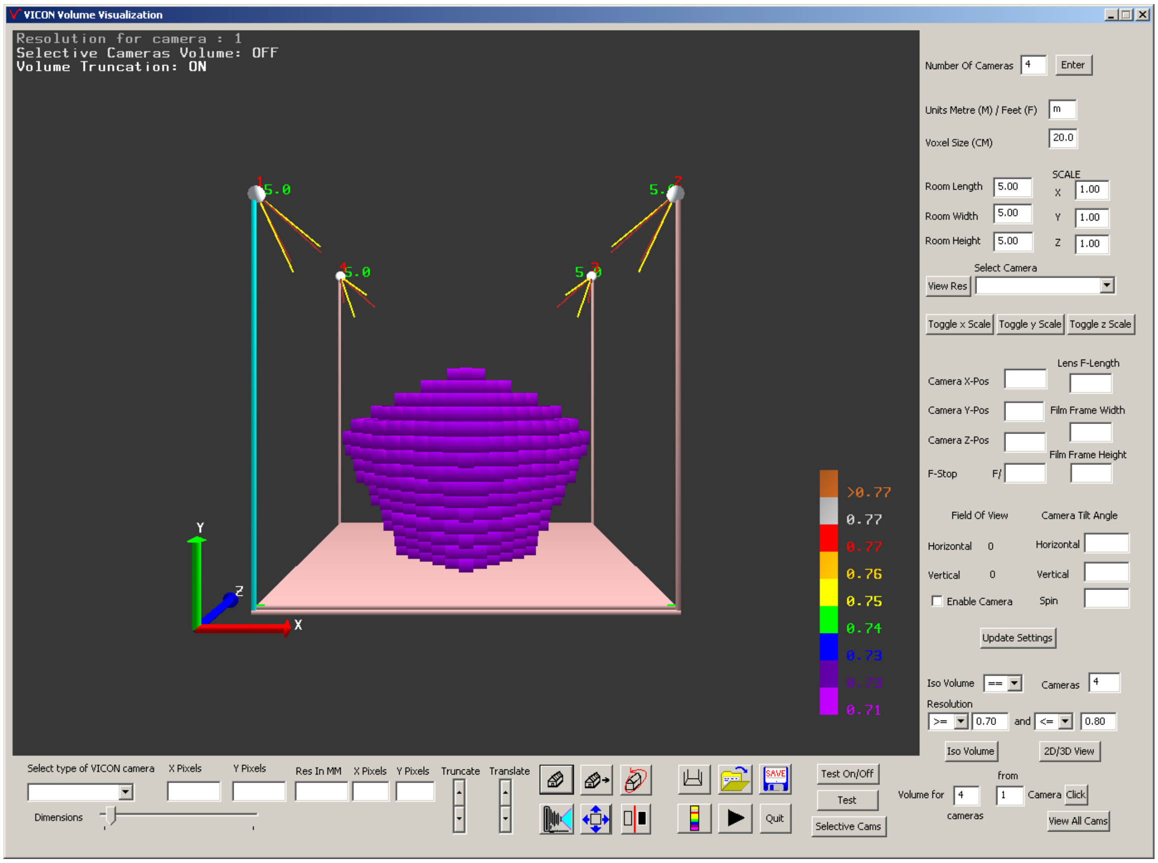
Step 8: *End*. The array VOXEL_CENTER_POINTS contains the center points of all voxels into which the whole volume of room has been divided into.

In the above procedure, step 3 might produce floating point values if the voxel is maintained in an uncommon way like 13cm 12.5cm. Hence the floor value of the result of multiplication is cast as an integer so that the room is divided into the nearest minimum number of voxels.

3.10 *System Architecture*

3.10.1 *Platform*

The application VICON Volume Visualization tool has been implemented, which is a dialog-based-window Windows application, in C++ using OpenGL application programming interface (API) and MFC class library. OpenGL v1.1, MFC v6.0 and toolkit Visual Studio .NET 2003 have been used in developing the application. The application window with four cameras placed at top four corners of room is shown in figure 3.17.



```
Resolution for camera : 1
Selective Cameras Volume: OFF
Volume Truncation: ON
```

Number Of Cameras 4

Units Metre (M) / Feet (F) m

Voxel Size (CM) 20.0

Room Length 5.00, Room Width 5.00, Room Height 5.00

SCALE X 1.00, Y 1.00, Z 1.00

Field Of View: Horizontal 0, Vertical 0

□ Enable Camera

Iso Volume == Cameras 4

Resolution >= 0.70 and <= 0.80

Volume for 4 cameras from 1 Camera

Figure 3.17: Screen-shot of the application showing the view volume common to four cameras placed at top four corners of the room.

The application is compatible with Windows XP and Windows 2000 operating systems and general system requirements for better performance are,

- PC running under Windows 2000 or Windows XP
- Intel Pentium IV
- 512 MB RAM
- Graphics card with hardware acceleration. For example NVIDIA, Radeon, etc.

3.10.2 *Program Structure*

Figure 3.18 depicts a self-elucidating class diagram of the application VICON Volume Visualization. As depicted singleton objects of the classes Colour, DataLoader and ErrorLog were created. The class DataLoader is the main class which holds data that is central to the application and hence a singleton object has been created for it, so that the data can be utilized at any location in the application. Maintaining data in static scope would provide an alternative way but maintaining huge amounts of data in static scope is not efficient and hence singleton object has been preferred. Class colour has been used to specify OpenGL colour intuitively and as it needs to be accessed at various locations in the application, it has bee instantiated as a singleton object. Similarly, a singleton object of the class ErrorLog has been created, which is used to prepare a log of all the programmatic errors during debug and it has been disabled in the final release of the application. In addition, a singleton object for the class ViconCamera has been created, which maintains data regarding various specifications of VICON cameras.

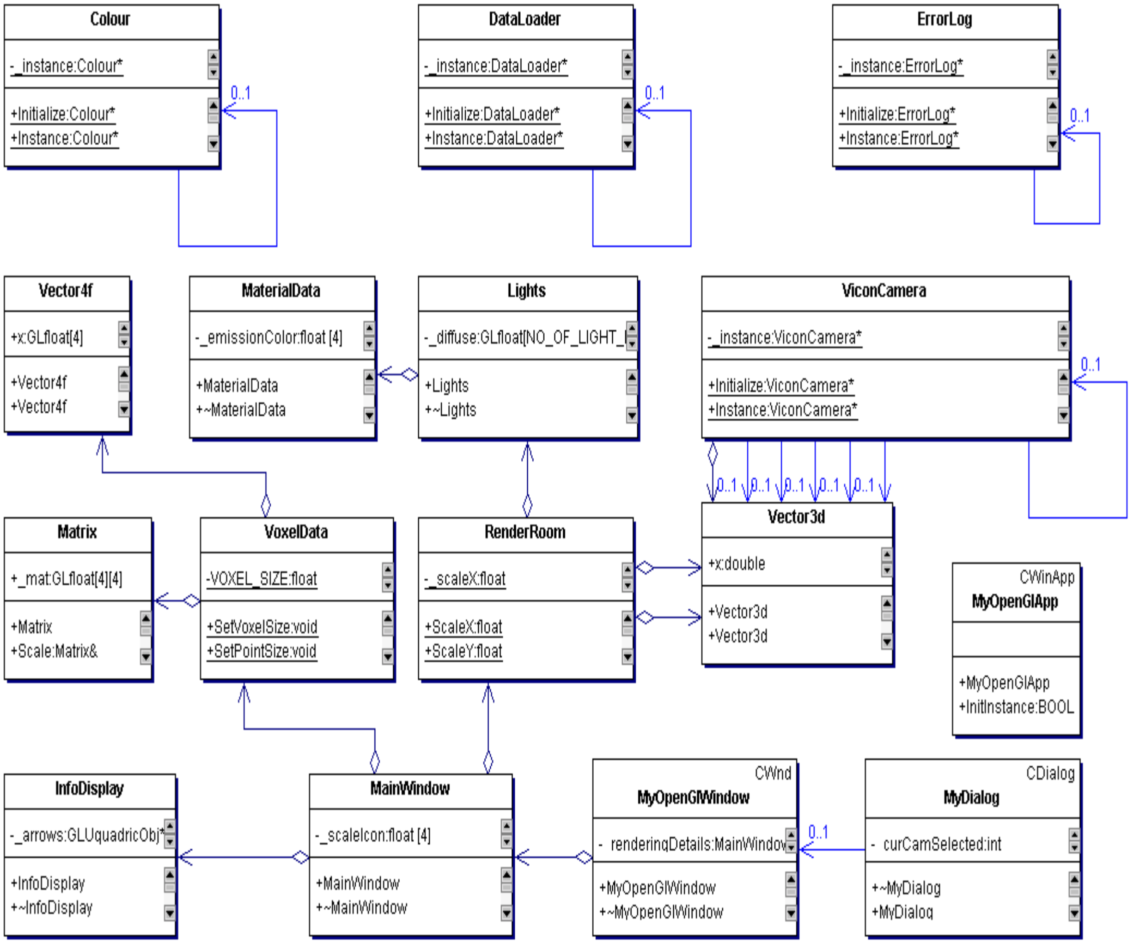


Figure 3.18: Class diagram

The class MyOpenGlApp, which is inherited from CWinApp, creates a modal dialog box using a local instance of MyDialog, which is inherited from CDialog. This creates a dialog based application. In turn, MyDialog constructs an OpenGL window with in it using the class MyOpenGlWindow, which is inherited from CWnd. With this action, an OpenGL window with in a dialog box is created. The CD accompanying this report provides full documentation for each of the classes depicted in figure 3.18.

4 *Implementation*

The application VICON Volume Visualization, as said in the chapter 1, is a dialog-based-window Windows application implemented in C++ using OpenGL and MFC class library. This chapter describes about the various methods chosen to design the application, describes the usage of the application and does critical appraisal of the methods implemented.

4.1 The main Application window

A window[3] is the main interface to display information on Windows family of Operating Systems. The information that the VICON Volume Visualization tool has to display is three-dimensional graphical information. Hence, OpenGL has been used as it provides a rich graphical set of API to render objects in three-dimensions efficiently. OpenGL API utilizes hardware capabilities of the graphics card to the maximum extent and helps in rendering large scale 3D objects with much faster frame rate.

In addition, the VICON Volume Visualization tool consists of a large set of parameters that shall need manipulation in real time. This requires a robust form of interface. On the Windows platform, Dialog Boxes with Text Boxes and other controls in it are an appropriate choice. Hence, it is necessary to have a window to display three-dimensional data along with dialog boxes and various controls placed in it, to view and manipulate the data.

For obtaining a better interface for the application, two possibilities were implemented.

- A separate Window with dialog boxes as child windows and
- A Window displayed with in a dialog box with all controls placed on the dialog box's frame.

Figure 4.1 shows the former type of interface, an OpenGL window with separate dialog boxes as child windows. Figure 4.2 shows the latter type of interface, an OpenGL window displayed with in a dialog box containing all of the necessary controls. Figure 4.2 shows the image of application window before final refinements were made. Figure 1.1 shows image of eventual application window.

Here after, all over through the report, the above two types of interfaces are referred to as,

- V1.0, (version 1.0) which indicates First type of interface and
- V1.1, (version 1.1) which indicates Second type of interface.

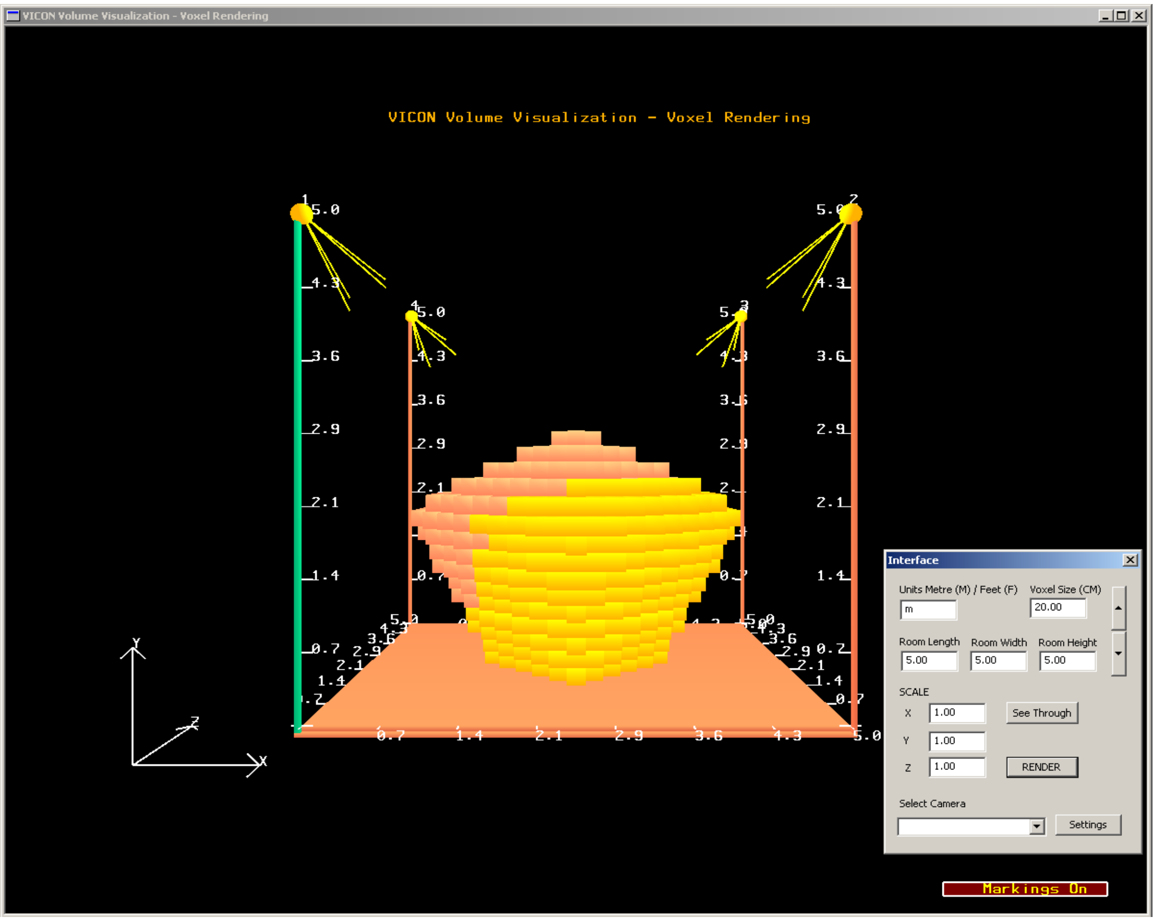
These Imply, A Window with separate dialog-boxes as child Windows and A Window displayed with in a dialog-box with all controls placed in it, respectively. In addition, the whole window which includes the OpenGL window, dialog box and all its controls is referred to as *Application Window* and the OpenGL window distinctively is referred to as *View Window*.

[3] Microsoft's GUI (Graphical User Interface) that surrounds the DOS operating system. Like other GUI interfaces, such as IBM's OS/2, Unix's XWindows, and the Macintosh' System Seven, Windows allows mouse-selectable icons, menus and data-sharing between applications. (Computer Support Center, "nd")

In order to utilize the VICON Volume Visualization tool to its maximum capacity the interface between the parameters and the user should be convenient and easily guidable. That is, user should be able to view, the parameters and the display, guided by those parameters, both simultaneously. User should also be able to manipulate the data in real-time. For the above advantages with the V1.1, it has been chosen as the final interface for VICON Volume Visualization tool.

The code samples for creating OpenGL window using MFC and creating OpenGL window with in a dialog box using MFC can be obtained from the CD accompanying this report.

The CD accompanying this report contains applications with the two types of interfaces but the application V1.0 does not include in it all of the functionalities that the application V1.1 includes.


VICON Volume Visualization - Voxel Rendering
VICON Volume Visualization - Voxel Rendering
1
2
5.0
4.3
3.6
2.9
2.1
1.4
0.7
X
Y
Z
Interface
See Through
RENDER
Settings
Markings On

Units Metre (M) / Feet (F): m
Voxel Size (CM): 20.00
Room Length: 5.00
Room Width: 5.00
Room Height: 5.00
SCALE
X: 1.00
Y: 1.00
Z: 1.00
Select Camera:

Figure 4.1: Main application window prior to the implementation of dialog box-based-window shown in figure 1.1. An interface to the data is provided through separate dialog boxes as shown above and it has been become necessary to provide the interface through multiple dialog boxes due to occlusion of scene by dialog box if it is large and contained all of the necessary controls.

4.2 User Controls

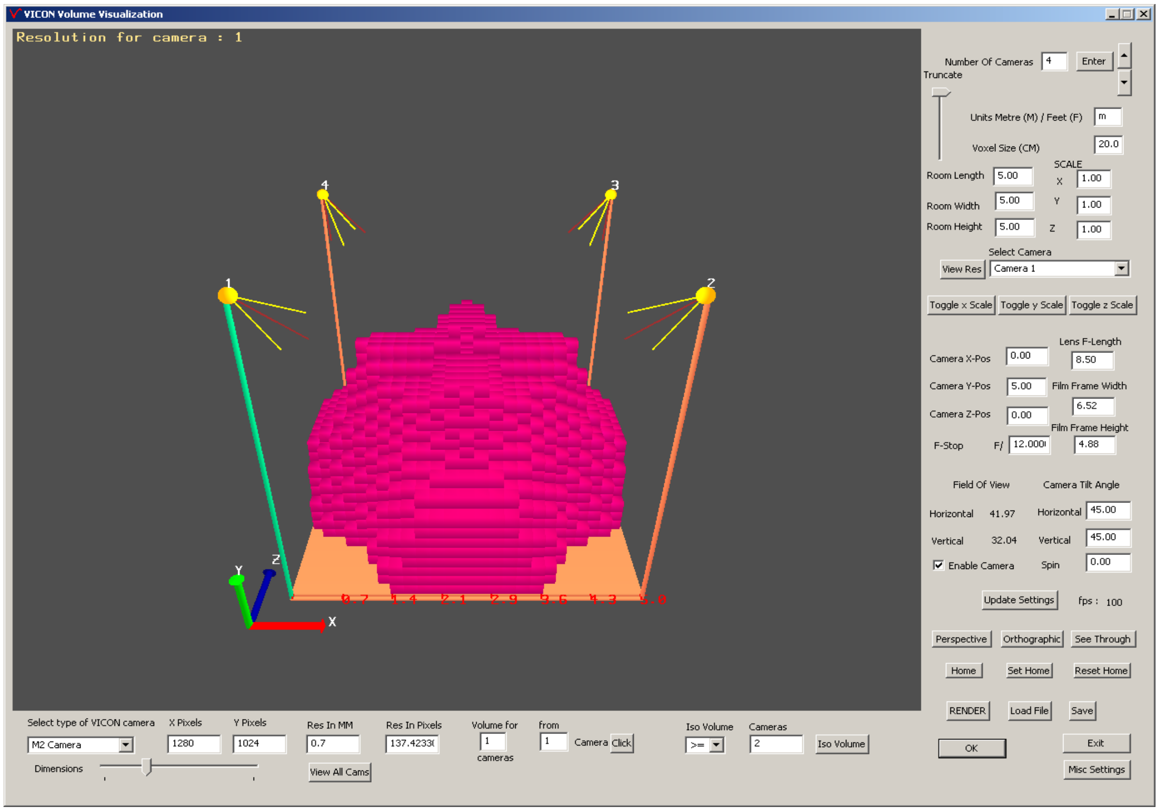
The application VICON Volume Visualization can be effectively controlled by user of any background. Various parameters governing the field of view of the cameras, their resolution at a particular point in space at a particular distance, the visualization of the capture volume, etc., can be controlled with the use of various controls provided. Their usage and suitability are described below.

4.2.1 *Start Up*

The application VICON Volume Visualization can be started up by double clicking on the executable file *VICON VolVis.exe*. This file can be identified by the ICON V and it can be obtained from the CD accompanying this report.

As soon as the application has been started up an initial window with a room (only floor and surrounding pillars) is displayed. By default dimensions of the room are set up to 5m X 5m X 5m which indicate length X width X height of the room, respectively. In addition, controls for holding the room dimensions data – Edit Text Boxes[4] – are initialized with the above values. And also, controls holding the *Scale* for dimensions of room are initialised with 1.0. This indicates that one unit in world coordinates is equivalent to 1m.

The start up screen of the application is as shown in figure 4.3. Eventually Voxels (explained in section 4.2.4) have been used to visualize the 3D volume. A number of different ways were manoeuvred before choosing voxels eventually. All of the different techniques implemented for visualizing capture volume have been described in the section 4.8.



Resolution for camera : 1

Number Of Cameras: 4
Units Metre (M) / Feet (F): m
Voxel Size (CM): 20.0
Room Length: 5.00
Room Width: 5.00
Room Height: 5.00
SCALE X: 1.00
SCALE Y: 1.00
SCALE Z: 1.00
Select Camera: Camera 1
Camera X-Pos: 0.00
Camera Y-Pos: 5.00
Camera Z-Pos: 0.00
F-Stop F/: 12.0000
Lens F-Length: 8.50
Film Frame Width: 6.52
Film Frame Height: 4.88
Field Of View Horizontal: 41.97
Field Of View Vertical: 32.04
Camera Tilt Angle Horizontal: 45.00
Camera Tilt Angle Vertical: 45.00
Spin: 0.00
☑ Enable Camera
fps : 100

Select type of VICON camera: M2 Camera
X Pixels: 1280
Y Pixels: 1024
Res In MM: 0.7
Res In Pixels: 137.42330
Volume for 1 cameras from 1 Camera
Iso Volume: >=
Cameras: 2

Figure 4.2: Main application window before refinement. This interface is providing a convenient way to deal with data. User can view the parameters and their effect, in real time.

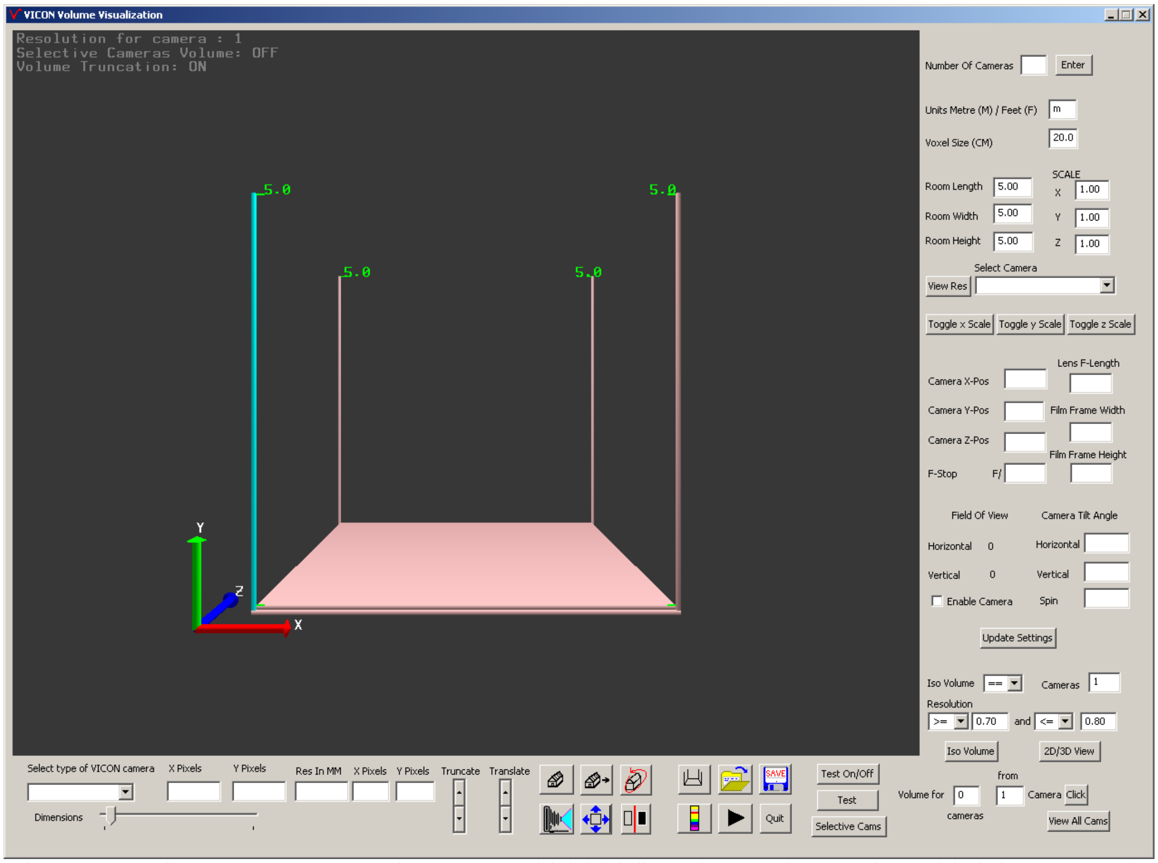
[4] Edit Text Boxes are one of the many types of controls that are provided by the Microsoft as a means of interfaces for data on windows platform. Using these data can be displayed and edited. The tutorial for creating Edit Text Boxes using MFC can be had from Hussain (1998).

4.2.2 *Initializing Number of cameras*

After the start up of the application the maximum number of cameras that the VICON system uses for a particular application shall be entered in the 'Edit Text Box' at the top right hand corner of the application which looks like as shown in figure 4.4.

User shall have to enter the number of cameras that are to be simulated and has to click on the 'Enter' button[5] next to the edit text box in order to view the capture volume common to all the cameras. That is, the volume that can be tracked by all the cameras in common.

In the application V1.0 described in the section 4.1, the interface for initializing number of cameras is through a separate dialog box which appears once at the starting of the project. This type of interface is shown in figure 4.5 and obviously is not as convenient as the interface provided in the application V1.1, in which the number of cameras in the system can be varied at any point of time.



```
Resolution for camera : 1
Selective Cameras Volume: OFF
Volume Truncation: ON
```

Units Metre (M) / Feet (F): m
Voxel Size (CM): 20.0
Room Length: 5.00 — SCALE X: 1.00
Room Width: 5.00 — Y: 1.00
Room Height: 5.00 — Z: 1.00
Horizontal: 0
Vertical: 0
□ Enable Camera
Iso Volume: ==
Cameras: 1
Resolution: >= 0.70 and <= 0.80
Volume for 0 cameras
from 1 Camera

Figure 4.3: Application start up screen shot. A room with initial dimensions can be seen along with their parameters in respective text box controls on the right hand side.

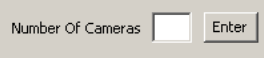


Figure 4.4: Edit text box control for entering number of cameras

[5] Buttons are another type of controls provided by Microsoft to be used as interfaces for inputting user actions on Windows platform. The tutorial for creating Buttons using MFC can be had from (Hussain, 1998) or MSDN.

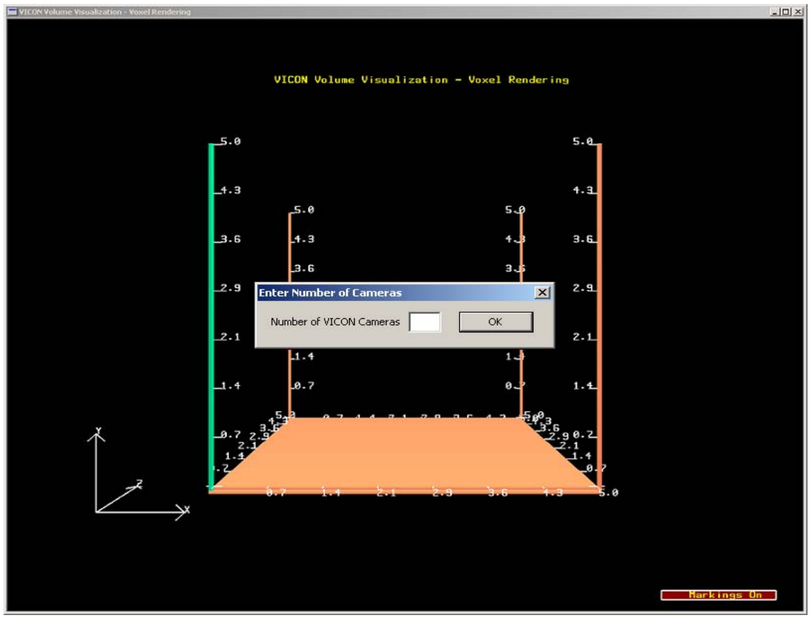
The interface shown in the figure 4.5 does not help user in modifying the number of cameras that are to be simulated. The second type of interface provides user with the *number* of cameras at all times and he can vary the number of cameras at any point of time in the simulation.

4.2.3 *Units of Dimensions*

The default units of dimensions are metres. They can also be specified in feet. The static text[6] control 'Units Metre (m) / Feet (f)' along with an 'Edit Text Box' on the right hand side the application window is the interface which displays the units of dimensions and accepts the format of dimensions from user. The control is as shown in the figure 4.6.

This control is similar in the V1.0 as well but it will not be displayed when altering the parameters of cameras technical specifications, which are displayed using another dialog box. The letter 'm' in the Edit Text Box stands for 'metres' and the letter 'f' stands for 'feet'.

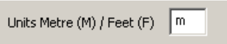
When the units where chosen as feet, all dimensions of the room are converted into equivalent metres. That is, one foot is converted into 30cm or 0.3m.


VICON Volume Visualization - Voxel Rendering
VICON Volume Visualization - Voxel Rendering
5.0
4.3
3.6
2.9
2.1
1.4
0.7
0.7
1.4
2.1
2.9
3.6
4.3
5.0
X
Y
Z
Markings On

Enter Number of Cameras

Number of VICON Cameras [ ] OK

Figure 4.5: A separate dialog box for initializing Number of cameras needed to simulate, in the first type of interface.



Units Metre (M) / Feet (F) m

Figure 4.6: Interface for units of dimensions

4.2.4 *Voxel size*

The 'Voxel Size' control on the right hand side of the application window displays current voxel size and allows user to change its size. A voxel is a short form of ***Volume Pixel*** (Webopedia, 2004). A pixel indicates a picture element (the smallest point on the screen). A voxel represents a distinguishable cube of certain size in 3D space. Voxels are one of the many types of methods to visualize the volume in 3D. A cluster of voxels can be used to represent an irregular volume of space in 3D.

[6] Static Text controls are another type of controls provided by Microsoft which are used as a means to display data which is not necessary to be modified by user. That is the data can be seen but cannot be edited by user. The tutorial for creating static text boxes can be had from (Hussain, 1998) or MSDN.

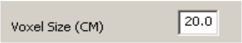

Voxel Size (CM) 20.0

Figure 4.7: Interface for manipulating voxel size

The voxel size is in centimetres (cm). One centimetre represents 0.01 units in world coordinates. The units 'centimetre' is the better way to quantify the voxel size as it is intuitive. It would be uncommon to have voxel size represented in metres or feet as done for room dimensions and they are also such small entities that 'centimetres' would be the correct units to quantify them.

Voxel size can also be represented in 'inches' when 'Units of Dimensions' has been changed to feet and in 'centimetres' when 'Units of Dimensions' has been changed to metres. But in the user group 'centimetre' is a more commonly used word than 'inch' to measure small entities. Hence for ease of users the voxel size is implemented in 'centimetres' alone.

The description above is purely in the terms of computer graphics and does not correlate voxels to the VICON system. But there is a relation between this functionality and the VICON system. The relation is, the *marker size*. More or less the size of a voxel can be correlated with the maximum marker size in use by the subjects. But it might not be necessary to consider the marker size at all times and in all applications. Because it is obviously true that if particular volume has been shown by the application that is being tracked by certain number of cameras, the markers in that volume will be tracked regardless of their size.

In contrast, this functionality finds appropriate usage when accurate volume has to be identified at the boundaries of the capture volume that is being visualized by the application. There exists a tolerance for this application which is equivalent to half the length of voxel and hence if it is necessary to find the accurate boundaries of the capture volume the voxel size should be minimized. Accuracy increases with the decrease of voxel size. But there is a trade-off between the voxel size and the speed of the system. As the voxel size reduces, accuracy of the system increases but at the same time the speed of the system reduces.

On the basis of trial and error method voxel size of 20cm has been chosen which gives a good balance between the accuracy and speed of the system.

4.2.5 ***Translating Image Back and Forth***

The basic need for any image is to be able to navigate through. Moving forth and back is one of the primary navigations that a user wants to perform with an image along with other types of navigations. The 'Spin Control'[7] entitled "Translate" at the bottom of the application window is the control which allows user to move image (consisting room, cameras and volume) forth and back. The control is as shown in the figure 4.8. Figure 4.8-a represents an unrefined style of control implemented in the starting stage. Figure 4.8-b shows an eventual style of spin control implemented.

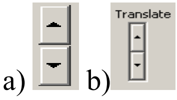


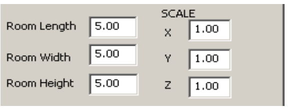
Figure 4.8: 'Spin Control' control. a) Unrefined b) Refined

[7] Spin Control is a type of control provided by Microsoft which is commonly used for incrementing and decrementing a value by clicking on the pair of arrows. It is also known as Up-Down control. Good introduction to spin controls can be had from MSDN (Microsoft1, 2004).

In the very basic implementation of this application *this* navigation is provided through simple hits of keyboard keys 'Up Arrow Key' and 'Down Arrow Key'. But the spin control proved to be an easy to use interface for navigating through the image than pressing keys on keyboard.

4.2.6 *Room Dimensions and Scale*

On the right side of the application window along with all other types of interfaces, an interface to view and update dimensions of the room and an interface to view and update the scale of those dimensions are provided. These controls are as shown in the figure 4.9 below.



Room Length 5.00
Room Width 5.00
Room Height 5.00

SCALE
X 1.00
Y 1.00
Z 1.00

Figure 4.9: Interface for viewing and updating dimensions of room and scale of dimensions of room

The interfaces are Edit Text Boxes which help view and update the data. The static text boxes indicate the respective *type* of parameters in edit text boxes.

The units of dimensions of the room are in metres (m) by default and if the units have been changed to feet through the 'Units of Dimensions' control the parameters in the text boxes shown in figure 4.9 indicate dimensions in feet.

'Scale' indicates the scale of dimensions of room with respect to the world coordinates. By default the scale is 1.0 about all the three axes and it means that 1m is equivalent to 1 unit in world coordinates about all axes. If the scale has been changed to 2, 1m is equivalent to 2*1 (2) units in world coordinates and if the scale is 1.2, 1m is equivalent to 1.2*1 (1.2) units in world coordinates and similarly if the scale is 0.5, 1m is equivalent to 0.5*1 (0.5) units in world coordinates about the axis for which the scale has been changed. The control provides the flexibility to vary the scale for all three axes independently.

In the application V1.0 also, similar controls were used but they will not be displayed at all times of the usage of the application which, might cause inconvenience in some situations. In contrast, the application V1.1 displays the parameters and allows users to update the values at any point of time. Hence the same controls were adopted in the second type of interface but presented in a different way to user.

4.2.7 *The Coordinate System*

The application uses its own coordinate system which is shown in all the screen shots of the application here up to. It has been indicated by a three axes coordinate system at the bottom left hand corner of the room. In the application V1.0, as shown in figure 4.1, the coordinate system is drawn using simple arrows which, is also shown in figure 4.10.

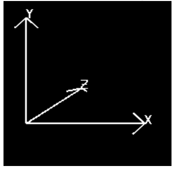


Figure 4.10: The coordinate system as used in the application V1.0

And in the *second type of interface*, the type of technique used to display coordinate system is similar to the one used in many graphics designing packages like 3D studio max etc., which is easily guidable than the one used in the *first type of interface*. The later type of axes display for coordinate system used is as shown in figure 4.11 below.

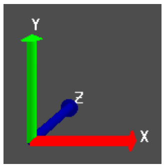


Figure 4.11: The coordinate system as used in the application V1.1.

As figures 4.10 and 4.11 indicate the XZ plane form the horizontal plane with Y axis pointing upwards. The reason that it has been displayed at the front bottom left hand corner of the room is that this point of the room falls as the reference point for dimensions of the room and positions of cameras.

Example:

If the room dimensions are 5m X 5m X 5m which indicate length X width X height of the room, respectively, they are rendered as,

room with length of 5m parallel to X axis,
room with width of 5m parallel to Z axis and
room with height of 5m parallel to Y axis.

If a point in the room has to be specified for positioning the camera or for any other purpose, the front bottom left hand corner of the room has to be taken as the reference point. For example, if a camera has been positioned at the point 2m X 2m X 2m, it has to be perceived as 2m in X axis, 2m in Z axis and 2m in Y axis.

4.2.8 *Display of Dimensions and their controls*

Dimensions of the room are displayed numerically along the floor and pillars of the room as shown in figure 4.12. They are displayed in the respective colours of the coordinate system. That is, length in the colour of ‘red’, width in the colour of ‘blue’ and height in the colour of ‘green’ as indicated by the coloured coordinate system axes.

Controls

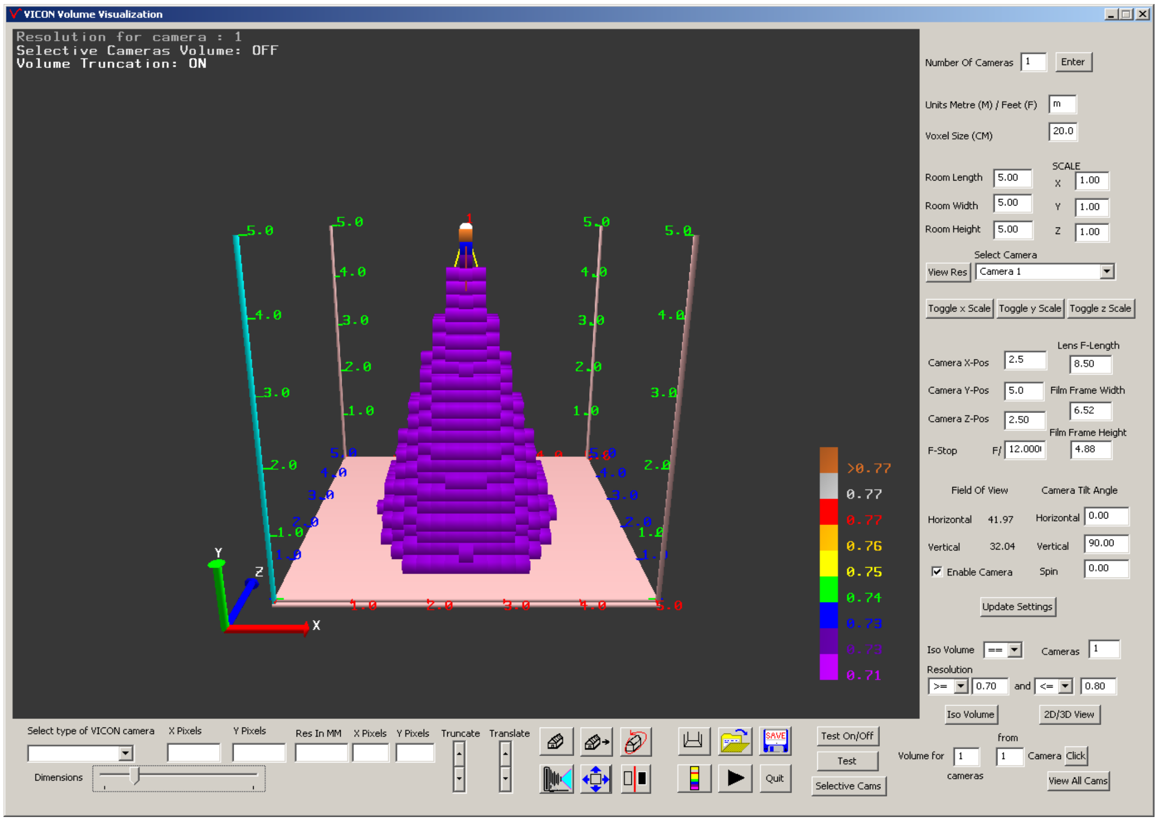
User has the ability to control, the display of dimensions about each axis and the number of partitions for the display of dimensions.

Display of dimensions

The three ‘button’ controls ‘Toggle X axis’, ‘Toggle Y axis’ and ‘Toggle Z axis’ in the middle portion of the right hand side of the application window allow user to toggle the display of dimensions about respective axes indicated by ‘button’ controls. The three ‘button’ controls are shown in figure 4.13.

4.2.9 *Number of partitions for dimensions*

The application VICON Volume Visualization also provides user the ability to control the number of partitions that are to be marked or displayed. For example, if the length of the room is displayed in only two parts, the first marking, indicates half length of the room and second marking indicates full length of the room. In figure 4.12 the number of partitions are four and hence five markings are displayed about each axis. User can vary the number of markings and view the effect in real time with the help of the *Slider Control*[8] provided at the bottom left hand corner of the application window. The control appears as shown in the figure 4.14 below.



VICON Volume Visualization

```
Resolution for camera : 1
Selective Cameras Volume: OFF
Volume Truncation: ON
```

Number Of Cameras: 1

Units Metre (M) / Feet (F): m

Voxel Size (CM): 20.0

Room Length: 5.00 — SCALE X: 1.00

Room Width: 5.00 — Y: 1.00

Room Height: 5.00 — Z: 1.00

Select Camera: Camera 1

Camera X-Pos: 2.5 — Lens F-Length: 8.50

Camera Y-Pos: 5.0 — Film Frame Width: 6.52

Camera Z-Pos: 2.50 — Film Frame Height: 4.88

F-Stop F/ 12.000

Field Of View — Horizontal 41.97, Vertical 32.04

Camera Tilt Angle — Horizontal 0.00, Vertical 90.00, Spin 0.00

☑ Enable Camera

Iso Volume == — Cameras 1

Resolution >= 0.70 and <= 0.80

Volume for 1 cameras from 1 Camera

Figure 4.12: Screen shot of application window showing dimensions of the room.

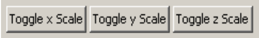


Figure 4.13: Three Button Controls for controlling the display of dimensions of room about three axes

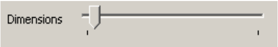


Figure 4.14: The Slider Control to vary the number of markings for dimensions of the room.

In the application V1.0 no control has been provided for varying the number of markings for the dimensions due to the size constraints of the dialog box and its overlap over the view window displaying the volume, initially.

In the application V1.1 the slider control provides a way to manipulate the number of markings on each axis. And also, the time taken for updating of number of markings in the view window after user changed the position of the slider button is negligible and hence slider control has been chosen as the interface type which is also easy to use. The number of markings varies after user changes the position of the slider button and releases the mouse button.

[8] Slider control is a Microsoft Windows control equipped with a small bar, also called a thumb, that slides along a visible line (FunctionX, 2003). The tutorial for using this controls can be had from (FunctionX, 2003) or MSDN.

4.2.10 *Selecting a Camera and viewing its resolution*

The total number of cameras that have been initialised in the system, using the control 'Number of cameras' as described in the section 4.2.2, will be listed in the Combo Box[9]. The combo box is as shown in the figure 4.15 below and is in the middle portion of the right hand side of application window.

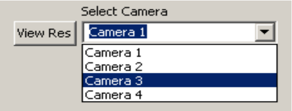


Figure 4.15: Combo Box listing the cameras for user to select along with the Button 'View Res' which stands for View Resolution.

This control proves to be a better way of listing the cameras to user to select as it does not occupy much space on the dialog box than it was initially assigned unless it has been opened to select a camera.

Viewing Resolution

Each camera has its own resolution depending upon its technical parameter focal length and the resolution will vary from point to point in space depending upon its distance. For visualizing the resolution, a colour map is over laid over the volume which depicts the resolution of a particular camera selected. The colour map is fixed and a legend for the colour map will be displayed at the bottom right hand corner of the view window. By clicking on the button 'View Res' displayed immediate left to the Combo Box shown in figure 4.15, the resolution will be mapped over the volume for the camera selected. It might take a while before the new colour map has been overlaid over the capture volume, as it depends upon the dimensions of the room and the volume of the total *capture volume* being displayed. The colour map and its suitability are described in detail in the chapter 3.

4.2.11 *Camera Parameters – viewing and updating*

Viewing and updating camera parameters is the most important of all the controls that are provided in the application *VICON Volume Visualization*. These controls can be used to view and update the parameters of a camera which govern the field of view, orientation, position and other parameters of it. The whole set of controls for camera parameters are as shown in figure 4.16-a.

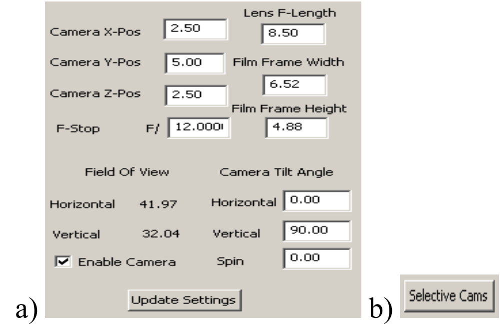

Camera X-Pos 2.50
Camera Y-Pos 5.00
Camera Z-Pos 2.50
F-Stop F/ 12.000
Lens F-Length 8.50
Film Frame Width 6.52
Film Frame Height 4.88
Field Of View
Horizontal 41.97
Vertical 32.04
Camera Tilt Angle
Horizontal 0.00
Vertical 90.00
Spin 0.00
☑ Enable Camera

Figure 4.16: a) Controls for viewing and updating camera parameters. b) Control for toggling *Selective Cameras Volume* option

[9] Combo Box is a Microsoft Windows control which consists of a list box with either static control or edit control. It provides a means to list various types of data to user. The tutorial for working with Combo Boxes can be had from MSDN at (Microsoft 2, 2004).

As soon as user selects a camera from the Combo Box shown in figure 4.15, parameters pertaining to the camera selected are displayed in the respective fields and those parameters are as explained below.

The controls ‘Camera X-Pos’, ‘Camera Y-Pos’ and ‘Camera Z-Pos’ define the position of camera in the room with respect to the reference point as explained in the section 4.2.7. The controls ‘Lens F-Length’, ‘Film Frame Width’ and ‘Film Frame Height’ define the focal length[10] of the camera, width and height of the CCD sensor, respectively. The control ‘F-Stop’ indicates F-Number of the camera. F-Stop numbers are generally denoted as F/number which implies 1/number times the Focal Length of lens. F-Stop is the measure of the diameter of the diaphragm aperture of lens in terms of its focal length. This in common words specifies the size of the aperture at lens through which light enters the camera (worldHistory.com, 2004b).

Field Of View

The Field of View controls shown in figure 4.16-a are static text box controls and help user view the horizontal and vertical Fields of View (FoVs) of camera. The horizontal and vertical FoVs of camera are dependant upon the focal length of the camera, CCD sensor width and CCD sensor height.

Whenever the values in any of the, focal length or film frame width or film frame height controls are varied the resultant horizontal and vertical fields of view for the respective values are displayed immediately beside the ‘Horizontal’ and ‘Vertical’ static text controls as shown in figure 4.16-a. As the camera specifications are specified in the terms of focal length, *CCD sensor* or *film frame* dimensions which in-turn govern the values of field of view, the controls for horizontal and vertical fields of view are made static as it is not a general way to manipulate the camera FOV parameters. As user enters the technical details of the camera in the respective fields the resultant horizontal and vertical FoVs of the camera are displayed for analysis.

Camera Tilt Angle

The Edit Text Box controls ‘Horizontal’, ‘Vertical’ and ‘Spin’ under the heading ‘Camera Tilt Angle’ specify the orientation of the camera around Y, Z and X axis, respectively, at the position specified, in the 3D space of room, with respect to the coordinate system described in the section 4.2.7.

The control ‘Horizontal’ specifies the horizontal tilt angle of the camera around Y axis in Counter-Clock-Wise (CCW) direction.

The control ‘Vertical’ specifies the vertical tilt angle of the camera around Z axis in Clock-Wise (CW) direction. The reason for choosing the CW direction for Vertical tilt angle is that this direction is intuitive to general user group to deal with vertical tilt angles. To elucidate, it is the general way of indicating angles made by bodies placed at some elevated places. And also, in general, most of the VICON cameras are placed in the top portions of a room.

The control ‘Spin’ specifies the tilt of the camera around itself and in VICON Volume Visualization application, this values indicates spin around X axis in CCW direction.

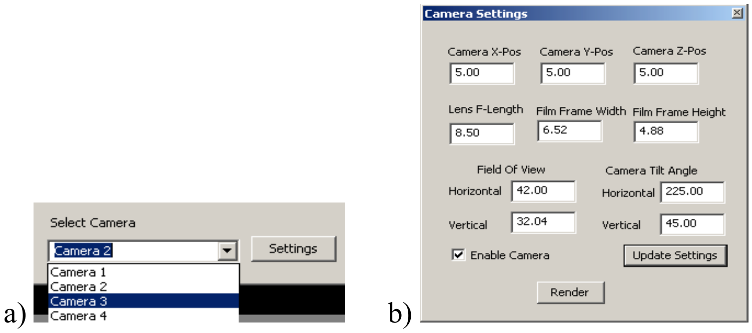
[10] The Focal length of a lens is the distance along the optical axis from the lens to the focus (focal point) (WorldHistory.com 1, 2004).

Enabling / Disabling the camera
The Check Box[11] control with the title 'Enable Camera' as shown in the figure 4.16-a indicates whether a particular camera is enabled or disabled. After a camera has been selected using the control described in the section 4.2.10, if the camera is checked, it indicates that, that particular camera is enabled and contributes to the capture volume that is displayed in the view window. If the check box is cleared, it indicates that, that particular camera is disabled and does not contribute to the capture volume that is displayed in the view window.

If the previously set option is toggled, the button control 'Selective Cams', which is shown in figure 4.16-b, has to be clicked and it is necessary to verify whether the *Selective Cameras Volume* option on the top left hand corner of the view window, which is shown in figure 4.12, shows *on* or *off* next to it. If it is *off* it has to be toggled to *on* by clicking on the button 'Selective Cams'. This implies that the volume that is being built is for particular selected cameras. And, if this option is turned *on* then *Volume Truncation* option, which is shown in figure 4.12, will automatically be set *off* by the application. This can be set *on* by clicking again on the button control 'Selective Cams'. This infers that, volume built for selected number of cameras cannot be truncated. The same applies to the functionality *Volume for certain cameras* explained in the section 4.2.12.

Updating
To save the updated parameters, which are explained above, the button 'Update Settings' as shown in the figure 4.16 has to be clicked on.

The controls for manipulating camera parameters in the application V1.1, which is described above, prove to be a better way over the interface that was implemented in the application V1.0. In the application V1.0 a separate dialog box appears, when a particular camera has been selected, for manipulation of its parameters. These are depicted the figure 4.17 below.



Select Camera: Camera 2

Camera X-Pos: 5.00
Camera Y-Pos: 5.00
Camera Z-Pos: 5.00
Lens F-Length: 8.50
Film Frame Width: 6.52
Film Frame Height: 4.88
Field Of View
Horizontal: 42.00
Vertical: 32.04
Camera Tilt Angle
Horizontal: 225.00
Vertical: 45.00
☑ Enable Camera

Figure 4.17: Selection of camera and manipulation of its parameters in the *first type of interface.*

The figure 4.17-a shows the selection of a camera through the means of a Combo Box which is similar to the application V1.1 as explained in the section 4.2.10. After a particular camera has been selected a dialog box as shown in the figure 4.17-b is displayed with parameters of the camera selected. The 'Camera Settings' dialog box as shown in figure 4.17-b does not include all the parameters that have been provided in the application V1.1. For example, camera spin wasn't implemented. If user updates the Horizontal and vertical fields of view, they will disregarded and values based on previous 'lens focal length', 'film frame width' and 'film frame height' were calculated and displayed in respective fields unless the button 'Update Settings' has been clicked.

[11] Check Box is a control which is used to input binary type of data like switch on/off, for example. Tutorial for using Check boxes can be had from MSDN at (Microsoft 3, 2004)

Hence, the application V1.1 provides a better interface to manipulate the parameters of camera than the application V1.0.

4.2.12 *Volume for certain cameras*

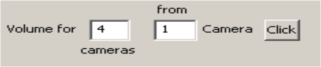
As soon as the application has been started and initialized with the number of cameras that a VICON system includes and need to be simulated, the volume that is shown in the view window is the volume that is common to all the cameras in the system.

The *VICON Volume Visualization* tool provides user with the functionality to view volume for certain number of cameras from a certain camera, serially, with out actually disabling them, which, is explained in the section 4.2.11.

This functionality is particularly useful if the system includes a large set of cameras and there is necessity to analyse the capture volume of certain number of cameras.

This task can be achieved by enabling/disabling cameras as explained in section 4.2.11, but if the system includes large number of cameras, enabling/disabling them is a time consuming task and the current functionality proves to be a better approach for the task.

Figure 4.18 below shows the interface for this functionality in the application V1.1. Certain number of cameras has to be entered in the left Edit Text Box control and the starting camera has to be entered in the right Edit Text Box control and the Button control 'Click' has to be clicked on for the change to take effect.



Volume for 4 cameras from 1 Camera

Figure 4.18: Interface for selecting certain number of cameras from a certain camera, serially.

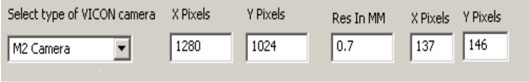
And, if all the cameras have to be enabled, a click on the 'View All Cams' button control is sufficient. This button is shown in figure 4.19 and is located at the bottom right hand side of the application window.

Figure 4.19: Button control for enabling all cameras in the VICON system.

In contrast to the application V1.1, application V1.0 does not include this functionality. Hence the application V1.1 provides a better interface for a user than the application V1.0.

4.2.13 *Selecting a VICON camera for its resolution*

Figure 4.20 depicts the interface which lets user to select a type of VICON camera and know the resolution of the capture volume in pixels. The *VICON Volume Visualization* tool depicts the range of the resolution of the whole volume of the room for a particular camera selected.



Select type of VICON camera	X Pixels	Y Pixels	Res In MM	X Pixels	Y Pixels
M2 Camera	1280	1024	0.7	137	146

Figure 4.20: Interface for selecting a type of VICON camera for analysing the resolution of volume in pixels

The Combo Box under the title 'Select type of VICON camera' provides a list of all generally available VICON camera types listed in VICON ("nd"b), VICON3 ("nd"c) and VICON ("nd"d). As user selects a camera the resolution of the camera CCD sensor is displayed in the Edit Text Box controls under the titles 'X Pixels' and 'Y Pixels' on the left of 'Res In MM'. It is necessary for user to key in the resolution in millimetres (mm) for he has to view the resultant resolution in pixels for that value of resolution.

The value of the resolution that user needs to key in the Edit Text Box control under the title ‘Res In MM’ is the value of resolution for a generic camera irrespective of whether the camera being used is digital or opto-mechanical with CCD sensor and film frame as photo sensitive parts, respectively. The value indicates the *spread* of a point in space *on* the film frame at the back of the lens. Units for this value are millimetres (mm) and represents size in both horizontal and vertical direction. This has been explained in detail in the chapter 3.

The resultant resolution in pixels for the value entered will be displayed in the Edit Text Box controls titled ‘X Pixels’ and ‘Y Pixels’ to the right of ‘Res In MM’, which represent the spread of a point in horizontal and vertical directions, respectively. The resultant resolution can be written in standard form as X * Y.

For example, in the figure 4.20 above, the type of VICON camera selected is M2 Camera and the maximum resolution of the camera’s CCD sensor is 1280 X 1024 pixels (VICON, “nd”b). The resolution value keyed in was 0.7mm and it resulted in 137 X 146 pixels. Hence, each point in the capture volume for which the resolution is 0.7mm will span 137 X 146 pixels.

In addition and as a means of guide, the resolution range for the whole volume of the room will be displayed in the right hand bottom corner of the view window. This legend which is shown in figure 4.21, aids user in knowing the maximum and minimum resolution of a camera that has been selected with in the room. The calculation of maximum and minimum resolutions is explained in chapter 3.

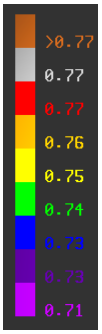


Figure 4.21: Legend for the resolution colour map. Guides to know the maximum and minimum resolution of a camera selected.

Figure 4.21 above shows the maximum and minimum resolution of camera located at the top front left corner of the room with dimensions 5m X 5m X 5m. The camera numbered ‘1’ in the figure 4.2 is the camera considered for the resolution map that is shown the figure 4.21. Here, the maximum resolution is 0.77 mm and the minimum resolution is 0.71 mm for the camera with 41.97 degrees as horizontal field of view and 32.04 degrees as vertical field of view for a minimum distance of half the voxel size and for a maximum distance of diagonal length of the room, respectively and with focal-length of 8.5mm.

This is an added functionality in the application V1.1. Hence, it provides a better set of functionalities for analysing the capture volume of VICON cameras.

4.2.14 *Rendering Iso-Volume and selecting range for resolution*

Rendering an Iso-Volume for the desired number of cameras, within the range of total number of cameras initialized and within a particular resolution range, is the most important of all functionalities that are provided. This functionality provides user the flexibility to analyse the capture volume in detail. For, example the figure 4.22-a below depicts the capture volume, of at least 2 cameras and with in the resolution range 0.71mm and 0.73mm (resolution settings by default) of first camera and with a room of 5m X 5m X 5m dimensions and cameras placed at positions as shown in the figure 4.2. And figure 4.22-b depicts the capture volume, of at least 2 cameras and within the resolution range 0.71mm and 0.72mm of the first camera.

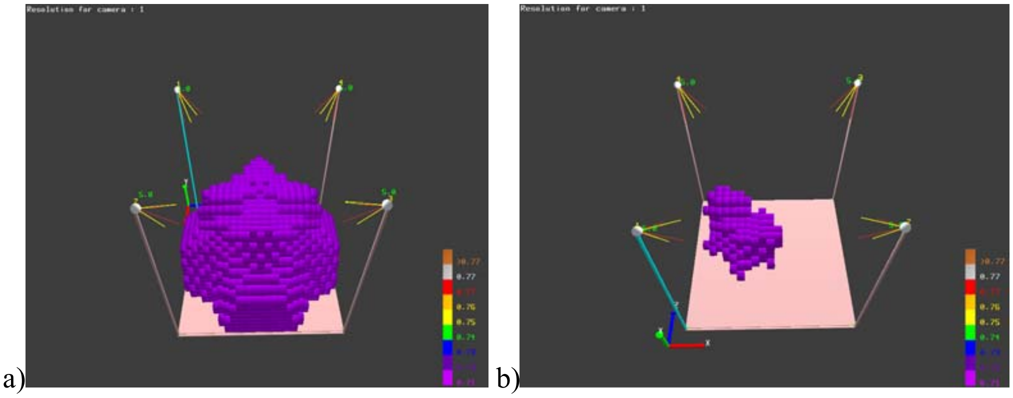
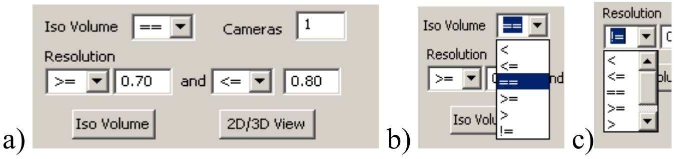


Figure 4.22: a) Iso-Volume of capture volume of at least two cameras with in the resolution range of 0.70mm and 0.73mm as indicated by legend for colour map. b) The same volume but with in the resolution range of 0.72mm and 0.73mm.

Here, the maximum resolution is 0.77 mm and the minimum resolution is 0.70 mm for the camera with 41.97 degrees as horizontal field of view and 32.04 degrees as vertical field of view for a minimum distance of half the voxel size and for a maximum distance of diagonal length of the room and with focal-length of 8.5mm.

As depicted in the legend of colour map the whole volume is mapped with violet colour, indicating the minimum resolution of, the camera selected.

The interface for this functionality is located at the bottom right hand side of the application window and is shown in figure 4.23 below. The drop down menu for the Combo Box entitled ‘Iso Volume’ lists all the controls that are available for generating the iso-volume. An iso-volume can be generated for exactly ‘n’ number of cameras (= =), for at most ‘n’ number of cameras (<, <=), for at least ‘n’ number of cameras (>, >=) and for not equal to ‘n’ number of cameras (! =). The ‘n’ number of cameras has to be specified in the Edit Text Box control entitled ‘Cameras’.



Iso Volume == Cameras 1

Resolution >= 0.70 and <= 0.80

Figure 4.23: a) Interface for rendering Iso-Volume and selecting resolution range. b) Drop down menu to select a control for generating iso-volume. c) Drop down menu to select a control for resolution range.

The Combo Box entitled ‘Resolution’ lists all the controls that are available for selecting the resolution range. The options in the Combo Box are <, <=, ==, >=, < and !=. The resolution range can be selected in different ways. For example >=x and <=y, <=x and >=y, >x and <y, <x and >y, !=x and >y, !=x and <y etc., where x and y are values for resolution that user has to enter in

the two Edit Text Box controls provided next to the two consecutive Combo Box controls under the title ‘Resolution’.

This functionality hasn’t been included in the application V1.0 and hence the application V1.1 obviously provides much more ways to analyse the capture volume of VICON cameras.

4.2.15 *Two Dimensional view of the volume*

The capture volume that has been depicted in figure 4.22 can be viewed in 2D. That is the whole volume will be visualized in 2D slices. Figures 4.24-a and 4.24-b below show the volume that is shown in figure 4.22-a in 2D.

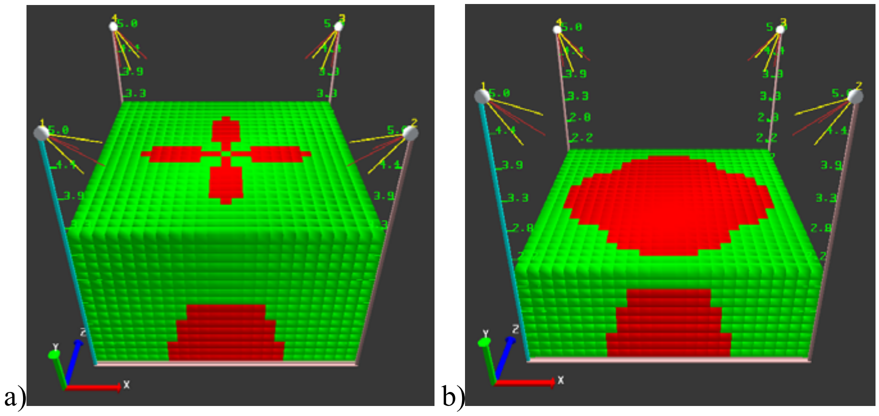


Figure 4.24: a) Volume truncated at top portion of the volume. b) Volume truncated at the middle portion of the volume.

In this method the volume with in a particular range of resolution values cannot be visualized and only depicts the volume irrespective of the range of resolution selected.
The 2D way of visualization of volume is obviously not as intuitive as the volume visualized in 3D. This is obvious by look. The 2D way doesn’t aid in viewing the whole volume and it has to be truncated slice by slice to analyse the whole volume and even that truncation might not give an insight of 3D volume. But it has its own advantage. If it is necessary to view volume of certain number of cameras at a certain height of the room, truncating the 2D volume up to that height provides a clear cut view of volume at that height as does figure 4.24 depicts.

Conclusively, of the two methods that are implemented and analysed the 3D visualization of the volume which is shown in figure 4.22 is proving to be a better approach.

The 2D view of the volume can be enabled by clicking on the Button control titled ‘2D / 3D View’ shown in figure 4.23-a. The button toggles the view between 2D and 3D views but initially the view will be a 3D view.

4.2.16 *Truncation of the Volume*

In the above section 4.2.15 it has been stated that, the volume has been truncated to the top position of the volume and middle position of the volume. For this action a Spin control has been used which is located at the bottom part of application window under the title ‘Truncate’. Figure 4.25-a depicts the control. Initially the maximum height of room will be set as the maximum height of the volume that is to de displayed. Hence all of the capture volume will be visualized initially. As user acts by clicking on up arrow and down arrow on the Spin control bar the height is increased or decreased, respectively. When the maximum height of room is reached, clicks on up arrow will have no effect and similarly when the minimum height of room is reached, clicks on down arrow will have no effect.

Prior to Spin control a Slider control has been used and it has been replaced with the Spin control. It is replaced due to the reason that Spin control offered more intuitive form of interface than the Slider control. Reason is elucidated below. Figure 4.25-b depicts the Slider control. The default position of the thumb control is set to the maximum height of the room and the range varies according to the dimensions of the room and the total height of the view volume that has been built. The minimum range of the slider control will be set to the minimum height of the room (zero) and maximum range will be set to the maximum height of the room.

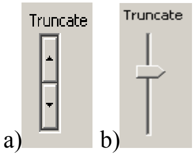


Figure 4.25: Slider control for truncating the capture volume from top to bottom horizontally. a) Using Spin Control. b) Using Slider Control.

With the slider control, user has to estimate how much sliding of the thumb has been done before actually the truncated volume is rendered in the view window. That is, the truncated volume is rendered in the view window only after the thumb button of the slider control is moved by holding left mouse button and releasing it. User should actually estimate how much of the volume will be truncated before the left mouse button is actually released from the thumb button.

The use of a Spin control proved to be a better approach as with this, it is not necessary for user to estimate how much of the volume will be truncated. The volume can be truncated in real-time by just simple clicks on up arrow or down arrow of Spin control, which varies the height of volume by one slice at a time.

This functionality has not been implemented in the application V1.0 and thus the application V1.1 is a better interface to user with much more functionalities added to it.

4.2.17 *Common User Controls*

As shown in figure 4.26 the application VICON Volume Visualization provides user with intuitive and iconic general purpose controls like file open, file save, perspective view, orthographic view, setting home position for the scene, moving to home position, resetting home position, seeing through the view volume, toggle the view of room, toggle the view of legend for colour map, start rendering and quit the application.

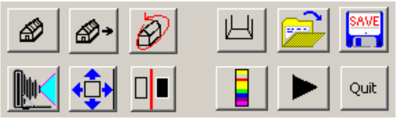


Figure 4.26: Common User Controls

Table 4.1 provides a quick reference of the controls shown in figure 4.26, individually.

4.2.18 *Data File*

The application VICON Volume Visualization provides a way, whereby data pertaining to locations of cameras and their specifications can be loaded from a file and can be saved to a file. As shown in table 4.1 the button control 📂 is for opening the data file which loads cameras p

Button Control	Function
	Move scene to home position.
	Set current position of scene as home position
	Reset home position to original position
	Toggle appearance of room in the scene
	Open a data file
SAVE	Save camera specifications and positions in a file
	Perspective view
	Orthographic view
□■	See through the view volume
	Toggle the appearance of colour map legend
▶	Render the scene
Quit	Exit application

Table 4.1: Common User Controls and their functions. Icons source Guidance Laboratory (2004).

```
20

0.0     5.0     0.0     8.5     12.0    6.5342  4.8807  0.0     45.0    45.0
2.5     5.0     0.0     8.5     12.0    6.5342  4.8807  0.0     90.0    45.0
5.0     5.0     0.0     8.5     12.0    6.5342  4.8807  0.0     135.0   45.0

0.0     5.0     2.5     8.5     12.0    6.5342  4.8807  0.0     0.0     45.0
0.0     5.0     5.0     8.5     12.0    6.5342  4.8807  0.0     315.0   45.0
5.0     5.0     2.5     8.5     12.0    6.5342  4.8807  0.0     180.0   45.0

5.0     5.0     5.0     8.5     12.0    6.5342  4.8807  0.0     225.0   45.0
2.5     5.0     5.0     8.5     12.0    6.5342  4.8807  0.0     270.0   45.0
0.0     0.0     0.0     8.5     12.0    6.5342  4.8807  0.0     45.0    -45.0

5.0     0.0     0.0     8.5     12.0    6.5342  4.8807  0.0     135.0   -45.0
5.0     0.0     5.0     8.5     12.0    6.5342  4.8807  0.0     225.0   -45.0
0.0     0.0     5.0     8.5     12.0    6.5342  4.8807  0.0     315.0   -45.0

2.5     0.0     0.0     8.5     12.0    6.5342  4.8807  0.0     90.0    -45.0
5.0     0.0     2.5     8.5     12.0    6.5342  4.8807  0.0     180.0   -45.0
2.5     0.0     5.0     8.5     12.0    6.5342  4.8807  0.0     270.0   -45.0

0.0     0.0     2.5     8.5     12.0    6.5342  4.8807  0.0     0.0     -45.0
2.5     2.5     0.0     8.5     12.0    6.5342  4.8807  0.0     90.0    0.0
5.0     2.5     2.5     8.5     12.0    6.5342  4.8807  0.0     180.0   0.0

2.5     2.5     5.0     8.5     12.0    6.5342  4.8807  0.0     270.0   0.0
0.0     2.5     2.5     8.5     12.0    6.5342  4.8807  0.0     0.0     0.0

# camera x position, y position, z position, focal length, fstop number 1/x, sensor frame width, sensor frame height, camera spin angle, horizontal tilt angle and vertical tilt angle
```

Figure 4.27: Format of data file for loading camera locations and their specifications

ositions with reference to the coordinate system described in the section 4.2.7. The data file necessarily should be in a particular format which is shown in figure 4.27. The first number indicates number of cameras that are required in the scene and each row of line below indicates data of a single camera. Order of the data is described in figure 4.27 and it is camera x position, y position, z position, focal length, f-number 1/x, sensor frame width, sensor frame height, camera spin angle, horizontal tilt angle and vertical tilt angle. The data file provides quantifiable amount of flexibility to the user. If the data file contains data of huge number of cameras, say 50, and it has been necessary to load data of only twenty of those cameras, modifying the starting number of file is sufficient which indicates number of cameras in the scene. The rest of file is ignored, loading data of only specified number of cameras.

Data shown in figure 4.27 simulates twenty uniform cameras. Scene resulted by this data has been described in the section Appendix 3.

The button control SAVE is for saving the data of cameras in a file. After loading a file or after creating an own data set, it can be altered as per user wish and it can be saved by clicking on the button SAVE and by giving a file name. Files can only be stored as normal text files with extension *.txt*. Similarly only normal text files can be used to load data into the application.

5 *Validation*

In order to publicise any type of prototypic application, it has to be declared foolproof and for any application to be declared as foolproof it has to be validated either theoretically or empirically.

The tool *VICON Volume Visualization* has been tested empirically to the maximum extent. It has been validated through the aid of VICON system located in the Hull Immersive Virtual Environment (HIVE) lab situated in the Department of Computer Science, University of Hull, Hull. The VICON system used for testing is a V8i system with 12 M2 cameras. The layout of lab with cameras' orientations is depicted in figure 5.1-a. In addition, as a reference VICON camera calibration data is obtained from the VICON system which specifies the positions of cameras, their orientations and other technical specifications of cameras like, their resolution, intrinsic and extrinsic parameters, etc. This calibration data has been used to simulate cameras in the application in order to test the accuracy of it with reference to that of actual VICON system.

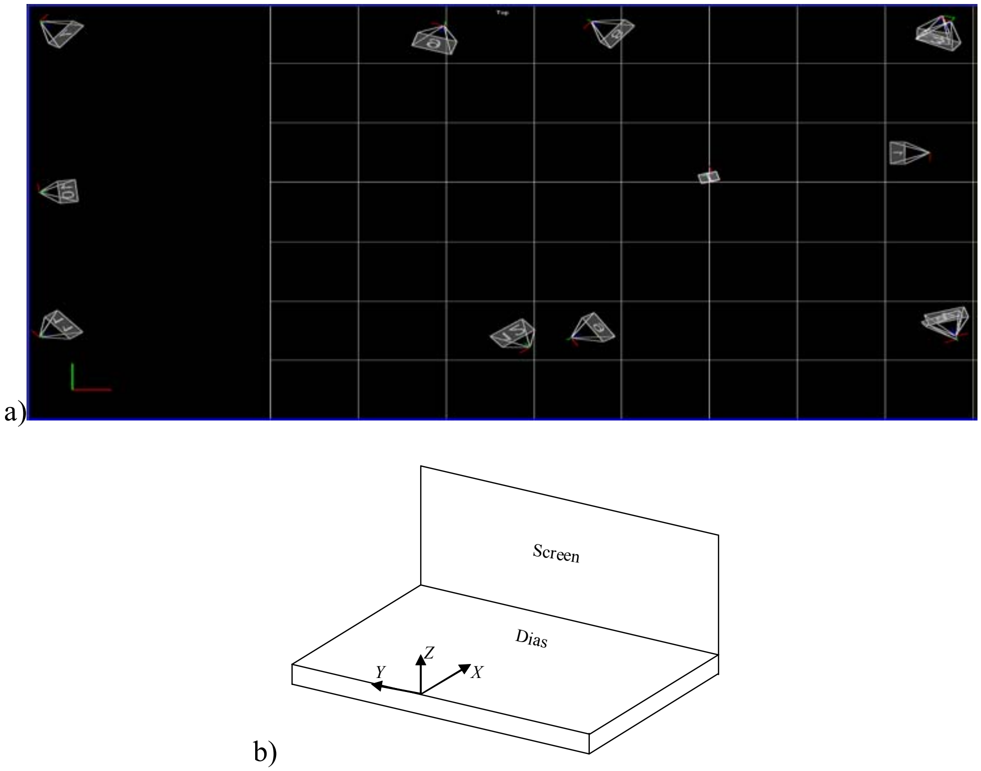
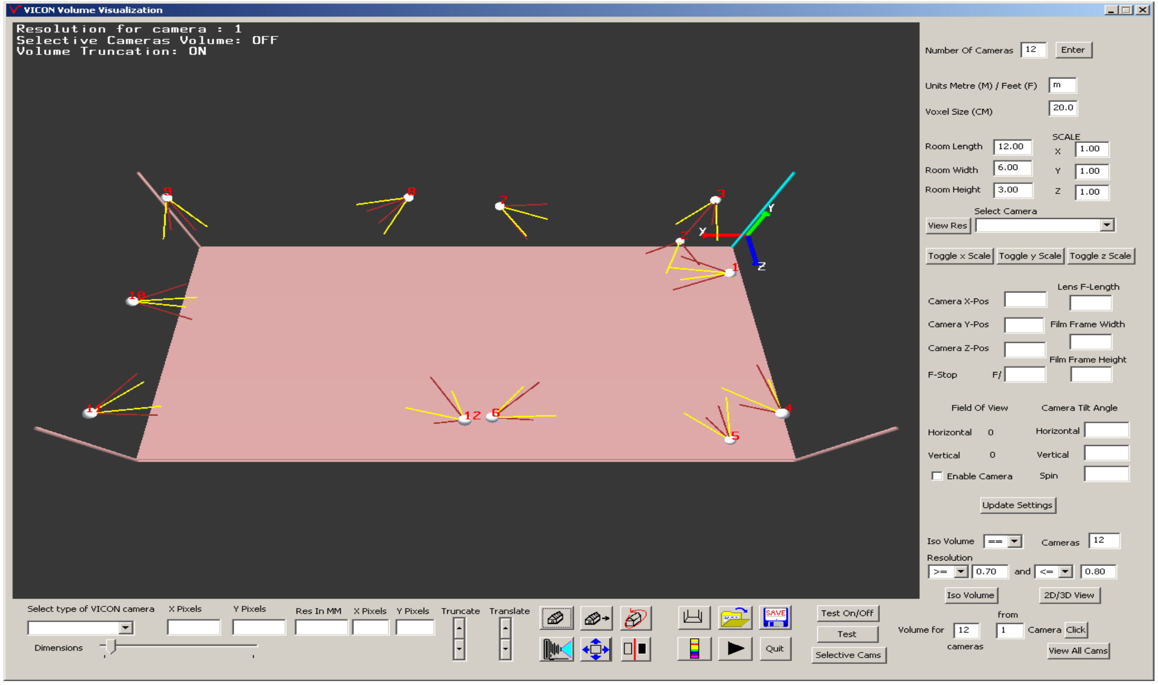


Figure 5.1: a) Screen grab from VICON IQ showing the plan of layout of cameras in the HIVE lab. The small rectangular object in the middle represents the origin of coordinate system for deducing camera positions. b) Coordinate system used for positioning the cameras in the HIVE lab. The point of origin shown here is the point indicated by rectangular object in figure-a.

Figure 5.1-c shows the simulated cameras by the application VICON volume visualization. The calibration data file that has been obtained from VICON system can be found on the CD accompanying this report bearing the file name *ViconCameraCalibration.txt*. The file presents data for twelve cameras and a block of that data pertaining to a single camera (camera #1) is shown in figure 5.1-d. The second row of the data specifies the position of cameras with reference to the point indicated by rectangular box in figure 5.1-a. The next three rows i.e. rows three, four and five present the row-wise rotation matrix that specifies the orientation of camera. Rows nine and ten specify the resolution of camera in the order X * Y.

c)



```
Resolution for camera : 1
Selective Cameras Volume: OFF
Volume Truncation: ON
```

Number Of Cameras 12

Units Metre (M) / Feet (F) m

Voxel Size (CM) 20.0

Room Length 12.00

Room Width 6.00

Room Height 3.00

SCALE X 1.00 Y 1.00 Z 1.00

Horizontal 0

Vertical 0

Iso Volume ==

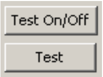
Cameras 12

Resolution >= 0.70 and <= 0.80

Volume for 12 cameras from 1

```
3               19333.9
2511.49         496.04          2597.02
0.0072789       -0.999912       0.0111298
-0.537431       0.00547427      0.84329
-0.843277       -0.0121196      -0.537343
5.73909e-07     2.03532
2.21832e-10     -4.11157e-19    0               0               0               1               7717.6          6364.8
1.25
1560
1024
0
-7859.57        8841.77         -6538.09        7202.21         0               0
```

d)



e)

Figure 5.1: c) Simulated cameras by the application. d) Calibration data of a single camera as presented by VICON system. e) Button Controls for loading VICON calibration data for the purpose of testing the application.

An additional functionality has been added to the application in order to test it. This is, to load the calibration data file directly into the application. Figure 5.1-e shows the button controls that are meant for this function. At first the room has to be set to the dimensions 12m X 6m X 3m implying length, width and height, respectively, through the edit text box controls described in the section 3.2.6. Secondly, positions of cameras in the HIVE lab shall be loaded which were pre computed and saved in the file *ActualHiveDesc.txt* which can be found on the CD accompanying this report. After loading the file, cameras at locations with awkward orientations can be seen. Improper rotations is due to the reason that Euler angles which specify camera horizontal and vertical tilt angles and camera spin angle were computed manually through the aid of orientation matrix obtained from calibration data and which are improper. Thirdly, by clicking the button Test a file open dialog box helps in loading the file *ViconCameraCalibration.txt*. By loading this file the orientation matrix (figure 5.1-d) for all cameras will be loaded which orient cameras in appropriate directions.

The button Test On/Off is used to toggle the mode of testing. When the button Test has been clicked the application switches into test mode which operates on the calibration data loaded. By clicking on the button Test On/Off the mode will be toggled and the application runs in the normal data mode. This has been provided for the reason that the application can be tested at any point of time, even in the middle of simulation of a particular data set created by user.

The dimensions of HIVE lab (figure 5.1) are 12m X 6m X 3m implying length, width and height, respectively. These dimensions were calculated based on the camera calibration data obtained from VICON system in the HIVE lab. The data specifies positions of cameras. Analysing these, the least possible dimensions that the HIVE lab could be of are 12m X 6m X 3m and in order to test it, the HIVE lab was measured manually through the aid of a measuring tape which are 11.95m X 6m X 2.9m. Both manually measured dimensions and calculated dimensions are quite appropriate and indulging.

The capture volume being visualized by the tool *VICON Volume Visualization* has been tested against the capture volume that has been simulated by VICON IQ. VICON IQ is one of the applications provided by VICON and it is used for visualizing live motion capture of reflective markers over the whole volume and at the same time helps viewing motion capture by each camera in an infant way. The paragraphs below describe the actual tests that were performed and their contending results.

5.1 *Tests and their results*

Testing procedure

At first a wand, shown in the figure 5.2 below, is placed at a trackable point in the room. The VICON IQ is set to visualize capture volume of each camera individually. A screen shot of this mode of operation by VICON IQ is shown in figure 5.3. Through this, visually, the number of cameras that can track the Highlighted (black coloured) Reflective Marker (HRM) on the wand is found out. The position of HRM has been obtained through the use of a simple command-line based program called “Tracker”. Tracker is a simple program that takes the input of identifier of particular motion capture system, in this case it is ‘vicon’, and a port address to connect to the system on which the motion capture system is running and using the system data computes the location of a specified point, in this case it is ‘HRM’.

As a second step, the tool VICON Volume Visualization is simulated for showing the capture volume of *n* number of cameras. Where, *n* is the number of cameras tracking the wand. The coordinates of highlighted reflective marker obtained through the program “tracker”, are checked in the VICON Volume Visualization tool in 2D and in 3D (section 3.2.15). A check is performed to find out whether the point is falling with in the simulated capture volume by truncating the volume and by viewing it in 2D and 3D. The paragraphs that follow describe actual series of tests performed.

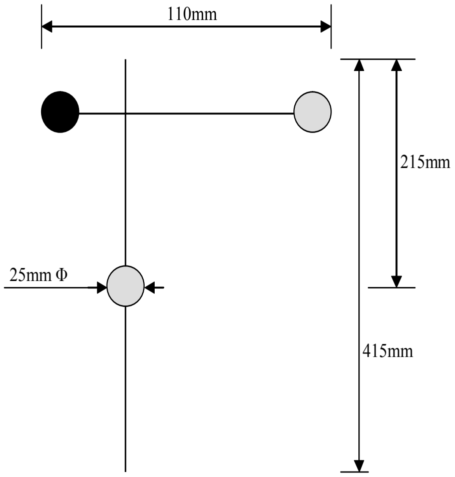


Figure 5.2: Schematic diagram of wand with reflective markers used for testing. The marker highlighted is the marker that has been considered for obtaining the coordinates of point being tracked.

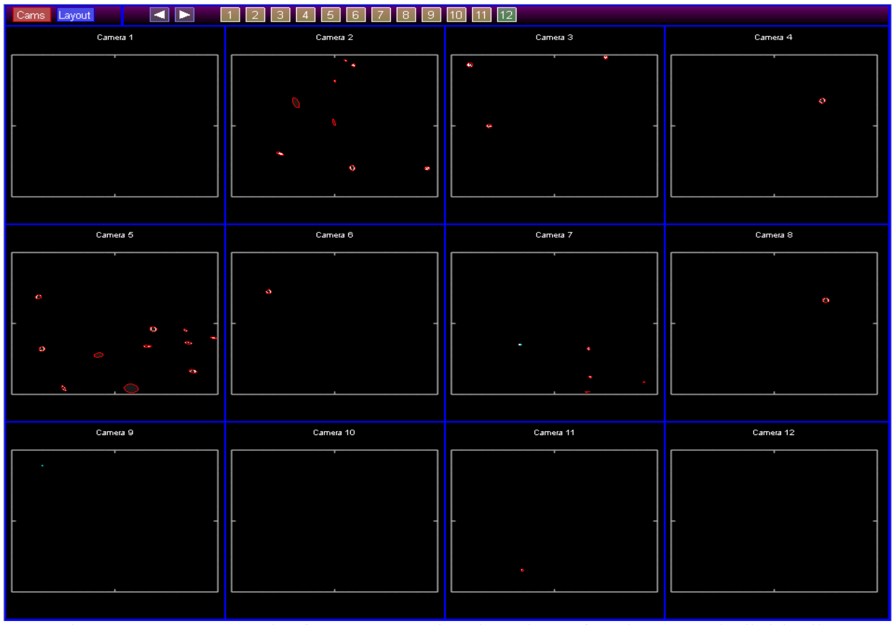


Figure 5.3: Screen grab of VICON IQ showing view of each camera individually.

Test #1

The HRM[12] is placed at a point in the lab with coordinates

(X, Y, Z) = (-0.00915, 0.00653, -0.00763) ~ (0.0, 0.0, 0.0) – Origin.

Through the aid of VICON IQ, the number of cameras that are tracking the HRM are known visually. When the HRM is placed at this point the number of cameras that were able to track it

[12] As indicated in the paragraph number 1 on page 48, HRM stands for Highlighted Reflective Marker. The reflective marker coloured in dark black in figure 5.2.

were 6. The screen grab of the VICON IQ is shown in figure 5.4 which depicts the number of cameras able to track HRM positioned at origin.

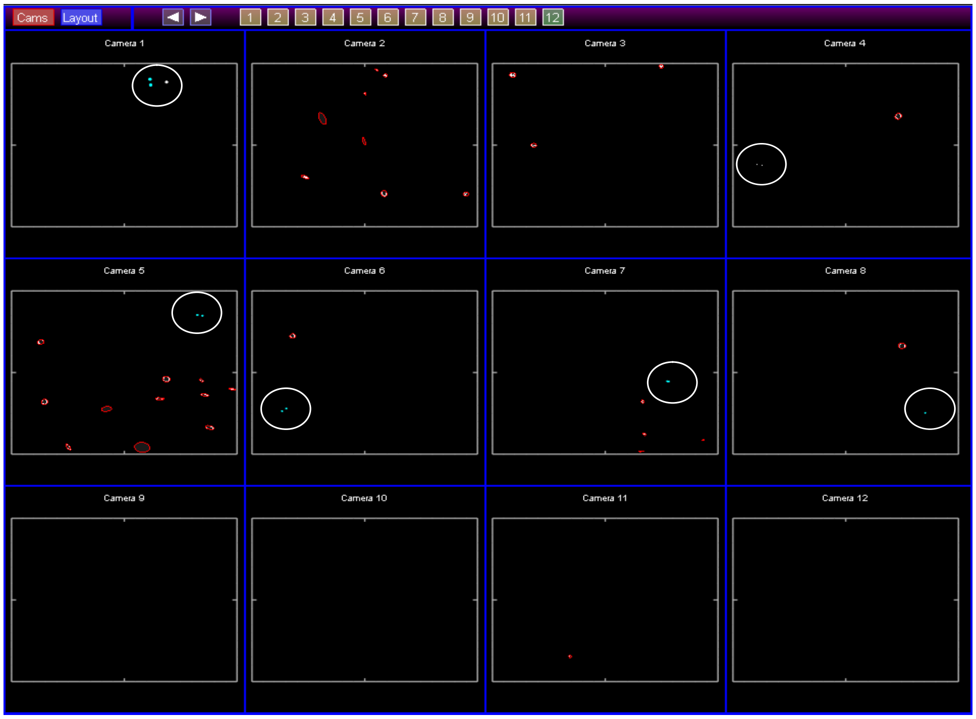


Figure 5.4: Screen grab of VICON IQ showing number of VICON cameras tracking the HRM. The HRM tracked by cameras is sacked with a white circle for identification.

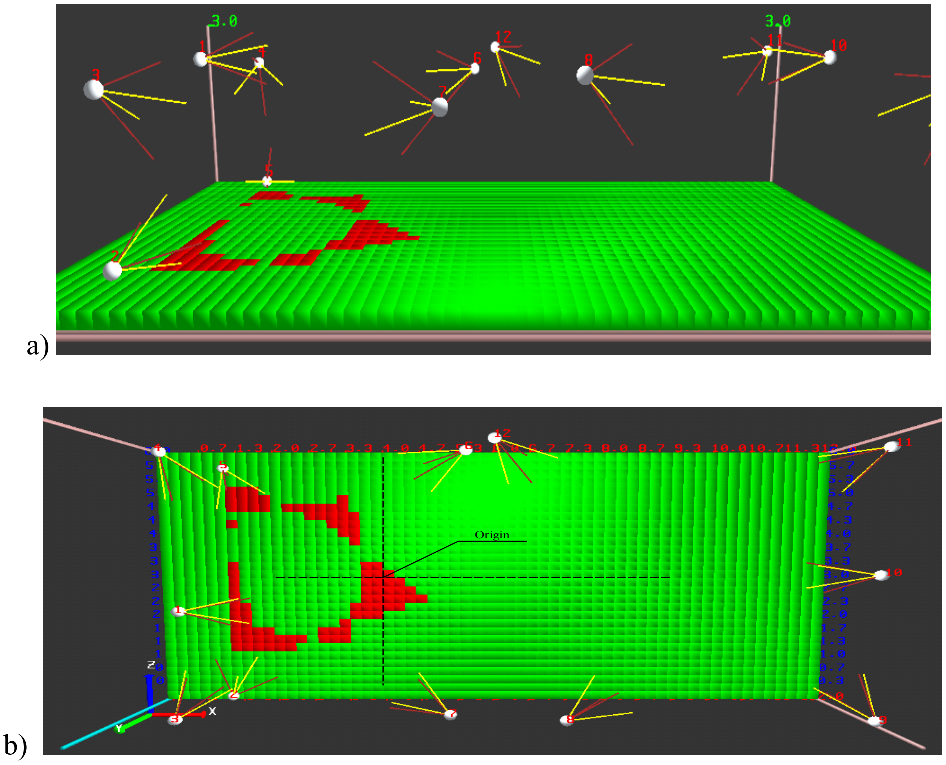


Figure 5.5: a) 2D view of the volume truncated to view origin. b) Origin indicated through dimensions and showing that six cameras are tracking the origin.

The point where HRM was positioned has been measured manually with measuring tape and the measurements were (3.9, 2.9) which indicate length and width, respectively. As the HRM has been positioned on the ground the height is almost zero. Hence the view volume resulted has been truncated to the lowest level in the room. Figure 5.5-a, shows the truncation level and depicts the last slice of the view volume as the height is zero or negligible.

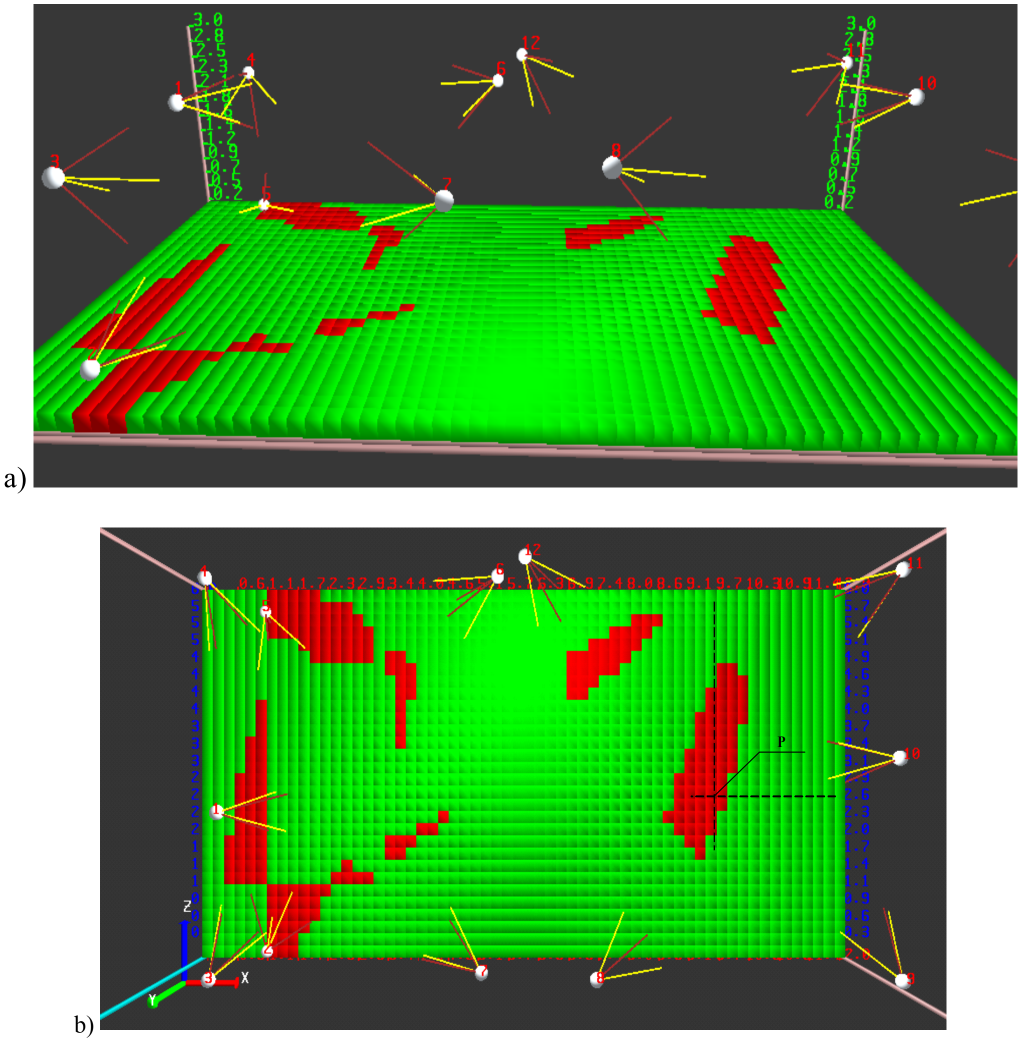


Figure 5.6: a) volume captured by five cameras at ground level. b) The exact point **P** indicated.

The application VICON volume Visualization tool is simulated to visualize the capture volume of 6 cameras and the figure 5.5-b shows the volume, elucidating that the origin has been visualized by the application under the simulation for *exactly* six cameras.

With reference to figure 5.5-b it can be deduced that the application VICON Volume Visualization tool is simulating in a right way.

Test #2

HRM is placed at a point with coordinates (X, Y, Z) = (-5.76, 0.361, 0.226) with respect to the coordinate system shown in figure 5.1-b. Through VICON IQ total number of cameras that were able to track the HRM was known and it is 5. The location of point in application's coordinate system is 9.7, 2.6, and 0.2 implying length, width and height, respectively. The application VICON Volume Visualization is simulated for showing the capture volume of five cameras. The truncated volume simulated by the application is as shown in figure 5.6-a.

The point **P** at which HRM is positioned is indicated in figure 5.6-b. The dotted lines refer to the length and width of that point. Figure 5.6-a indicates the height of the sliced volume, that is, 0.2m or 20cm. Only one slice is displayed which is equal to 20cm, the size of voxel.

With reference to figure 5.6-b it is obvious that the volume simulated by the application and that shown by VICON IQ are akin.

Test #3

HRM is placed at point (X, Y, Z) = (-2.34, -2.78, 1.74) with respect to the coordinate system shown in figure 5.1-b. This point in the terms of application's coordinate system has the measurements 6.34m X 1.74m X 5.38m implying length, width and height, respectively. The maximum number of cameras that can track this point are two as simulated by the application VICON Volume Visualization but VICON IQ can show that this point is being tracked by only one camera. The reason has been found out. It is the sensitivity of camera that is causing this error along with high intensity ambient and diffuse light present in the room. The application VICON Volume Visualization does not take into account the sensitivity of the VICON cameras and surrounding light intensities. Sensitivity of camera can be related to its Depth of field. It can be defined as the clarity with which a camera can capture before and beyond its point of focus (WorldHistory.com, 2004b). These are discussed in the chapter 6.

The point being tracked as shown by VICON IQ is depicted in figure 5.7. It shows that camera 7 (refer to figure 5.1-a) is tracking the point. At the same time the point is lying with in the FOV of camera 9 but the camera is not able to sense the reflective marker because of its sensitivity and extraneous light intensity present in the room. To test this case, the wand has been moved slightly inwards of the room to a length of 10cm, and the HRM is shown as being tracked by camera 9. The screen grab is shown in figure 5.8. Even when the point is positioned at its original position, it is being tracked by camera 9 but it is spurious. This has been concluded by observing spurious blinks in the FOV of camera 9 and the point is unsuitable for archive.

The volume simulated by the application VICON Volume Visualization is shown in figure 5.9 with the point of interest indicated as **Q**. Figure 5.9-a shows the length of point, 5.9-b shows width of point and 5.9-c shows height of point.

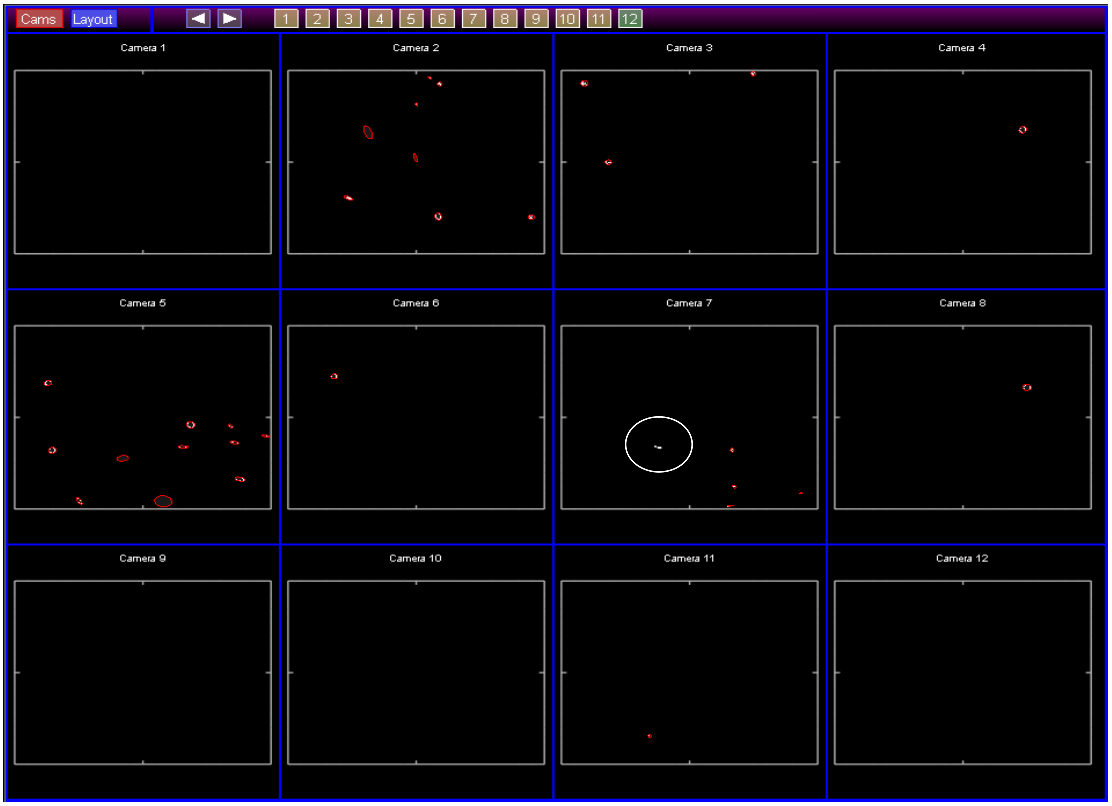


Figure 5.7: Screen grab of VICON IQ showing the point that is being tracked by 7th camera.

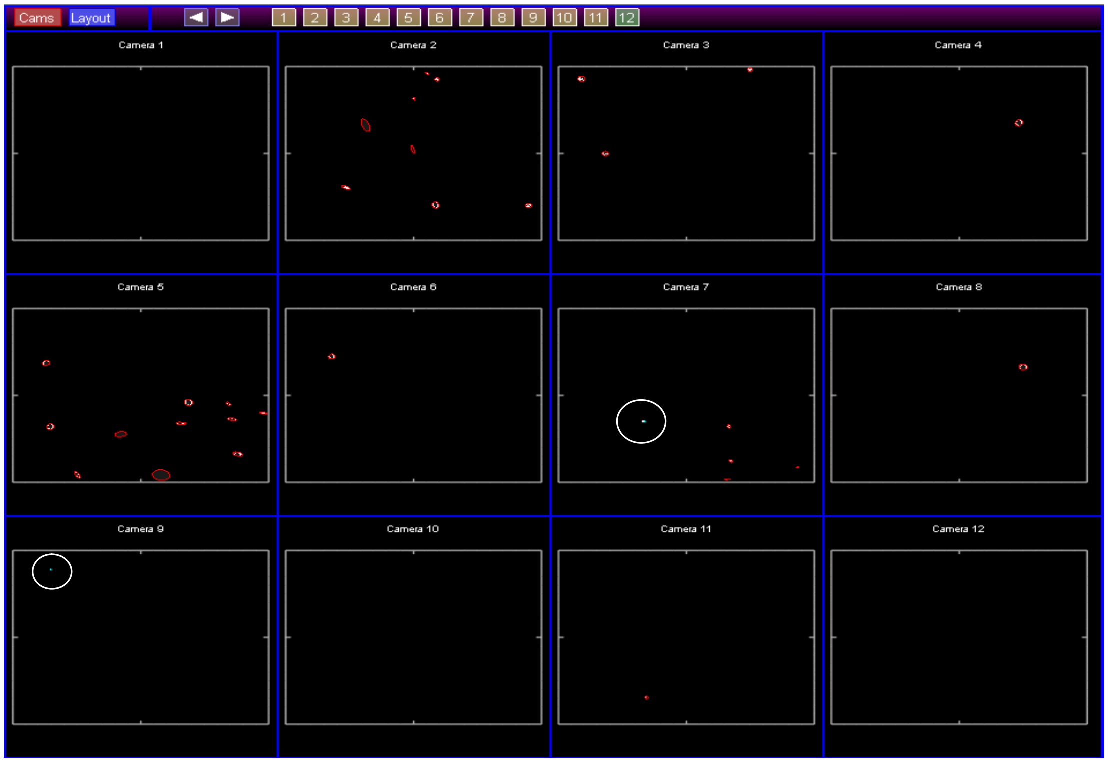


Figure 5.8: screen grab of VICON IQ showing the point that is being tracked by cameras 7 and 9.

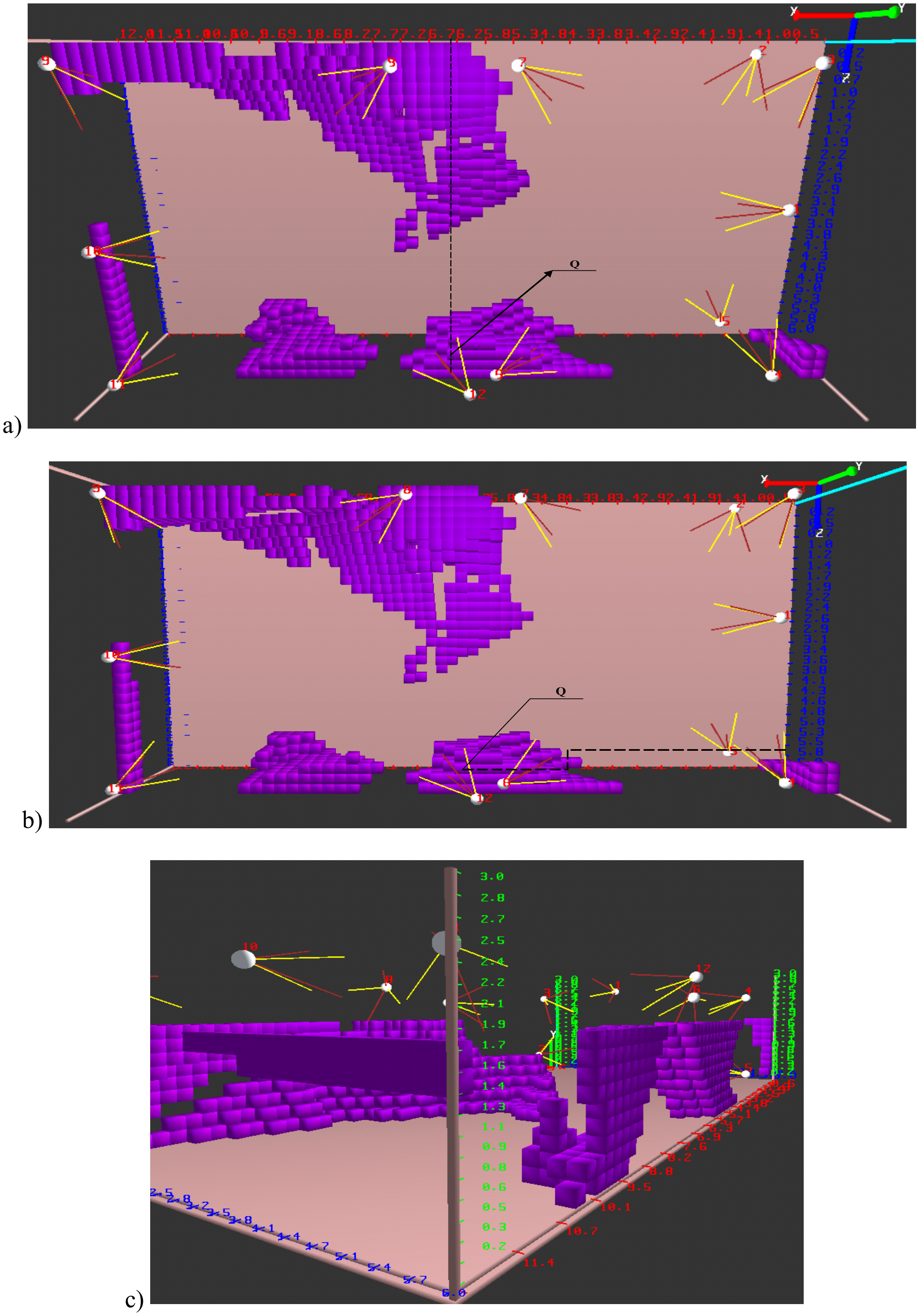


Figure 5.9: a) Point **Q** indicated at 6.3m length. b) Point **Q** indicated at 5.7m width. c) Height of the volume set to 1.7m.

With this it can be understood that the application VICON Volume Visualization is simulating appropriate capture volume.

Test #4
HRM is positioned at a point (X, Y, Z) = (0.953, 0.00947, -0.00466) which is near to the origin of VICON system and is shown in figure 5.1-b. Measurements of the point with respect to the application's coordinate system are 3.15m X 3.99m X 0.0m implying length, width and height, respectively. The number of cameras that can see this point are 7 and it is known through the aid of VICON IQ. Figure 5.10 shows the screen grab depicting that 6 cameras are tracking HRM positioned at this point. As in the case of *Test #3*, tracking by camera 11 (shown in figure 5.1) is spurious. The reasons are the distance the point is away from camera 11, its sensitivity and high intensity of light with which the environment of lab is bathed with. The screen showing very tiny blinking dots which indicates that HRM is being tracked with camera 11 has not been archived as it was unable to grab.

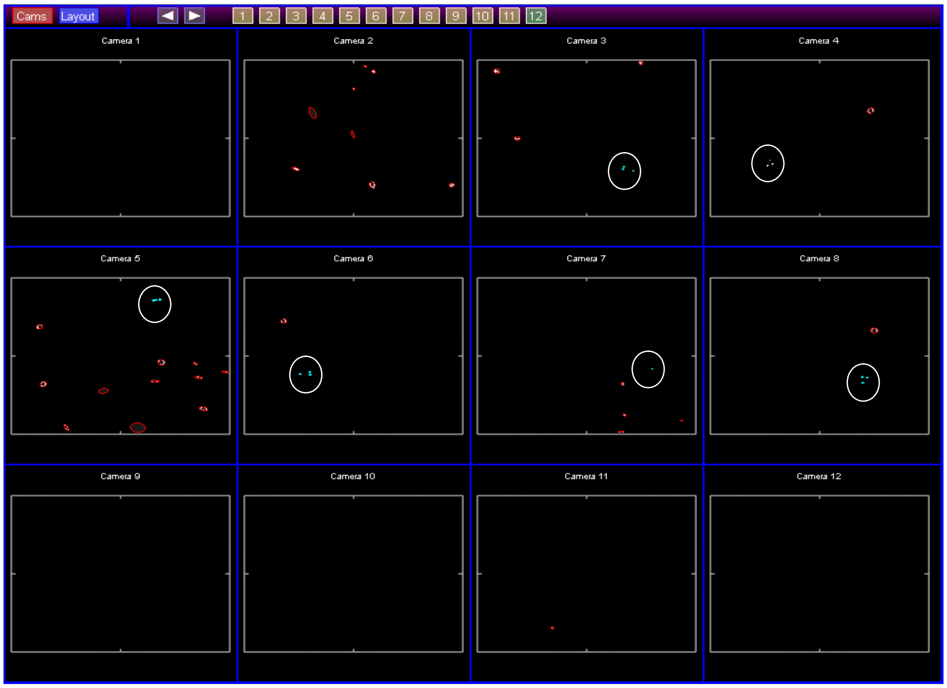


Figure 5.10: Screen grab of VICON IQ depicting that six cameras are able to track HRM. Tracked points are circled for the sake of identification.

The application VICON Volume Visualization is simulated to visualize the capture volume of seven cameras and it is shown in figure 5.11 and the point where HRM is placed is indicated with the letter **R**. Dotted lines indicate an approximate measure of the measurements of the point in the application's coordinate system which were measured manually.

With reference to this test it is understood that the application VICON Volume Visualization is simulating appropriate capture volume.

Test #5
Tests one through four described above are performed by placing wand at a fixed position in the lab. In addition to these, a series of tests were performed to check the volume roughly. That is capture volume for certain number of cameras has been simulated and through the aid of VICON IQ and the volume shown by the application VICON Volume Visualization is tested by waving the wand roughly through out the volume.

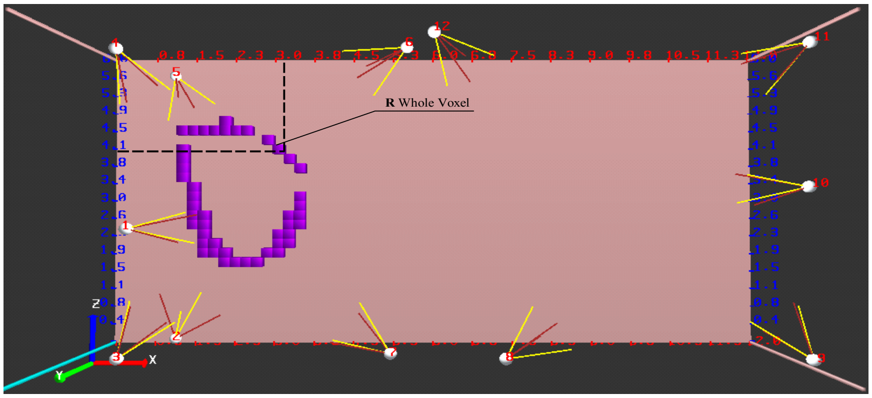


Figure 5.11: Position of HRM is indicated by **R**. Volume for seven cameras.

The test has been performed for capture volumes simulated for eight, three and two cameras. Very few of the intricacies which govern the motion capture by VICON cameras were considered while simulating the capture volume by the application VICON Volume Visualization. Inexplicable features like other light sources present in the room, sensitivity of the camera were not considered wile simulating the capture volume. Hence an approximate measure of the volume has been considered while testing.

Roughly the wand has been waved through out the volume that the application is showing and checked whether the VICON IQ is showing that the wand is being tracked by the similar number of cameras. The results were indulging.

Capture Volume roughly tested for two cameras and three cameras were quite indulging. The capture volume tested for eight cameras was also quite indulging but it is necessary to describe the spurious faults that occurred during this test.

While testing roughly over the volume simulated for eight cameras, it was observed that VICON IQ is visualizing that the wand is being tracked perfectly by seven cameras while a spurious and intermittent tracking by eighth (camera #11) camera is noticed. The camera is *camera 11* (figure 5.1) which is located at the back end of the room. The reasons, as described in the *Test #4* and *Test #3*, are sensitivity of VICON cameras and extra light sources that are present in the room. These intricacies were not considered while capture volume is being simulated. Ideal conditions were laid down while simulating the capture volume. In the chapter 6 these intricacies were discussed which can be implemented as added functionalities to the application VICON Volume Visualization.

Section 3.1 describes a simple test done between the simulated FOV values and actual values proving that they are accurate.

6 *Future Work*

There can always be a better approach for accomplishing a task and it can always be improved. One approach can be proved better than the other, but none can be stated as the best approach. Like wise the tool VICON Volume Visualization can always be made better in terms of its *user interface* and added functionality.

Some approaches that would grant better added functionalities are outlined in this chapter.

The user interface implemented in the application V1.1 provides a better way to operate with the tool when compared with that of application V1.0. But as said above, a better approach can always be implemented replacing the existing approach. In this way some alterations can be made to the interface for effective interaction. The paragraphs below indicate the alterations that would prove to be a better approach over the existing approaches.

The VICON Volume Visualization tool did not consider any features like sensitivity of the camera, surrounding light intensities which affect the motion capture of cameras.

6.1 *Affect of surrounding light intensities*

As outlined in the table 6.1 the surrounding ambient and diffuse light in the room affects the motion capture of reflective markers by VICON cameras.

For example, if the environment that is being tracked is filled with a light of similar hue as that of the hue of strobe lights surrounding the lens of VICON cameras, the tracking of retro-reflective markers and their resolution (number of scan lines per marker) might be affected. The VICON Volume Visualization tool does not take into account the surrounding environmental lighting and all the simulations implemented by the application are at ideal conditions. But, ideal conditions rarely exist and hence it would be a better added functionality to the VICON volume visualization tool to simulate the commonly existing environmental conditions and to build an interface for customizing various light settings.

The data provided by VICON which elucidates the effect of surrounding ambient light is as below.

	Filter	No filter
500 Watt Light Off	Very clear markers	Slightly enlarged markers
500 Watt Light On	Markers merge	White saturation

Table 6.1: Ambient light tolerance with and without dye filter. Data Courtesy of VICON Motion Systems Ltd.

Table 6.1 shows the effect of ambient light on the tracking of reflective markers with and without filters that help pass light with the hue as that of strobe light's hue into the camera. With filter in action and with out any ambient light the markers can be seen clearly. With filter in action and with ambient light of luminous intensity 500watt the tracking of markers is merged with surrounding ambient light. With no filter in action and with out any external light disturbances the markers are slightly enlarged. The reason for slightly enlarged markers is unobvious. With 500Watt light on and with out filter the scene is saturated with white light.

In common, the VICON cameras are used in an environment consisting of lower intensity incandescent lamps with adjustable luminous intensity, so that, intensity of the lamps can be decreased while performing motion capture. So, the future work includes consideration of surrounding light intensity while simulating the VICON cameras and adding an interface through which lights with various hues and intensities can be introduced into the environment and simulated with those conditions in effect.

This will help user analyse the effect of various environmental settings before the actual room for motion tracking has been made ready and cameras have been fixed. With the help of this added functionality light intensity needed in the room and which doesn't affect the motion capture of VICON cameras can be predicted before hand.

6.2 *Analysis of scan lines for reflective markers*

The application VICON Volume Visualization helps analyse the capture volume in terms of its resolution. That is, the application visualizes the maximum and minimum values of the resolution of a particular camera in generic terms, as explained in the section 3.4. A generic camera has been considered here and the resulting resolution in millimetres has been visualized to user in the equivalent pixels through the VICON Cam Type interface as described in the section 4.2.13. But in practical, VICON cameras analyse a particular reflective marker in the terms of number of scan lines per inch. These scan lines indicate signals to VICON system which imply that a reflective marker is being tracked by a camera. For this reason even if a naked human hand is waved close to the camera, the hand will be tracked. This has been observed while performing the tests that were described in chapter 5. Hence, a better way of analysing the capture volume would be in terms of scan lines of reflective marker at a particular distance when tracked by a particular camera in the room. The number of scan lines also depends upon the external light intensities that were present during motion capture. Hence the functionality described in the section 6.1 along with analysis of capture volume in terms of number of scan lines at a point per a marker of certain size would prove to be the better ones.

Each VICON camera has its own scan frequency which implies number of scan lines per inch (VICON, "nd"e) of the volume captured by camera. Each scan line is again sampled with a particular sampling frequency (VICON, "nd"e) which results in the number of pixels (VICON, "nd"e). The video rate of the camera defines sampling frequency (Tebbutt, Wood, King 2, 2002).

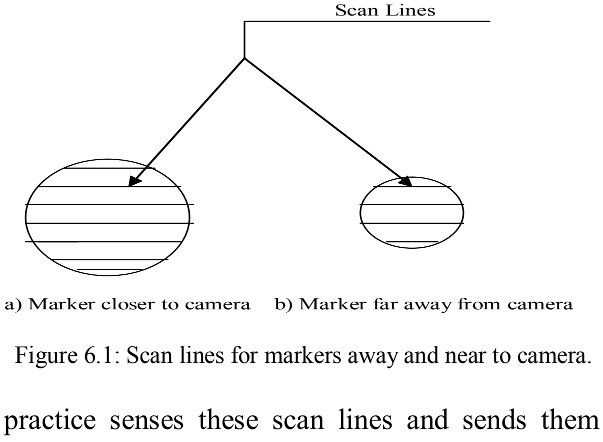
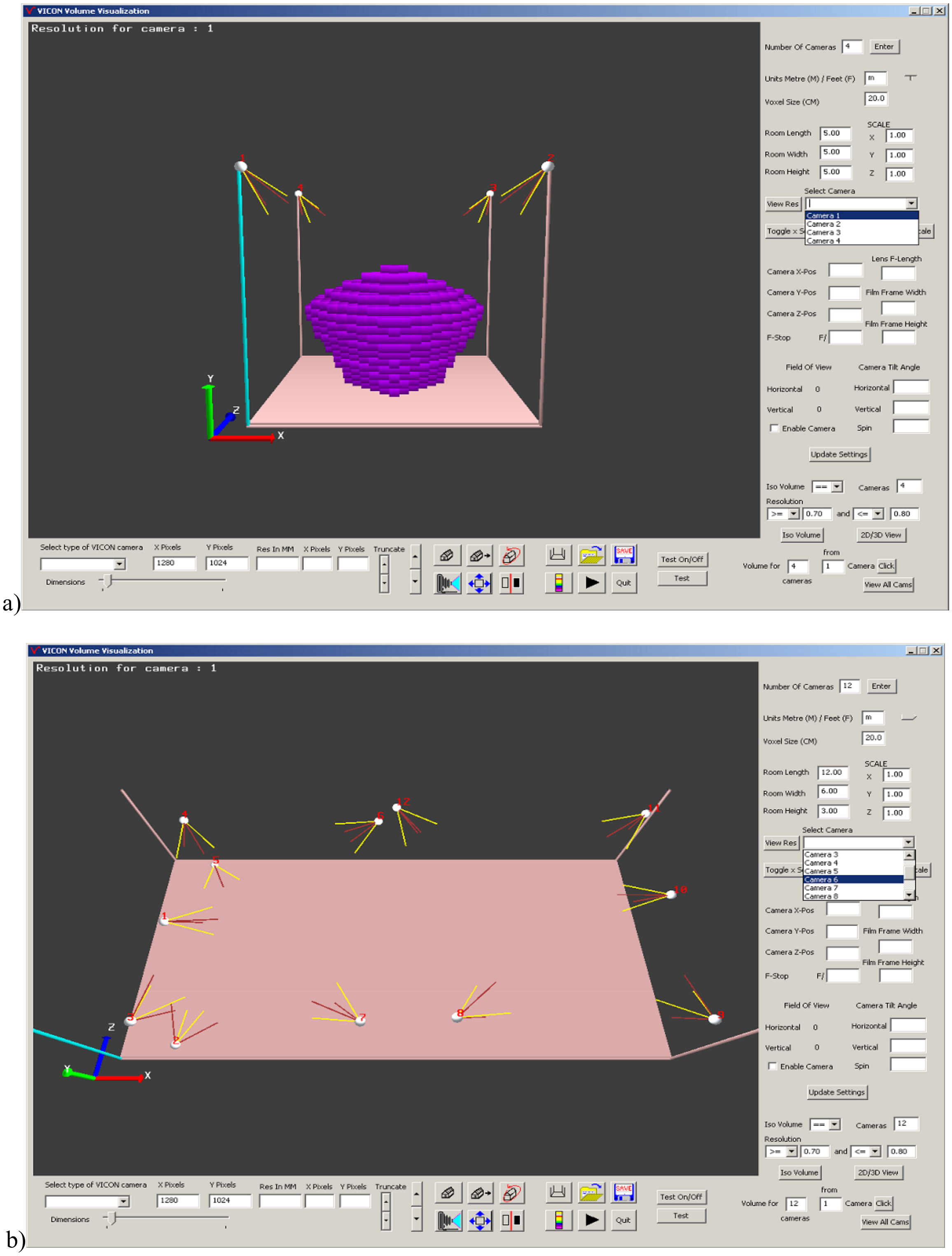


Figure 6.1: Scan lines for markers away and near to camera.

A VICON camera in practice senses these scan lines and sends them to the Datastation for analysis and Datastation then senses the total number of scan lines by sensing the start and stop signals of each scan line. The total number of scan lines governs the sensitivity of a reflective marker which varies with the distance that a marker is away, from a camera.


a)

VICON Volume Visualization

Resolution for camera : 1

Number Of Cameras 4

Units Metre (M) / Feet (F) m

Voxel Size (CM) 20.0

Room Length 5.00, Room Width 5.00, Room Height 5.00

SCALE X 1.00, Y 1.00, Z 1.00

Field Of View: Horizontal 0, Vertical 0

Iso Volume ==, Cameras 4

Resolution >= 0.70 and <= 0.80

X Pixels 1280, Y Pixels 1024

Volume for 4 cameras from 1 Camera

b)

VICON Volume Visualization

Resolution for camera : 1

Number Of Cameras 12

Units Metre (M) / Feet (F) m

Voxel Size (CM) 20.0

Room Length 12.00, Room Width 6.00, Room Height 3.00

SCALE X 1.00, Y 1.00, Z 1.00

Field Of View: Horizontal 0, Vertical 0

Iso Volume ==, Cameras 12

Resolution >= 0.70 and <= 0.80

X Pixels 1280, Y Pixels 1024

Volume for 12 cameras from 1 Camera

Figure 6.2: a) A room with four cameras. b) A room with 12 cameras.

Hence, analysing the capture volume in terms of number of scan lines would add up to the already existing functionality to analyse the resolution of capture volume in millimetres thereby providing a better insight of the capture volume visualized.

6.3 *Intuitive selection of cameras*

As described in the section 4.2.10 the application VICON Volume Visualization provides a better interface for selecting a camera and manipulating its parameters. An interface that proves to be better than this interface has been identified and is outlined in this section.

Instead of selecting a camera from the list of cameras presented in the combo box, a camera can be selected directly by clicking with mouse on the camera in the view window. Figure 6.2-a depicts four cameras and 6.2-b depicts 12 cameras. When there are only four cameras selecting a particular camera from the list provided in the combo box is not a tedious task but as the number of cameras increases, selecting them from the list would be tedious, as shown in figure 6.2-b. Hence, selecting a camera by simply clicking on the camera in the view window would be a better way.

The approach
In order to implement this type of interface, Viewport coordinates are to be transformed to window coordinates by using a form of viewport transformations.

When user clicks on the view window a message[13] will be sent to the application by the Windows operating system for it to be handled by the application and perform required action. For example, when user clicks left mouse of button by placing cursor on camera (sphere) numbered seven, which is shown in figure 6.2, a message called *WM_LBUTTON_DOWN* will be sent by operating system to the application. *WM_LBUTTON_DOWN* implies Windows Message Left Button Down. A call back function[14] that is logically attached to this message receives (X, Y) coordinates of the location of cursor on the view window when the button is originally clicked. These coordinates are called Viewport Coordinates and the real-world coordinates (X, Y, Z) which represent the actual location of camera are called as Window Coordinates (Scott, 1995). The formulae given below shall be used to convert Viewport coordinates to window coordinates.

$$Xc = left + (right * ((2.0 * X) / windowXSize))$$
$$Yc = bottom + (top * ((2.0 * Y) / windowYSize))$$

As shown in figure 6.3, (Xc, Yc) are the calculated world coordinates, (X, Y) are viewport coordinates (coordinates of point on screen) which are sent to the call back function that is meant for handling mouse button click event, (windowXSize, windowYSize) indicate the total number of horizontal and vertical pixels on the screen, respectively and (left, right, top, bottom) indicate near plane boundaries of the view frustum. The depth is ignored as eventually all coordinates are mapped on to screen where there is no information regarding depth. Finally the point in real world coordinates be (Xc, Yc, Zc), where Zc shall be equal to the value of near plane.

[13] Windows operating system issues messages to the application when ever an event such as moue click, key hit, etc., occurs as a result of user action. These messages are to be handled by the application for it to take necessary actions. A call back function is attached logically to an event and when ever an event occurs that function will be called by the application.

[14] A call back function is just like all other functions but they will be called by the application when ever an event occurs.

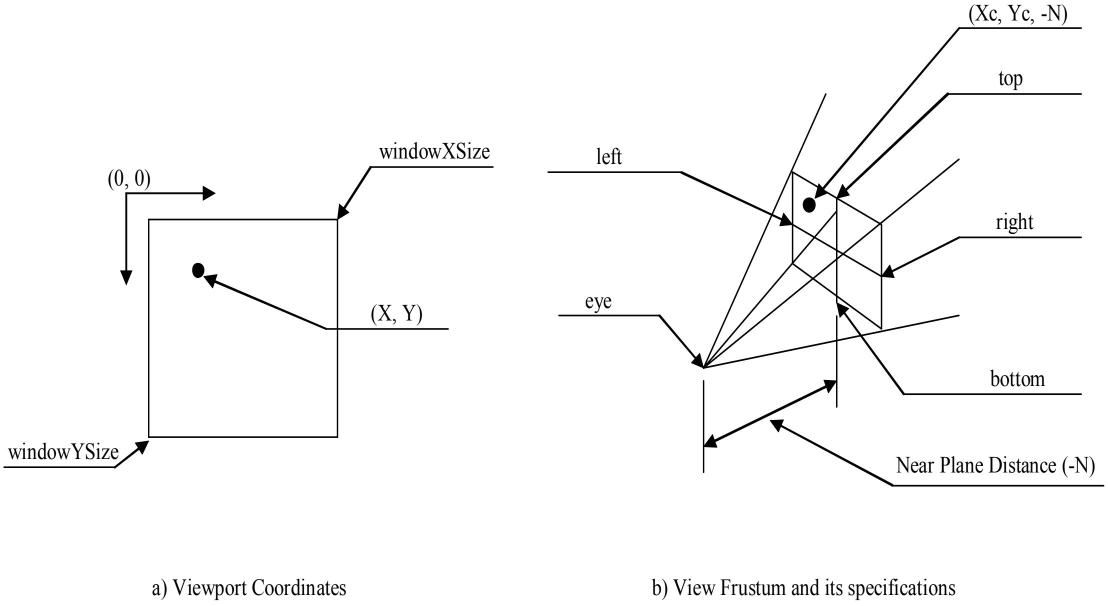


Figure 6.3: Viewport Window coordinates and specifications

Selection of Camera

In the application VICON Volume Visualization a camera is represented by a point which indicates location of camera in world coordinates and four vectors which are used to represent FOV of camera as described in the section 3.2. Let the real world coordinates at which camera is located is (Xp, Yp, Zp).

A simple test between the coordinates (Xp, Yp) and (Xc, Yc) reveals whether the point at which user clicked is on the camera or not. The test can be done as described below. In the application VICON Volume Visualization a camera is represented as a sphere. Practically, a sphere appears as a circle on the display screen. So let us consider a bounding square around the camera as shown in figure 6.4. As shown in figure 6.4 coordinates for four corners of bounding square can be calculated.

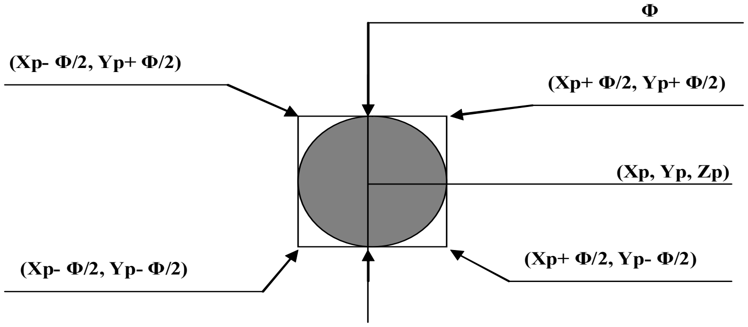


Figure 6.4: Coordinates of bounding square for a circle with diameter **Φ**

A simple test between (Xc, Yc) and the four corners of square reveals whether the point (Xc, Yc) falls with in the square or not. If the (Xc, Yc) lies with in square then the camera with coordinates (Xp, Yp, Zp) will be selected.

This method is rather easy to implement when there is a single viewport. When there are multiple viewports, as it is in the application VICON Volume Visualization, viewport coordinates should be exactly mapped to respective viewport's world coordinates which is rather complicated. Further information regarding transformations from viewport coordinates to window coordinates can be obtained from Hill (1990) and Scott (1995).

7 Conclusions

The application VICON Volume Visualization tool offers VICON Motion Systems a better approach for setting up the VICON system at its client areas. Knowing the dimensions of room, the application provides an easy way to know the capture volume of cameras placed at certain positions. Various parameters pertaining to the cameras can be manipulated and their effect can be realised in real-time. Thus, this application provides an easy way for a client to decide exact specifications of cameras that best suits their needs and their locations in the room.

This application provides a robust way of setting up VICON system for an engineer. All of the basic functionalities that a technician shall need to engineer during setup of the system have been provided in this application.

The application VICON Volume Visualization tool has been made foolproof and solicits no fiscal resources from the firm VICON Motion Systems Ltd., for it to incorporate into its package and utilize it for a better approach during setup of the system at its client areas.

All of the primary and secondary objectives that were laid out, and which were described in the section 1.2, were achieved. However, any application can ever be improved and made better to its previous forms. In fact, it can never be declared that, one particular form is the best. It can only be said that, "one is better than the other and hence by implementing that procedure the application has been improved over its previous forms."

And, hence, the tool VICON Volume Visualization can also be improved. An important improvement has been formulated which can help simulate the capture volume of cameras in a more effective and practical way. It is,

- Affect of surrounding light (ambient, diffuse or specular) on the motion capture of reflective markers.

This has been discussed in chapter 6.

The application provides a better and easy to use interface for manipulating various parameters pertaining to cameras which govern the view volume and also provides a clean interface to manipulate various parameters pertaining to the application.

Appendix 1

Finding angle between two vectors

Vector algebra provides a way through which angle between two vectors or angle between two intersecting lines can be calculated. This can be achieved through Dot Product of two vectors. Dot Product is used to solve a number of important geometric problems in graphics (Hill, 1990).

Definition: Dot Product: The dot product d of two n-dimensional vectors $\mathbf{v} = (v_1, v_2, v_3, \ldots, v_n)$ and $\mathbf{w} = (w_1, w_2, w_3, \ldots, w_n)$ is denoted as $\mathbf{v.w}$ and has the value $d = \mathbf{v.w} = \sum v_i w_i$ for i = 1 to n. (Hill, 1990)

Figure A1.1 below shows the 2D case, where the vectors B and C lie at angles β and α, respectively, relative to the x-axis. From elementary trigonometry,

$$B = (|B| \cos \beta, |B| \sin \alpha)$$

and

$$C = (|C| \cos \beta, |C| \sin \alpha).$$

The Dot Product of B and C is

$$\begin{aligned} B . C &= |B||C| \cos \alpha \cos \beta + |B||C| \sin \beta \sin \alpha \\ &= |B||C| \cos (\alpha - \beta). \end{aligned}$$

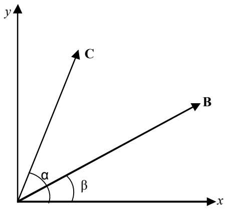


Figure A1.1: Finding the angle between two vectors. Source: Hill, 1990.

Thus for any two vectors B and C,

$$B . C = |B||C| \cos \theta, \qquad \text{(A1.1)}$$

where θ is the angle from B to C.

The equation A1.1 can be represented in a more compact form. By diving both sides of equation A1.1 by |B||C| and by using Unit Vector notation $B^{\wedge} = B / |B|$ (Hill, 1990) the compact form of equation A1.1 which is given as equation A1.2 below can be obtained.

$$\cos \theta = B^{\wedge} . C^{\wedge}. \qquad \text{(A1.2)}$$

The equation A1.2 fits for vectors of three, four, or any number of dimensions. (Hill, 1990)

Appendix 2

Rotation of vectors using matrices

Affine Transformations, Affine Combination of Vectors and Homogenous Coordinates

Affine transformations are a fundamental cornerstone of novel graphics algorithms. They provide a compact way to keep straight distinction between points and vectors. Principally, points and vectors do not transform the same way. It is necessary to keep the distinction straight, which is offered by using coordinate frames and appropriate homogenous coordinates (Hill, 1990).

Coordinate Frame consists of a particular point ΰ called the origin and some mutually perpendicular vectors i, j and k, the 3D case, that serve as the axes of the coordinate frame. In the application VICON Volume Visualization the coordinate frame consists of,

ΰ = (0, 0, 0, 0),
i = (1, 0, 0, 0),
j = (0, 1, 0, 0) and
k = (0, 0, 1, 0).
in homogenous coordinates. (Hill, 1990)

Affine combination of vectors simply means, that a combination of vectors A and B whose coefficients add up to unity. For example, 3A + 2B – 4C is an affine combination of vectors. Simply speaking normalised vectors are affine form of vectors. (Hill, 1990)

The use of *homogenous coordinates* is one of the hallmarks of computer graphics, as it both helps to keep straight the distinction between points and vectors and provides a compact notation for working with affine transformations. In addition, matrix manipulations in homogenous coordinates provide an easy way to distinguish between a point and a vector (Hill, 1990).

To go from ordinary to homogenous coordinates,

If the object is a point, append a 1.
If the object is a vector, append a 0. (Hill, 1990)

To go from homogenous coordinates to ordinary coordinates,

If the object is a vector delete the final coordinate 0.
If the object is a point, delete the final coordinate 1. (Hill, 1990)

All matrix operations that were performed in the application VICON Volume Visualization are 4D and all of the vectors are represented in homogenous coordinates.

Rotations

Rotations in three dimensions are common in graphics, for we often want to rotate an object or a camera in order to obtain different views. There is a much greater variety of rotations in three dimensions than in two dimensions, since we must specify an axis about which the rotation occurs, rather than just a single point. Matrices provide very compact and easy method for performing geometric transformations for objects, points and vectors.

Rotation about the *x*-axis is called an "*x*-roll", rotation about *y*-axis is called as "*y*-roll" and rotation about *z*-axis is called as "*z*-roll". In each case the rotation is through an angle β, about the given axis. Individual matrices for producing *x*-roll, *y*-roll and *z*-roll can be represented. Positive angles are defined using "looking inward" convention. (Hill, 1990)

> *"Positive values of β cause a CCW rotation about an axis as one looks inward from a point on the positive axis toward the origin."* (Hill, 1990)

The three basic positive rotations are illustrated in figure A2.1.

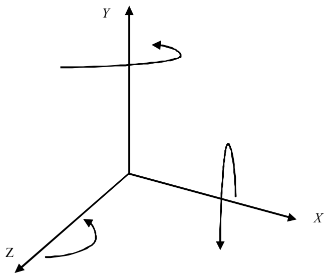


Figure A2.1: Positive Rotations about three axes. (Hill, 1990)

The following three matrices represent transformations that rotate points through an angle β about a coordinate axis. The suggestive notation R*x*(), R*y*() and R*z*() are used to denote *x*-, *y*- and *z*-rolls, respectively. The parameter β is the angle in radians and *c* stands for Cosine (β) and *s* stands for Sine (β). (Hill, 1990)

1. An *x*-roll:

$$Rx(\beta) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & C & -S & 0 \\ 0 & S & C & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

2. A *y*-roll:

$$Rx(\beta) = \begin{pmatrix} C & 0 & S & 0 \\ 0 & 1 & 0 & 0 \\ -S & 0 & C & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

3. A *z*-roll:

$$Rx(\beta) = \begin{pmatrix} C & -S & 0 & 0 \\ S & C & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

The last row and column in all matrices are all zeroes with a lone 1 common to both. This represents that there are no translations involved and the matrices are represented in homogenous coordinates. (Hill, 1990)

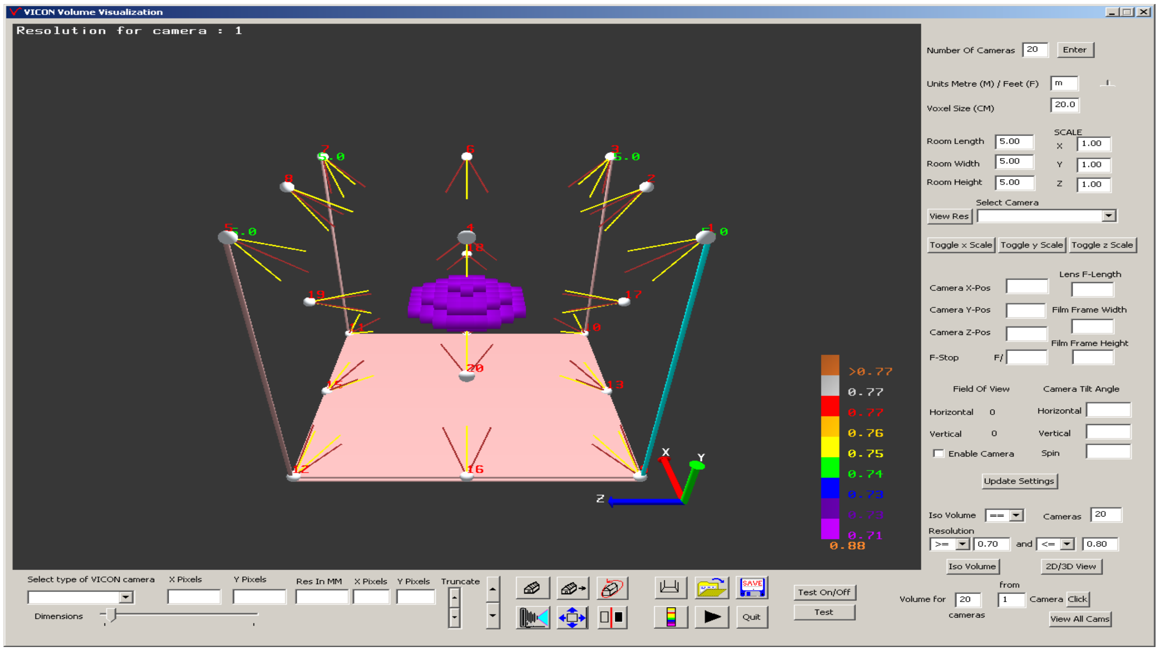
Appendix 3

Screen Grabs and Working Examples

Working examples which introduce usage of the application *VICON Volume Visualization* tool are provided in this chapter. A simple step by step explanation with some data sets that closely match with commonly existing studios and laboratories has been provided. Screen shots along with the explanation provide a clear illustration of the application. All of the data sets described can be found on the CD accompanying this report.

Illustrations using examples

Figure A3.1 shows the screen grab of application with a room of dimensions 5m X 5m X 5m implying length, width and height, respectively. To obtain this view,



VICON Volume Visualization

Resolution for camera : 1

Number Of Cameras 20

Units Metre (M) / Feet (F) m

Voxel Size (CM) 20.0

Room Length 5.00

Room Width 5.00

Room Height 5.00

SCALE X 1.00 Y 1.00 Z 1.00

Field Of View Horizontal 0 Vertical 0

☐ Enable Camera

Iso Volume == Cameras 20

Resolution >= 0.70 and <= 0.80

Volume for 20 cameras from 1 Camera

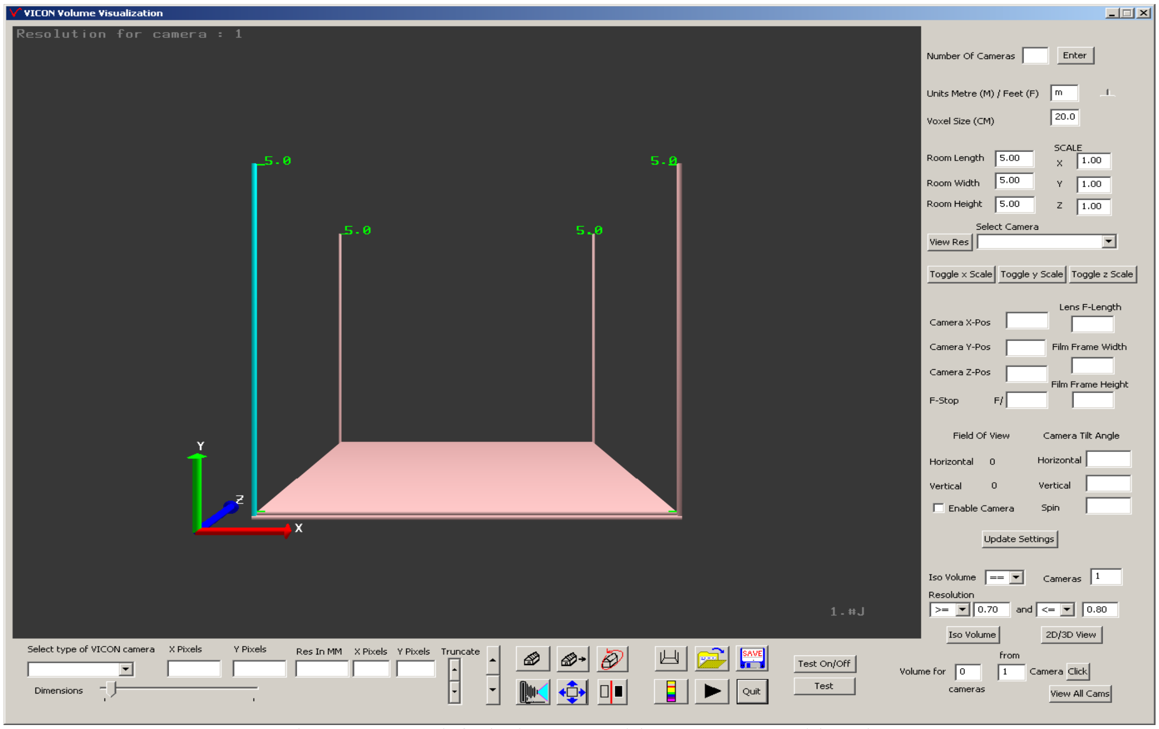
Figure A3.1: A scene with twenty cameras.

First start the application *VICON Volume Visualization* by running the file *VICON VolVis.exe* which can be obtained from the CD accompanying this report.

Initially a room with dimensions of 5m X 5m X 5m implying length, width and height, respectively, will be displayed without any cameras, as shown in figure A3.2.

A data file which describes locations and technical parameters for twenty cameras that results the scene shown in figure A3.1 has already been defined and it can be found on the CD accompanying this report with filename *TwentyCamDesc.txt*.

The file can be loaded by clicking on the File Open button control as shown in table 4.1. The file when loaded, results the scene that is shown in figure A3.1 visualizing capture volume that is common to all twenty cameras. The data file for twenty cameras that produces this scene is shown in figure 4.27.



VICON Volume Visualization

Resolution for camera : 1

Number Of Cameras

Units Metre (M) / Feet (F): m

Voxel Size (CM): 20.0

Room Length: 5.00

Room Width: 5.00

Room Height: 5.00

SCALE X: 1.00 Y: 1.00 Z: 1.00

Field Of View Horizontal: 0 Vertical: 0

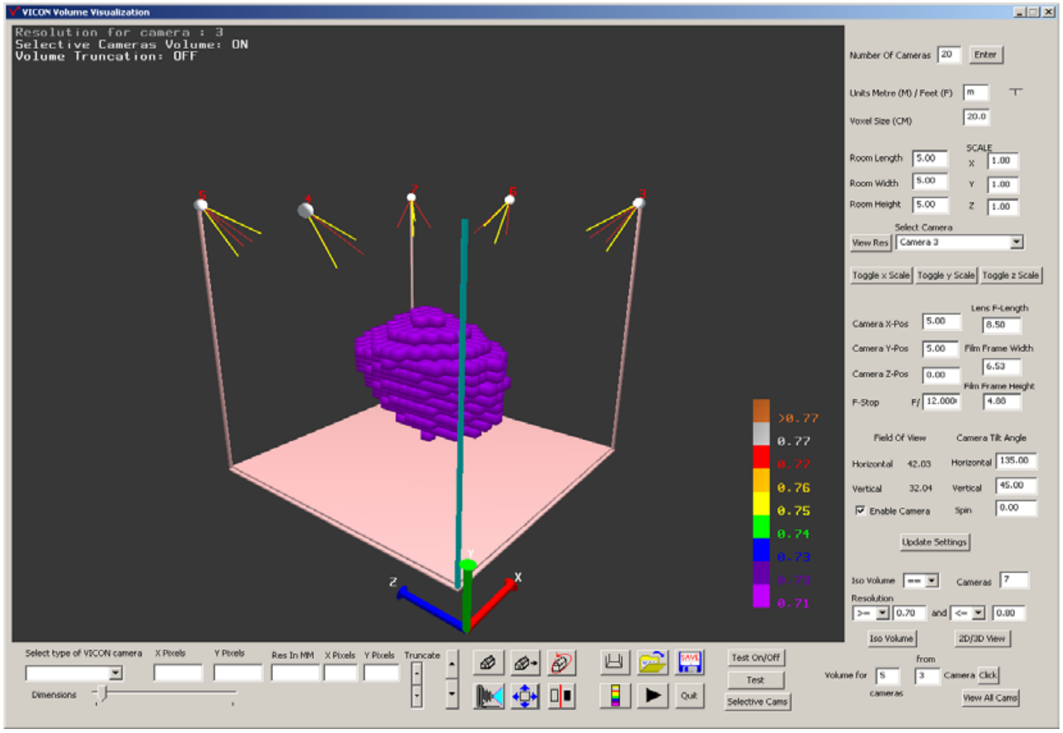
□ Enable Camera

Iso Volume: == Cameras: 1

Resolution: >= 0.70 and <= 0.80

Volume for 0 cameras from 1 Camera

Figure A3.2: A default size room with no cameras positioned.



VICON Volume Visualization

Resolution for camera : 3
Selective Cameras Volume: ON
Volume Truncation: OFF

Number Of Cameras: 20

Units Metre (M) / Feet (F): m

Voxel Size (CM): 20.0

Room Length: 5.00

Room Width: 5.00

Room Height: 5.00

SCALE X: 1.00 Y: 1.00 Z: 1.00

Select Camera: Camera 3

Camera X-Pos: 5.00

Camera Y-Pos: 5.00

Camera Z-Pos: 0.00

F-Stop F/ 12.000

Lens F-Length: 8.50

Film Frame Width: 6.53

Film Frame Height: 4.80

Field Of View Horizontal: 42.03 Vertical: 32.04

Camera Tilt Angle Horizontal: 135.00 Vertical: 45.00 Spin: 0.00

☑ Enable Camera

Iso Volume: == Cameras: 7

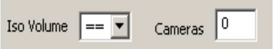
Resolution: >= 0.70 and <= 0.80

Volume for 5 cameras from 3 Camera

Figure A3.3: Volume for five consecutive cameras from camera #3.

Figure A3.3 shows capture volume common to five consecutive cameras starting from camera #3. This can be accomplished by using edit text box controls as described in the section 4.2.12. If only certain numbers of cameras are selected the capture volume visualized is not suitable for truncation. Volume can only be truncated only when visualizing 2D volume and volume of certain number of cameras.

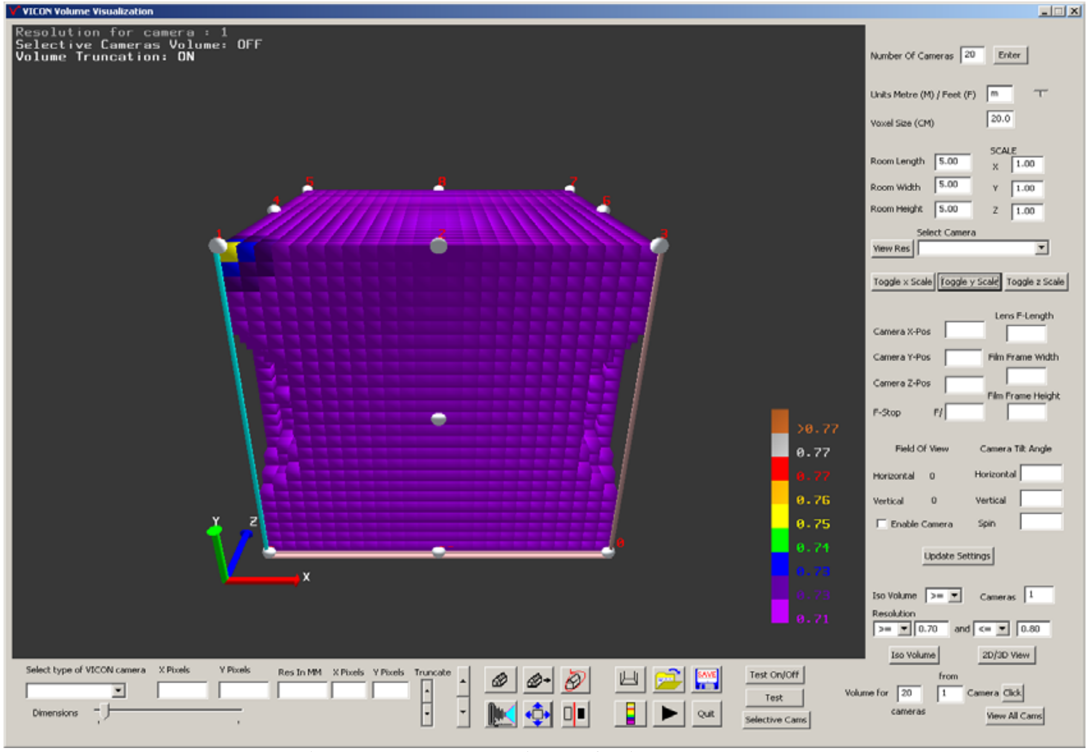
Figure A3.11 shows the volume of room which is not being captured by any of the cameras. This can be achieved by using the controls described in the section 4.2.14. The values of respective controls are shown in figure A3.4 below.



Iso Volume == Cameras 0

Figure A3.4: Values of controls for volume not tracked by any cameras.

Figure A3.5 shows the volume that can be seen by at least one camera. This can be achieved by using controls described in section 4.2.14.



```
Resolution for camera : 1
Selective Cameras Volume: OFF
Volume Truncation: ON
```

Number Of Cameras 20

Units Metre (M) / Feet (F) m

Voxel Size (CM) 20.0

Room Length 5.00 SCALE X 1.00

Room Width 5.00 Y 1.00

Room Height 5.00 Z 1.00

Field Of View Horizontal 0 Vertical 0

Iso Volume >= Cameras 1

Resolution >= 0.70 and <= 0.80

Volume for 20 cameras from 1 Camera

Figure A3.5: capture volume of at least one camera.

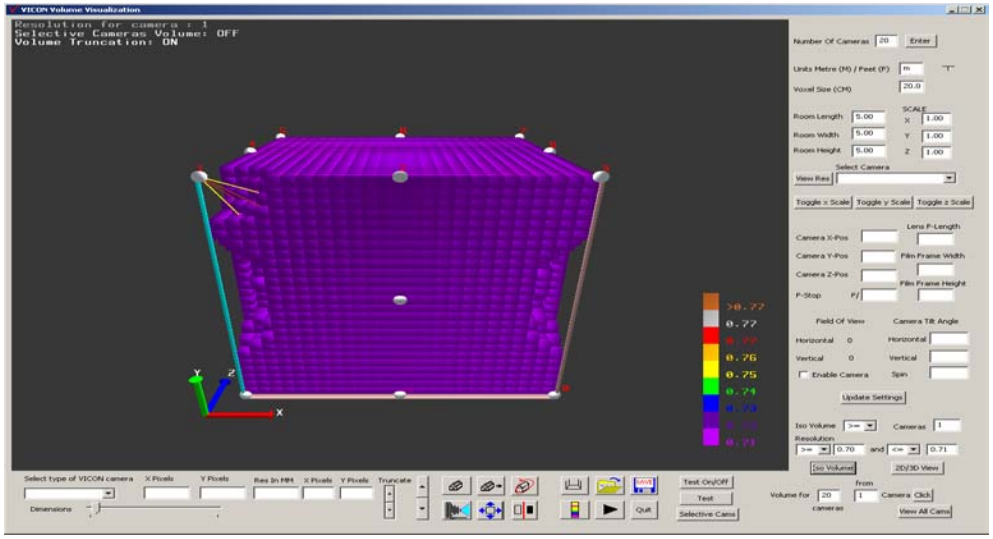
Figure A3.5 shows the resolution colour map with the colours indigo, blue and yellow for the camera one. Figure A3.6 shows the same volume as that of shown in figure A3.5 but with selected resolution range between 0.70mm and 0.71mm, inclusive. It can be observed that volume coloured in indigo, blue and yellow has not been visualized.

Figure A3.7 shows volume for at lest one camera with resolution range between 0.71mm and 0.73mm, inclusive. The resolution that is resulting for a camera are very high precision numbers and hence round off of numbers is a frequent problem is visualizing inaccurate volumes but the resulting volume would not be misleading as practically there is a negligible difference between values 0.7095319002mm and 0.71mm. The round off of value from 0.7095319002 to 0.71 during comparison results in a minute part of volume being discarded.
Figure A3.8 shows truncated volume that is visualized in figure A3.7. This can be achieved by using controls described in the section 4.2.16.

Figure A3.9 depicts orthographic view of capture shown in figure A3.1. This can be achieved by using button controls described in table 4.1.

Figure A3.10 depicts the scenes rendered with and without room surrounding the capture volume. This can be achieved by using button controls described in table 4.1. It can be observed that only four cameras were in the scene.

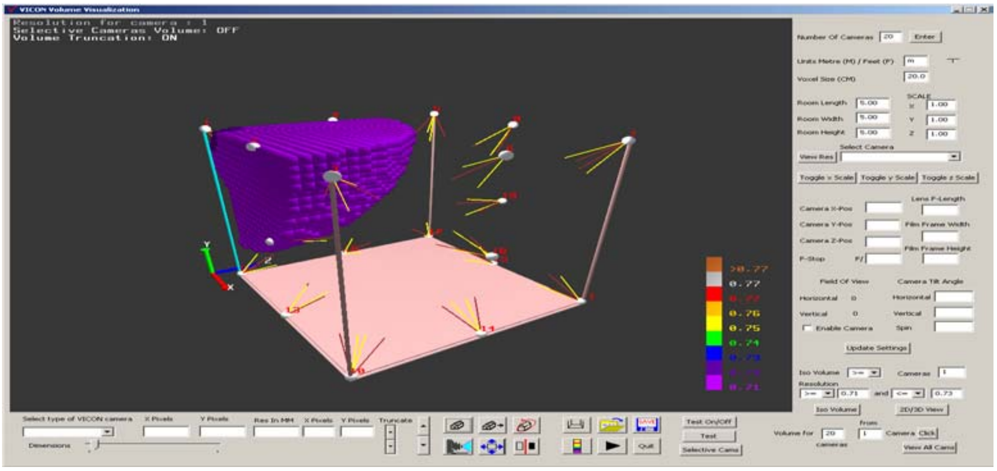
Any number of cameras can be initialized at any point of time using the control described in the section 4.2.2. First four cameras are by default placed at top four corners of the room and if less than four cameras were initialized, say two, those two cameras will be placed on top left front corner of the room and top right front corner of the room in order proceeding in that direction (CCW) up to four corners if four cameras were given. If more than four cameras were initialized using the control described in the section 4.2.2, first four will be placed on top of four corners of room and the rest will be positioned at the centre point of the room.



```
Resolution for camera : 1
Selective Cameras Volume: OFF
Volume Truncation: ON
```

Number Of Cameras: 20
Units Metre (M) / Feet (F): m
Voxel Size (CM): 20.0
Room Length: 5.00 · SCALE X: 1.00
Room Width: 5.00 · Y: 1.00
Room Height: 5.00 · Z: 1.00
Field Of View: Horizontal 0, Vertical 0
□ Enable Camera
Iso Volume >= Cameras 1
Resolution >= 0.70 and <= 0.71
Volume for 20 cameras from 1

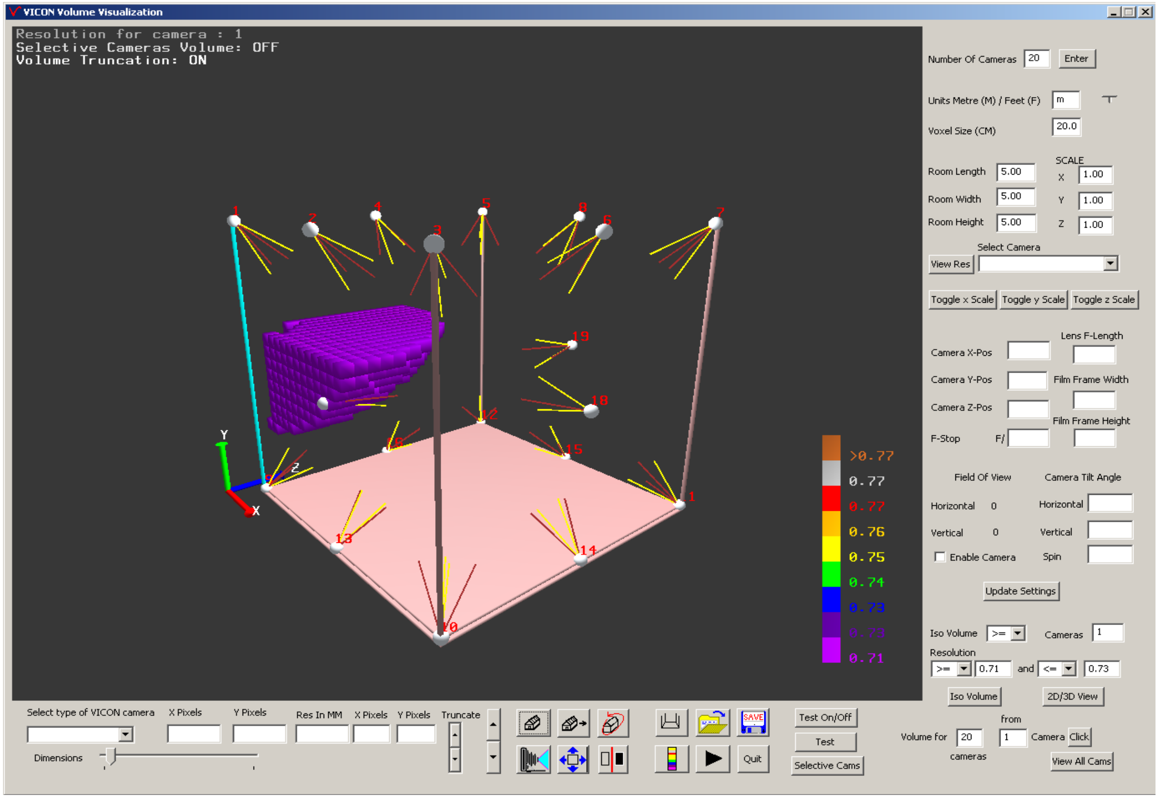
Figure A3.6: Capture volume for at least one camera within the selected resolution range 0.70mm and 0.71mm, inclusive.



```
Resolution for camera : 1
Selective Cameras Volume: OFF
Volume Truncation: ON
```

Number Of Cameras: 20
Units Metre (M) / Feet (F): m
Voxel Size (CM): 20.0
Room Length: 5.00 · SCALE X: 1.00
Room Width: 5.00 · Y: 1.00
Room Height: 5.00 · Z: 1.00
Field Of View: Horizontal 0, Vertical 0
□ Enable Camera
Iso Volume >= Cameras 1
Resolution >= 0.71 and <= 0.73
Volume for 20 cameras from 1

Figure A3.7: Capture volume for at least one camera within the selected resolution range 0.71mm and 0.73mm, inclusive.



```
Resolution for camera : 1
Selective Cameras Volume: OFF
Volume Truncation: ON
```

Number Of Cameras 20

Units Metre (M) / Feet (F) m

Voxel Size (CM) 20.0

Room Length 5.00 SCALE X 1.00

Room Width 5.00 Y 1.00

Room Height 5.00 Z 1.00

Select Camera

Camera X-Pos

Camera Y-Pos

Camera Z-Pos

F-Stop F/

Lens F-Length

Film Frame Width

Film Frame Height

Field Of View Horizontal 0 Vertical 0

Camera Tilt Angle Horizontal Vertical Spin

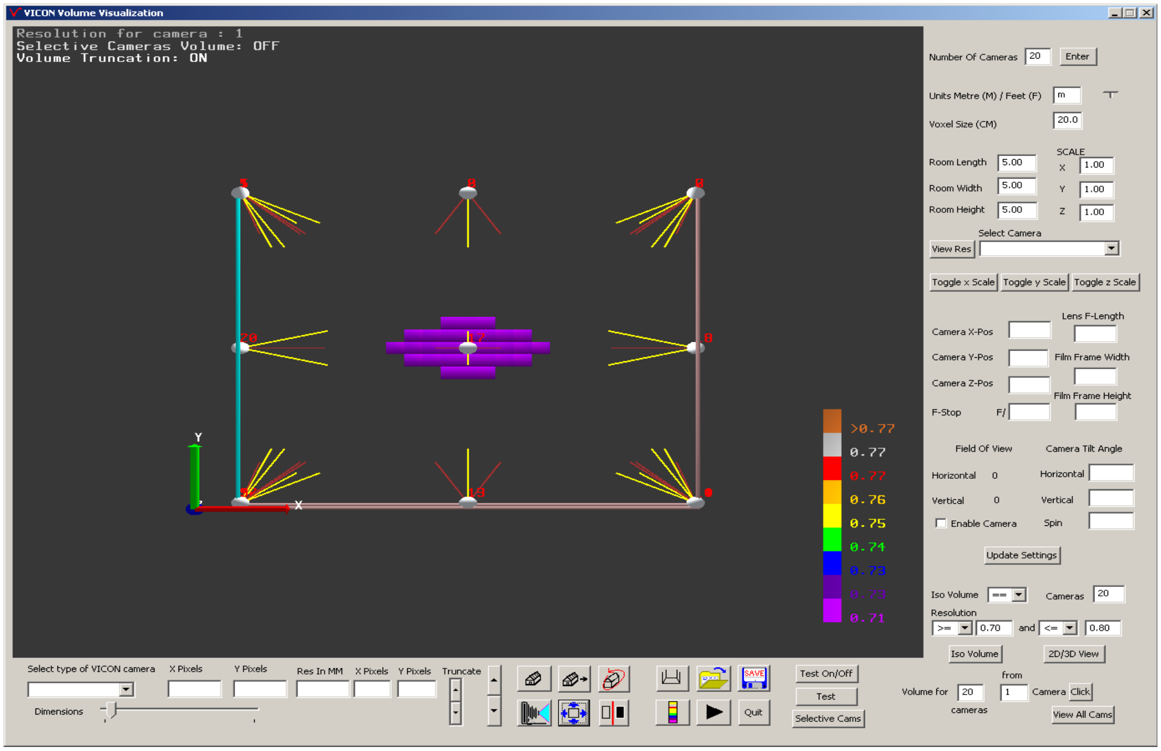
□ Enable Camera

Iso Volume >= Cameras 1

Resolution >= 0.71 and <= 0.73

Volume for 20 cameras from 1 Camera

Figure A3.8: Truncated Capture volume for at least one camera within the selected resolution range 0.71mm and 0.73mm, inclusive.



```
Resolution for camera : 1
Selective Cameras Volume: OFF
Volume Truncation: ON
```

Number Of Cameras 20

Units Metre (M) / Feet (F) m

Voxel Size (CM) 20.0

Room Length 5.00 SCALE X 1.00

Room Width 5.00 Y 1.00

Room Height 5.00 Z 1.00

Select Camera

Camera X-Pos

Camera Y-Pos

Camera Z-Pos

F-Stop F/

Lens F-Length

Film Frame Width

Film Frame Height

Field Of View Horizontal 0 Vertical 0

Camera Tilt Angle Horizontal Vertical Spin

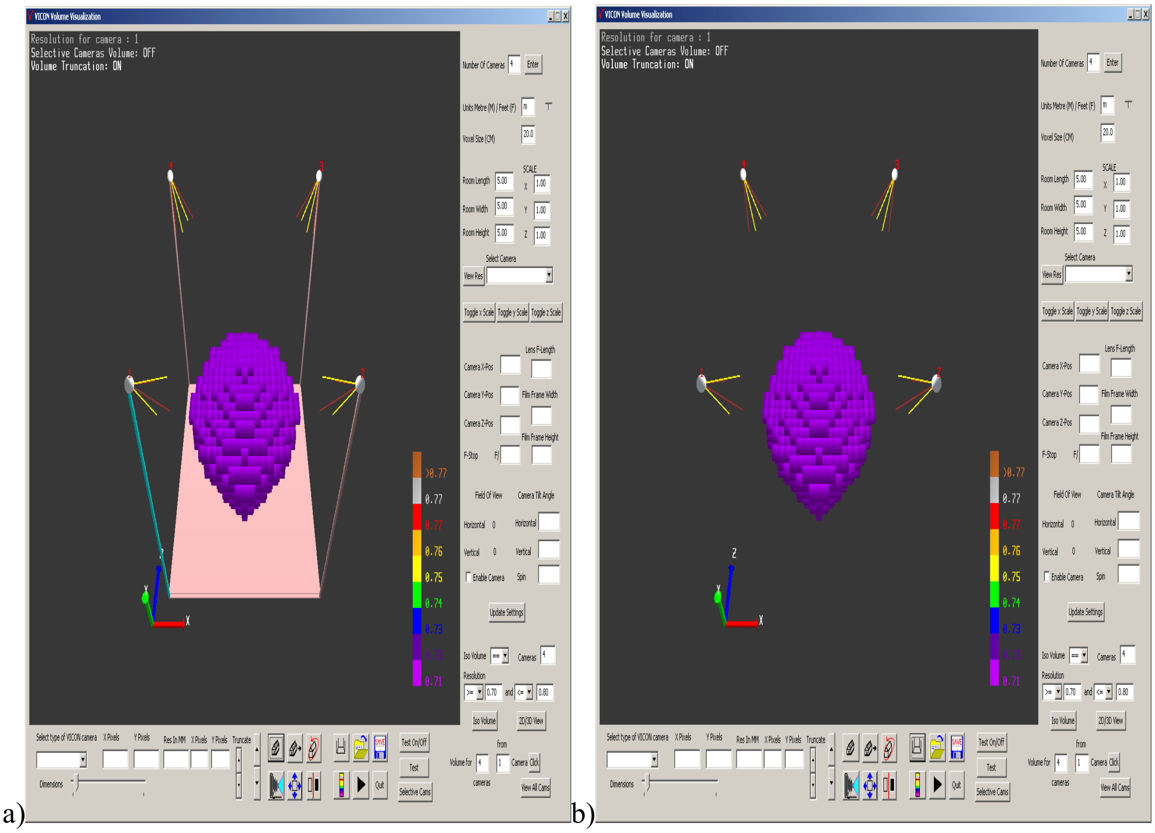
□ Enable Camera

Iso Volume == Cameras 20

Resolution >= 0.70 and <= 0.80

Volume for 20 cameras from 1 Camera

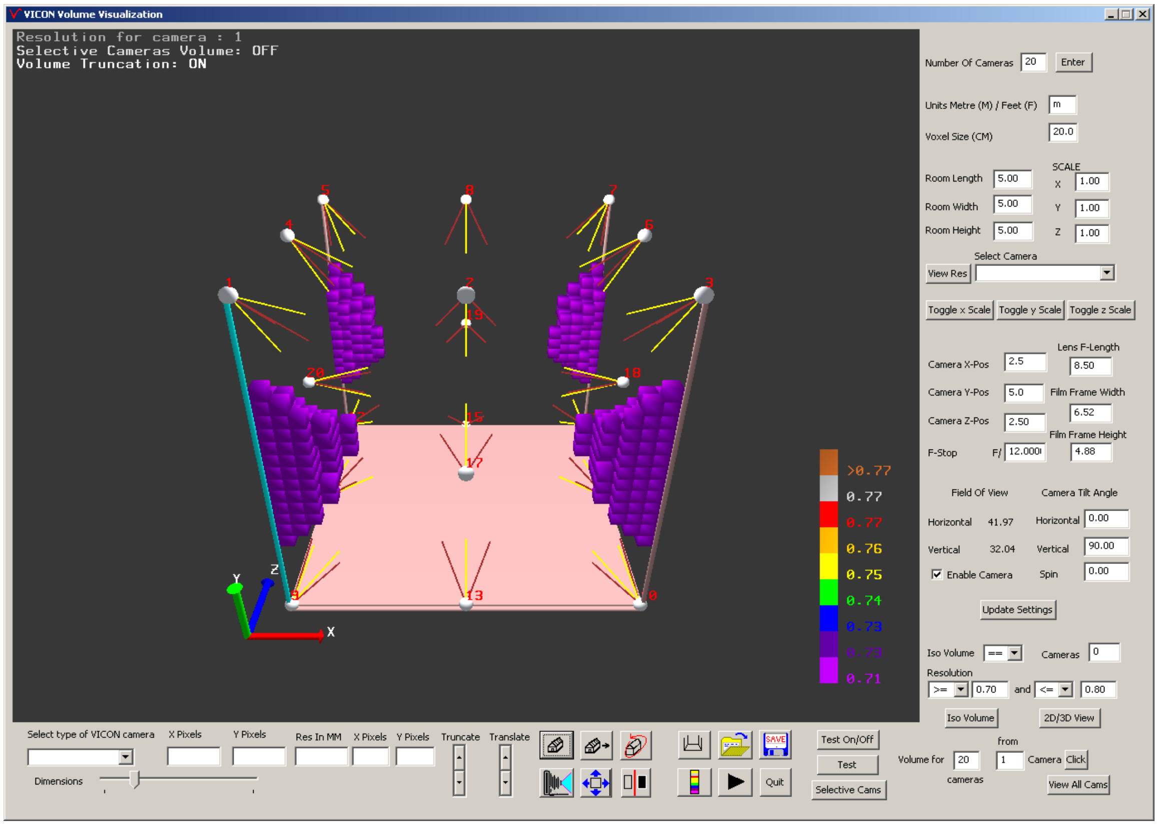
Figure A3.9: Orthographic view of capture volume common to 20 cameras



Resolution for camera : 1
Selective Cameras Volume: OFF
Volume Truncation: ON

Number Of Cameras 4
Units Metre (M) / Feet (F) m
Voxel Size (CM) 20.0
Room Length 5.00 X 1.00
Room Width 5.00 Y 1.00
Room Height 5.00 Z 1.00
Horizontal 0
Vertical 0
Iso Volume == Cameras 4
Resolution >= 0.70 and <= 0.80
Volume for 4 cameras from 1 Camera Click

Figure A3.10: a) Scene with room. b) Scene with rendering of room disabled.



Resolution for camera : 1
Selective Cameras Volume: OFF
Volume Truncation: ON

Number Of Cameras 20
Units Metre (M) / Feet (F) m
Voxel Size (CM) 20.0
Room Length 5.00 X 1.00
Room Width 5.00 Y 1.00
Room Height 5.00 Z 1.00
Camera X-Pos 2.5
Lens F-Length 8.50
Camera Y-Pos 5.0
Film Frame Width 6.52
Camera Z-Pos 2.50
Film Frame Height 4.88
F-Stop F/ 12.000(
Field Of View
Horizontal 41.97
Vertical 32.04
Camera Tilt Angle
Horizontal 0.00
Vertical 90.00
Spin 0.00
☑ Enable Camera
Iso Volume == Cameras 0
Resolution >= 0.70 and <= 0.80
Volume for 20 cameras from 1 Camera Click

Figure A3.11: Volume not being covered by any of the cameras.

References

Brodlie K W (Ed), 1992, *Visualization Techniques*, in Brodlie K W et al. (Eds.), *SCIENTIFIC VISUALIZATION: TECHNIQUES AND APPLICATIONS*, Germany: Springer – Verlag, pp 37-83.

Computer Support Center, "nd", *Information Technology* [online], Computer Support Center, Southern Illinois University Carbondale. Available: http://www.infotech.siu.edu/csc/help/glossary.htm [Accessed 02 August 2004].

Digicamhelp.com, 2004, *APERTURE – THE IRIS OF THE CAMERA* [online], Digicamhelp.com. Available: http://www.digicamhelp.com/advanced-digital-camera-settings/aperture.htm [Accessed 23 June 2004].

Fitzgerald, 2000, *Film Formats Larger than standard 35mm* [online]. Available: http://www.geocities.com/Hollywood/Makeup/4303/largeformats.html [Accessed 21 May 2004].

FunctionX, 2003, *The Slider Control* [online], FunctionX Inc. Available: http://www.functionx.com/visualc/controls/slider.htm [Accessed 08 July 2004].

Guidance Laboratory, 2004, *Icons* [online], Guidance Laboratory Inc. Available: http://citymap.slyip.com/f.php?f=6201&r=401 [Accessed 10 August 2004].

Hill F. S, 1990, *COMPUTER GRAPHICS USING OPEN GL*, 2nd edition, USA: Prentice-Hall Inc.

Hussain, 1998, *Microsoft Foundation Classes Part 1 Visual/Windows Programming* [online]. Available: http://web.mit.edu/dhussain/insanemath/MFC.htm [Accessed 08 June 2004].

Microsoft, 2004a, *CSpinButtonCtrl* [online], MSDN, Microsoft Corporation. Available: http://msdn.microsoft.com/library/default.asp?url=/library/en-us/vcmfc98/html/_mfc_cspinbuttonctrl.asp
[Accessed 03 August 2004].

Microsoft, 2004b, *CComboBox Class, MFC Library Reference* [online], MSDN, Microsoft Corporation.
Available: http://msdn.microsoft.com/library/default.asp?url=/library/en-us/vclib/html/_mfc_ccombobox.asp
[Accessed 04 August 2004].

Microsoft, 2004c, *Working with List boxes, text Boxes and Check Boxes* [online], Microsoft Corporation. Available: http://msdn.microsoft.com/library/default.asp?url=/library/en-us/cdo/html/_olemsg_working_with_list_boxes_text_boxes_and_check_boxes.asp [Accessed 04 August 2004].

Parente P, 2002, *LEGO Pinhole Camera* [online]. Available: http://www.cs.unc.edu/~parente/igv/hw1/parente_hw1.html [Accessed 23 June 2004].

Rigg J, 2002, *Calculating focal length* [online], Panoguide.com. Available: http://www.panoguide.com/reference/focal_length.html [Accessed 21 May 2004].

Scott P J, 1995, *Window-Viewport Transformations*, *Fundamentals of Computer Graphics* [online], JPS Graphics. Available: http://195.195.128.170/staff/scotty/fcg/wvxform.htm [Accessed 23 June 2004].

SparkNotes, 2004, *Solid As A Rock*, *The NewSAT* [online], SparkNotes, LLC, Barnes and Noble Learning Network. Available:
http://www.sparknotes.com/testprep/books/newsat/chapter20section7.html
[Accessed 19 July 2004].

Tebbutt P, Wood J and King M, 2002a, *The VICON Manual*, VICON Motion Systems, OMG Plc. (vicon_manual_v1_2.pdf on the CD accompanying this report)

Tebbutt P, Wood J and King M, 2002b, *Configuring the system*, *The VICON manual*, VICON Motion Systems, OMG Plc. (preparation_v1_2.pdf on the CD accompanying this report)

VICON, "nd"a, *VICON Cameras – Specifically designed for optical motion capture* [online], VICON Motion Systems, OMG Plc. Available
http://www.vicon.com/main/support/downloads/tech%20sheets/VMS027S_cameras_hi-res.pdf
[Accessed 30 July 2004].

VICON, "nd"b, *VICON Motion Systems – VICON Products* [online], VICON Motion Systems, OMG Plc.
Available http://www.vicon.com/main/technology/mcam2.html [Accessed 30 July 2004].

VICON, "nd"c, *VICON Motion Systems – Lcams* [online], VICON Motion Systems, OMG Plc. Available http://www.vicon.com/main/technology/lcam.html [Accessed 01 August 2004].

VICON, "nd"d, *VICON Motion Systems – Miscellaneous Cameras* [online], VICON Motion Systems, OMG Plc. Available http://www.vicon.com/main/technology/ircam.html [Accessed 01 August 2004].

VICON , "nd"e, *VICON Systems* [online], VICON Motion Systems, OMG Plc. Available:
http://www.hedcom.fi/files/hedcom/stig/CompleteBrochure.pdf [Accessed 23 June 2004].

Webopedia, 2004, *Voxel* [online], Jupiter Media Corporation. Available:
http://www.webopedia.com/TERM/V/voxel.html [Accessed 03 August 2004].

WorldHistory.com, 2004a, *Focal Length*, *Focal Length in the news* [online], WorldHistory.com. Available: http://www.worldhistory.com/wiki/f/focal-length.htm [Accessed 04 August 2004].

WorldHistory.com, 2004b, *F-Number, F-Number in the news* [online], WorldHistory.com. Available http://www.worldhistory.com/wiki/f/F-number.htm [Accessed 23 June 2004].