ACHIEVING EFFICIENT REAL-TIME VIRTUAL REALITY ARCHITECTURAL VISUALISATION

HII JUN CHUNG, DANIEL

NATIONAL UNIVERSITY OF SINGAPORE

2007

ACHIEVING EFFICIENT REAL-TIME VIRTUAL REALITY ARCHITECTURAL VISUALISATION

HII JUN CHUNG, DANIEL
B.A.Arch (Hons.) NUS

A THESIS SUBMITTED

FOR THE DEGREE OF MASTER OF ARTS IN ARCHITECTURE

DEPARTMENT OF ARCHITECTURE

NATIONAL UNIVERSITY OF SINGAPORE

ACKNOWLEDGEMENTS

Millions of thanks and gratitude to all the following people who have been giving me a lot of assistance for the past 2 years to make this thesis a reality:

- God - for blessings of everything and providing me people to guide and help me
- Wife - for all the love in the world
- Family - full support (economical, morale, physical, mental, psychological)
- Assistant Professor Dr. Tan Beng Kiang (research supervisor) - research advice, discussion, unlimited time, unlimited effort, patience, dedication, hardwork
- Assistant Professor Dr. Stephen Wittkopf - research advice, technical issues discussions, thesis discussions (especially during my architecture research think thank (arTT) presentations)
- Associate Professor Dr. Tan Chee Keong, Willie - statistics, thesis format discussions and advice
- Ms. Chua Yew Lan - proofreader and editor of this thesis
- Dr. Dave Bharat, Dr. Thomas Kvan, Dr. Thomas Fischer & Dr. Ramesh Krishnamurti - thesis discussion during and after CAADRIA2007 Postgraduate Consortium in Southeast University, Nanjing, China
- Mr. Roni Anggoro, Mr. Zhang Ji, Mr. Wykeith Ng & Mr. Chan Myre Lwin - technical discussions and advice (3ds Max® and EON^{TM} Studio^{TM} Professional^{TM})
- Mr. Chew Chun Boon, Mr. Chris Hee Kee Kwang, Mr. Siow Ming Khang - EON^{TM} Studio^{TM} Professional^{TM} experiment preparations and executions
- Mr. Alex Teoh Song Khong - statistics discussions and advice
- Mr. Hong Kai Shing - virtual reality lab support, technical discussions and advice
- Mr. Nils Andersson - EON^{TM} Studio^{TM} Professional^{TM} support and discussions
- Professor David Robson - Geoffrey Bawa information support
- Mr. Leong Wey Hsien - 3ds Max® tips, support and advice
- Mr. Ian Loh - computer scientist, real-time rendering discussions
- Mr. Goh Siong Thye - mathematics undergraduate, statistics discussions
- Dr. Chiu Shui Chan - virtual reality systems and CAVE discussions
- Associate Proffessor Heng Chye Kiang - permission to use models of Chang'An
- Ms. Soon Lay Kuan & Ms. Gauri Bharat - thesis format discussions and advice
- Mr. Kenneth Choo Wee Khiam - Quest3D® support
- Ms. Tan Puay Yong, Cindy, Ms. Katherine Chong Kwang Ping and Department of Architecture for sharing the digital archives collection of students 3D model design projects as experiment samples
- Quest3D® and EON^{TM} Studio^{TM} Professional^{TM} forum members – knowledge exchange and discussions

ABSTRACT

The value of using virtual reality (VR) for architectural visualisation is that it allows a user to walk through a building in real-time and experience it. Real-time rendering is frames replacement that is fast enough for human eyes detection. If there is lag in navigation through the 3D model, then it defeats the purpose of using VR visualisation. In such instances, many steps are done through trial and error to optimise the model so that it can run smoothly in real-time. It is often difficult to predict how much time and effort is required to create a satisfactory architectural VR visualisation. Using very powerful hardware may solve part of the problem. However, most architecture schools and firms cannot afford to constantly upgrade their hardware. Therefore, the best way is to make full use of the available resources in the fastest and smartest way possible. The objectives of this thesis are to identify ways to achieve efficient real-time VR architectural visualisation by optimising selected factors, predict performance and propose a faster and efficient workflow.

The research initially looks at the techniques of optimisation for VR visualisation. These techniques are then used in independent tests to find their individual relationships with frame rate, time taken to travel a distance, vertex memory and texture memory. Next, the quantitative and qualitative aspects of the VR visualisations are tested through experiments and a survey. The variables chosen for both are from findings of the earlier optimisation techniques and independent tests.

The quantitative aspect is done by conducting experiments with 105 3D models using a VR software on different hardware platforms. Frame rate is measured against four fundamental variables i.e. triangle, vertex, geometry and texture count. The statistical method of simple multiple regression is used to identify the relationships among them against frame rate and to derive equations to predict performance of frame rate.

As for the qualitative aspect, a survey is conducted to identify the minimum visual quality acceptable to users for three variables. The variables are triangles complexity, texture resolution and frame rate. Knowing the minimum acceptable visual quality will eliminate the wastage of processing power and memory to generate complex files if they do not contribute much to improved visual quality.

The findings are integrated into a workflow and best practice guideline for those who wish to use VR technology for presentation and visualisation of spaces. It aims at helping them create VR projects in the fastest possible manner and be more productive in VR visualisation with the resources available to them.

CONTENT

LISTS OF TABLES AND FIGURES

TABLES

FIGURES

CHAPTER 1.0: INTRODUCTION

In the field of architecture today, we are particularly interested in the capabilities of the virtual reality (VR) visualisation. VR is a substitute of being there at the real place. Static two-dimensional representations on canvas and computer screens do not represent the whole picture and sometimes, we need a series of them to present the full picture from different directions.

Representation of the real world is important because it helps us understand spaces that are yet to be built or spaces that no longer exist . We can understand how original building looked like by reconstructing completely destroyed buildings and ruins. VR can help in visualising different architectural disciplines from all possible views, including history, construction, mechanical and electrical services, technology, structure, material, design as well as detailing. The interest in visualisation started with 2D drawings and perspectives on traditional mediums to the current advanced mediums of computer still renderings, animations and real-time renderings.

The full picture can only be perceived, expressed, evaluated and appreciated with a dynamic walkthrough of the whole architecture. Traditional architectural presentations including 2D drawings, 3D still renderings and animations only allow us to be passive viewers. Virtual reality goes beyond these computer simulations and visualisations as it provides two-way interaction between humans and computers. This human-computer interaction in real-time is what makes the viewers active participants in the architectural presentation and interacting via input devices. The participants are actively involved in the presentation, because they can engage in the presentation and interact via input devices.

In addition, VR visualisation allows us to generate a 3D-designed world before it is built. The ability to visualise on big screens means that we can see the built environment on a real 1:1 scale. We can therefore gauge and feel the size of the spaces we design, so that they will not be in the wrong proportions. Using stereoscopic effect in VR also imparts depth and enables us to see how spaces connect to each other. We can see and feel the distance in the crafted spaces and make the necessary adjustments before the design is actually built. This potentially translates into huge cost savings than if the changes were to be made much later during the construction stage.

VR visualisation also gives us freedom of navigation. This is useful because we can literally walk or fly in real-time through spaces in whichever direction we choose to. This sense of immersion means that we are surrounded by the environment and we feel that we are there in the environment. It is also known as the sense of presence since you believe you are in a particular scene simulated by VR visualisation. This is what 3D still perspectives and animations fail to do.

Finally, we can do things beyond normal visualisation including looking through details, having information and dimensions all over space, sectional perspectives, and x-ray visions of services and structures behind the physical ceiling, wall and floor as we navigate around.

A disadvantage of VR technology is the high cost of investment. Depending on the type of technology invested in, we will require space to mount the screen, projectors and the speaker systems. Head-mounted display system is still very rare and costly. Also, only one person can view the same presentation from the same perspective view at a time. Alternatively, there is augmented reality, but its strength lies more in the ability to superimpose the real world with information generated by computers. The difficulties are the inability to generate huge amounts of information quickly and wirelessly in real-time, and the short battery lifespan to support completing the tasks at hand within an acceptable time period. The other major problem is cybersickness. People wearing head-mounted displays differ in their tolerance and acceptance of the speed of walking, turning and the amount of objects seen at the same time during navigation. The limited display area on mobile devices is another problem. The capabilities of mobile devices in generating 3D graphics are still very limited at this stage.

Ideally, virtual reality should be real-time in order to imitate the built environment. In reality, hardware and software are still not capable of generating real-time presentations of scenes of a huge scale or with immense details in a convincing manner. It is necessary to optimise what we can do with current technology, so that we can make full use of it with the least time and effort. Thus, this research aims to propose the best possible way to render a real-time VR presentation with current available technology. The complex projects we currently generate can barely be presented in real-time convincingly. The research looks at overcoming this frustration and the frustration of re-doing models to improve the performance.

1.1 Research Problems

In architecture, no real priority has been given or serious study done on an efficient 3D model construction process to reach the final product of presentation, be it a 3D still perspective rendering, an animation or a VR simulation. The process of getting a 3D model ready for visualisation is always time-consuming and the number of repetition cycles required to reach a satisfactory finished product is sometimes unpredictable. Apart from that, navigation in the VR visualisation will be slow when we deal with huge scenes (Steed 1997).

If the visualisation becomes too slow, we have to go back to earlier cycles such as the modelling or texture-mapping stage to improve it. Sometimes, when time is limited, some desired aspects of the VR project have to be compromised. An example is removing 3D trees in the scene and replacing them with a smaller number of flat tree textures. A detailed organic sculpture or complex furniture in a 3D scene may have to be replaced with simple linear versions. All these sacrifices are done because the final VR project cannot run in real-time. Ultimately, what is shown to the audience is not what the final design is suppose to be because the tool cannot visualise what we want well.

Most established architecture firms learn to perform certain processes more rapidly through experience. Similarly design students learn from experience, from peers or seniors, or from books and the internet. VR presentations are at the highest level of all these processes – above 2D drafting, 3D model construction, 3D still perspectives and video animation. The number of VR projects done is definitely less than 2D drawings, 3D model still perspectives and video animation because VR is a comparatively new technology.

The rarity of doing VR projects means that most people have less experience and they normally apply the same methods from conventional visualisation practice to produce VR projects. The process of preparing a VR presentation requires 3D model construction, applying textures and lighting, and if required, animation paths and other features required by specific projects. Most of the time, preparing very complex projects without planned strategies will result in too much time and effort being spent. It will take weeks or months to finish the job if no proper planning is done.

The price to pay for running simulations in the fastest way possible and utilizing the latest features is constant upgrade. Architecture schools and firms do not have the luxury of constant upgrades of the latest hardware and software versions. Everyone has a budget of their own to run at their own economic means and timeframe. A project that requires more time to finish will have added cost. Therefore, the only way to save time and money is to make full use of available resources in the fastest and smartest way possible. In general, hardware and software will be used for a period of two to three years. This research finding can help us make full use of the resources available during this period before the next upgrade. The samples of architectural VR projects in this research can be reused for re-calculation of the impact of each variable towards performance when the next generation of computer technology arrives.

The field of architecture highly demands having the best texture, quality of 3D models and even more light sources depending on the needs and requirements of different projects to showcase designs. This is a challenge because fulfilling the ideal of having as many triangles in a scene with the best textures in order to obtain the most realistic 3D scene will be too much for most systems to handle. Other fields, apart from gaming (Omernick 2004), may not demand as much quality for visualisations. The assembly of engineering motor parts, for example, does not even require textures. Colours applied would be sufficient for their needs with perhaps just a default ambient light to illuminate the scene.

The preparation processes for most complex or detailed projects will take a longer time, and include 3D rendering, animation and VR presentations. For VR presentation of these complicated projects, running real-time fails because of ghosting and lagging. The computer will struggle to run complicated 3D scenes and it is not ideal to navigate, resulting in cybersickness.

Figure 1.1 shows the steps to go through to reach the virtual reality stage. Many things need to be done in and between those steps before exporting out to the next level, and this complicates matters. A lot of trial and error will be required throughout the entire cycle and re-doing some of the stages add to wasted time and effort.

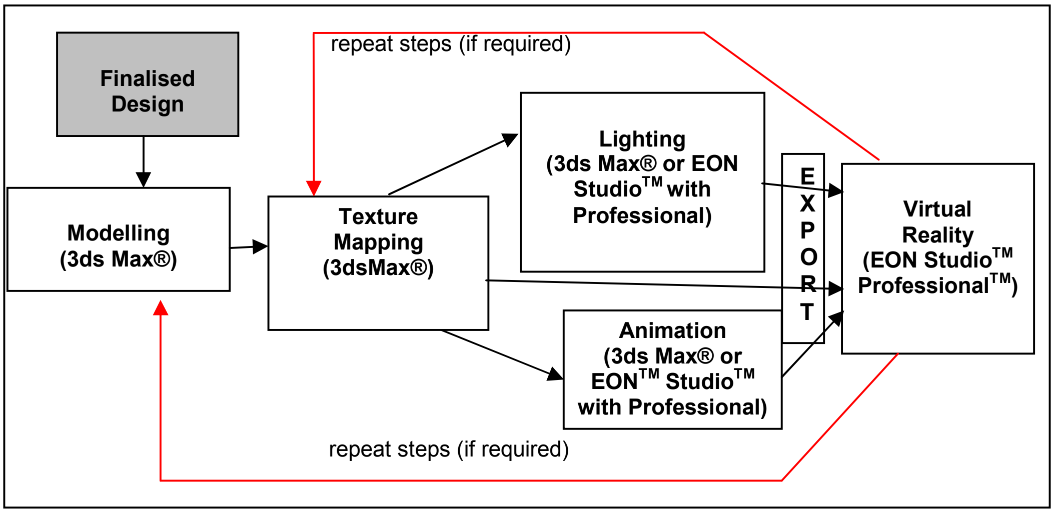


Figure 1.1. Virtual Reality Project Workflow

From literature review, there is very few research done on good workflow or best practice for VR architectural visualisation.

1.2 Research Objectives and Definition

1.2.1 Research Objectives

The research objective is to achieve efficient real-time virtual reality architectural visualisation. It is essential to research into making the visualisation efficient because it will help speed up the process of creating virtual reality architectural presentations. A non-efficient way is to optimise everything, no matter how much impact it will have on the final performance. This will waste a lot of time and effort. Most people learn from experience but sometimes even through trials and errors, they still may not optimise the crucial factors that affect performance the most.

The objectives of this research are:

(i) **To explore the techniques of optimisation that improve VR performance.** These techniques are identified from literature reviews as well as from the ones available in software. They help produce 3D models with the same visual quality by utilizing the least computer processing power and memory consumption.

(ii) **To explore the individual factors that affect VR performance.** These factors are elements that produce the 3D model, its physical appearance, as well as features that enhance the whole VR visualisation. All these elements and features contribute to a decline in performance. Hence, the intention is to identify the extent of impact on frame rate, time taken to travel a distance, and memory consumption. The common acknowledgement is that good model management (Burdea and Coiffet 2003), texture management, light management, and all the variables involved in the final 3D model come hand-in-hand for an overall efficient model for virtual reality visualisation.

(iii) **To explore the collective relationship among fundamental factors.**

(iv) **To identify the biggest contributors among the fundamental factors.**

(v) **To propose an equation that predicts performance from fundamental factors.** This is the quantitative aspect of the research which measures hardware performance. An experiment is conducted for the four most fundamental variables, i.e. vertex count, triangle count, geometry count and texture count (Wimmer and Wonka 2003) to run together in the simple multiple regression method against frame rate, which is the measurement to achieve real-time simulations. The aim is to observe the trend on different hardware setups, to predict frame rate from equations created from simple multiple regression, and to determine the weightage of each variable on the whole performance.

(vi) **To explore the human acceptance of 3D model visual quality.** This is the qualitative aspect of the research which measures human preference. Human acceptance of three factors of 3D model visual quality will help to efficiently plan for VR simulations. This is critical because each element can then be effectively designed to fulfill the accepted quality requirements without consuming too much computer processing power and memory. This ensures no wastage of resources. Savings of leftover processing power and memory will come in handy, especially when there are last minute features and 3D model parts need to be added to an already complicated project.

(vii) **To propose a workflow for VR users.** All the research objectives mentioned previously will be combined to achieve the final objective - that is to propose a guideline for VR users. This will greatly help the process of planning and building VR projects, especially when complex models with complicated parts are involved. The workflow will provide users with step-by-step instructions on how to reach the final stage of creating a VR presentation in the fastest way.

Some may argue that it is unnecessary to uncover the biggest contributing factors since hardware will always be improving over time, and the research results will be obsolete once the next generation of hardware is released to the market. However, there is compelling reason to do so because as the semiconductor's limit is being reached, hardware upgrades are slowing down.

Although hardware is improving, the computer chip has basically reached its practical limit of around 3GHz given the current materials of construction. Now, CPUs are being created with more cores- from single core to duo core to quad core for both Intel® and AMD, the leading CPU-makers in the world. The main problem is that multicore CPUs are not being taken advantage of by most software in the market. This is also the case for EON™ Studio™ Professional™ - the virtual reality visualisation software used in this research. It is therefore worthwhile to explore computers at the current upper limit of technology, as it will likely stay that way for quite some time

Slater and Chrysanthou, authors of the book "Computer Graphics and Virtual Environments: from Realism to Real-Time" have this to say about hardware performance:

> "One of the main requirements for a believable experience in a virtual environment is a high and constant frame rate. One might think that this will eventually be achieved through exploiting the faster and more powerful machines that are (always) coming onto the market. However, the size and complexity of the models as well as the expectations of the user tend to more than cancel out any benefits provided by hardware improvements. In spite of the exponential improvement in hardware performance, there remains a need for algorithms that can reduce the rendered geometry to a manageable size, without compromising the resulting image."
>
> (Slater and Chrysanthou, 2006).

1.2.2 Definitions

Efficiency

In this case, being efficient means the ability to create a VR presentation in the shortest possible time with minimal efforts. It also includes creating a VR presentation with the best possible quality within the constraints of the available computer processing power and memory. In other words, we will be able to create a presentation using the least input to generate the most output.

Real-time

Real-time means that the VR presentation is able to respond to external input and processes almost instantaneously, as what happens in real life**.** In other words, the navigation process feels like how it should in the real world. It has to realistically portray movements so that it feels natural. VR for this research will mean a simulation with 3D models where users have the freedom to navigate in a 3D space whichever way they desire. It provides more options of visualising architectural scenes compared to still rendering and video animation because of what one can do within it according to one's preference.

Virtual Reality

VR used in this research means 360 degrees of navigation freedom with visualisation on a flat screen display from back-lit projectors in stereoscopic mode. Other than visual input, the research does not take into account any other sensorial input performances. Therefore, surround sound and touch are not part of it. The VR simulations used in the experiments are all related to architecture and measured for technical performance and acceptable visual appearance by humans only. Other research interests in VR such as how it affects human perception of space, cognitive responses, psychology, emotions, memory and aesthetics appreciation are not be covered.

Architectural visualisation

Architectural visualisation refers to the ability to visualise what architectural 3D models should look like in the past, in the present when it has already been built, or in the future when it is built. All visualised 3D models are architectural elements or kit of parts which are useful in the discipline itself. It could cover all the fields which include design, reconstruction, technology, history or detailing.

Graphics Rendering Pipeline

The graphics rendering pipeline is the engine that creates images from 3D scenes onto our displays. The pipeline goes through 3 stages, namely application, geometry and rasterizer, as shown in Figure 1.2. A 3D scene has geometries (triangles, lines and vertices), light sources to illuminate the model, material properties of the geometry, and the textures (literally images glued to the geometry). A virtual camera is needed to define the position, direction vector, vector up, field of view, near and far clipping plane to enable navigation around a 3D scene.

At the application stage, which is executed on the CPU, the programmer decides what happens in the 3D scene, such as collision detection, animation, rotation, movement and so on. The most important task is to send rendering primitives, which are triangles, to the graphics hardware. The geometry stage allows moving of objects by matrix multiplication, moving of camera by matrix multiplication, computing lighting at vertices of triangle, projecting onto screen (3D to 2D) and applying clipping planes setup to avoid triangles outside screen from being projected.

At the rasterizer stage, output is taken from the geometry stage and is turned into visible pixels on the display screen. Textures and various other per-pixel operations are added. The visibility issues will be resolved there at this stage by sorting the primitives in the z-direction so that only things which are visible are displayed. Advanced shaders or programmable shaders are becoming popular now by adding the ability to program vertex shader and pixel shader to both the geometry and rasterizer stages. These add more control and give the programmer many more possibilities for image output.

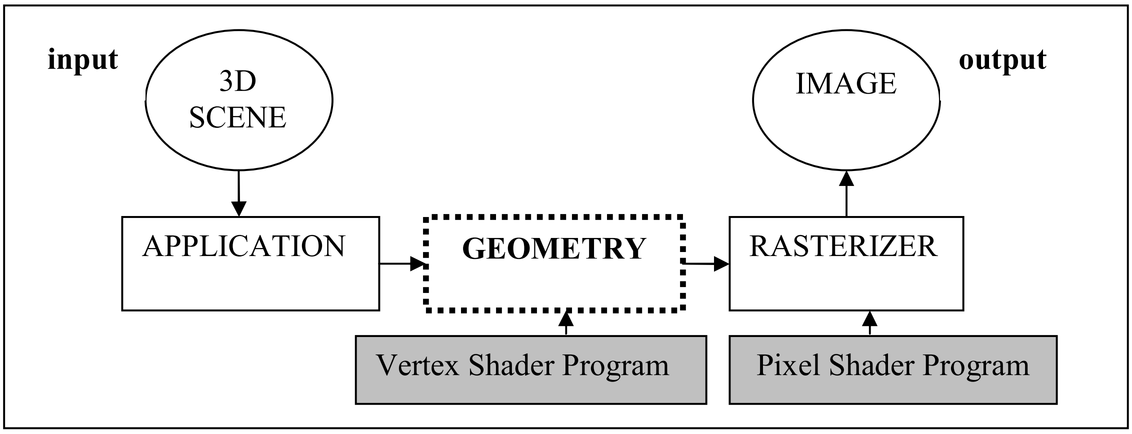


Figure 1.2. The Graphics Rendering Pipeline

Optimising the pipeline is critical in the computer world to determine the bottleneck. The stages in the pipeline are executed in parallel. Therefore, the slowest stage will become the bottleneck of the entire pipeline. As architecture students, we cannot optimise any of the specific bottlenecks created by hardware and software because this is not our expertise. However, we can optimise the geometry stage, where we have the control to input things into the scene. Therefore, this thesis specifically targets research on the optimisation of the geometry stage.

In conclusion, the research is interested in creating architectural visualisations using VR technology in real-time in the most efficient way possible. The process covers both the quantitative aspect, which is technical hardware performance, and the qualitative aspect, which is the humans' acceptance of the visual quality of 3D models based on their experience and exposure of how things should look in real life. Both of these aspects are crucial in achieving efficient real-time VR architectural visualisation.

1.3 Research Scope

From literature reviews, it is clear that research has been done to identify factors that affect real-time rendering (Af Klercker 2000, Wimmer and Wonka 2003). However, they do not identify which are the biggest contributors to degradation in frame rates and computer memory consumption, both of which will slow down the performance of a real-time walkthrough. Computer scientists are interested in optimising the algorithms as well as identifying the bottlenecks in the many stages of the graphics rendering pipeline to improve performance. Stages in the graphics rendering pipeline include the application, geometry and rasterizer stages.

As most architecture students do not possess programming skills, what can be done is to optimise the geometry stage of the graphics rendering pipeline (refer to 1.2.2 Graphics Rendering Pipeline p. 13) with the parameters we can control from the files we input to run the simulation. Therefore, the scope covered in the research is how to efficiently prepare real-time VR architectural visualisations by identifying the technical performance limits and human visual acceptance limits of 3D models quality. It specifically explores all the performance-affecting inputs we apply in the 3D scenes, and how to optimise them. It does not cover any cognitive or aesthetics aspects but focuses solely on the performance of VR visualisation within the quality acceptable by viewers.

For the technical portion of the research, computers ranging from workstations, desktops and laptops are used in the experiments because most architecture schools and firms cannot afford high-end systems. Hence, covering this range of computers is logical because the results can be applied by users of such systems. The initial tests are done to explore most of the common features used by architecture users of VR visualization software. The samples used are real complexity architectural 3D models or their parts. The final experiment will cover the four most fundamental aspects of a 3D model, namely triangle, vertex, geometry, and texture. It uses samples from architecture students' design projects because the scale and complexity will be similar to real architecture projects.

The four primary variables are also known as first generation variables. The research will not try to optimise the secondary variables (also known as second generation variables), which include particle systems, light count, programmable / advanced shaders count, scripting, collision detection, looped video and audio. Rather, it but will explore their impact independently against frame rate. This is because their usages in projects is optional and they only exist on top of the first generation variables. The belief of the research is that if the first generation variables are not efficient enough, no amount of efficiency of the second generation variables will be worth it. Efficient first generation variables have to be in place before second generation variables are applied on top of them.

For the visual portion of the research, the survey is conducted in a visualisation lab where there is a big screen with rear projection for stereoscopic visualisation. The tests are all within the boundaries of common 3D models used in architecture, textures resolution ranges commonly used in the field within the limits of current graphics technology and frame rate in the range of below and above real-time. Subjects who participated in the survey come from various disciplines because the audience who watches VR visualisation comprises people from all backgrounds.

1.4 Hypotheses

The hypotheses target the variables tested as well as the hardware and software used in VR architectural visualisation. The aim is to discover the major causes of slow performance during VR projects navigation. The findings will enable us to specifically target these causes so that the VR visualisation projects can be optimised smartly instead of arbitrarily without knowing how much it will impact the final performance.

The hypotheses are as follows:-

(i) Amongst all the variables of triangle, vertex, geometry and texture count, triangle count affects VR performance the most. This speculation is derived from observations of different VR projects created for presentation.

(ii) In a VR simulation, the time taken to navigate from one position to the other will increase when 3D models get larger. This speculation is derived from observations during navigations of VR projects.

(iii) VR performance will only be affected by hardware and not software. This speculation is derived from observations during preparation of VR projects on different hardware platforms such as desktops and workstations.

(iv) Each variable's weight of contribution will not be affected by hardware specifications. This is because influences remain consistent when different hardware platforms were used to run different projects.

The research will set out to prove all four hypotheses. The findings will help identify ways to achieve real-time VR architectural visualisation. Users will be able to use the guidelines generated from the research to speed up their process of creating VR projects. They will also be able to predict the frame rate in VR software before the 3D model is exported into the VR software.

1.5 Research Methodology and Organization

The research is divided into three stages – methods of optimisation, independent variable tests, and quantitative-qualitative aspects. There is a research design and method for all three stages. All stages contribute to the final aim of the research, which is to achieve efficient real-time VR architectural visualisation.

Stage 1: Methods of optimisation

Research Design: Experiment

Method: Data collection through computer display of the optimisation techniques available using software functions

Stage 2: Independent Variable Tests

Research Design: Experiment

Method: Data collection through computer display of samples of 3D models with their properties against frame rate measured individually

Stage 3A: Quantitative Aspect

Research Design: Experiment

Method: Data collection through computer display of 3D model samples with their four variables collectively measured against frame rate over different hardware platforms.

Stage 3B: Qualitative Aspect

Research Design: Survey

Method: Subjects from different disciplines within the university to view three different variables in six simulations and answer questionnaires.

In this research, the stages of modelling and texture-mapping are done in 3ds Max® while lighting and animation are done in 3ds Max® or EON^{TM} Studio^{TM} Professional^{TM} depending on the requirements of projects before visualising in the VR presentation by EON^{TM} Studio^{TM} Professional^{TM}. 3ds Max® is a visualisation software filled with features to create 3D still perspectives renderings as well as video animations. EON^{TM}

Studio™ Professional™ is the primary VR creation software used in this research. Game engines are not used because I lack computer programming background.

First, the thesis focuses on the literature reviews related to the possible fields of VR that are useful for this research. From there, optimisation techniques of VR presentations are discussed, drawing from literature reviews as well as similar methods provided from software used. The research will then cover all the possible variables independently in tests to determine their impact on VR simulation performance. Most of the samples used are parts of Mahaweli Building and Cloud Forest Biosphere. Both projects were selected because they are real world complex projects designed by Geoffrey Bawa and are thus more relevant to architectural visualisation. Next, the thesis discusses experiments conducted with the four basic fundamental variables which are present together in any architectural simulation. The 105 samples used for the experiment are Year 2 to Year 5 student design projects. These projects are done by architectural students for design exercises with the final output being still renderings and animations.

The hardware specifications used for the experiments consist of five different systems ranging from workstations to desktops and laptops. The first range of independent tests was done using the DELL™ Precision™ 650 workstation and the DELL™ Inspiron™ 9300 laptop. The final multiple variables experiment was done with all five systems which includes the DELL™ Precision™ 650 workstation, the DELL™ Precision™ 670 workstation, DELL™ Precision™ 380 desktop, DELL™ Inspiron™ 9300 laptop and DELL™ Inspiron™ 1520 laptop. All experiments were done in air-conditioned rooms.

The primary software used was EON^{TM} Studio^{TM} Professional^{TM} and the other used mostly for comparison was Quest3D®.

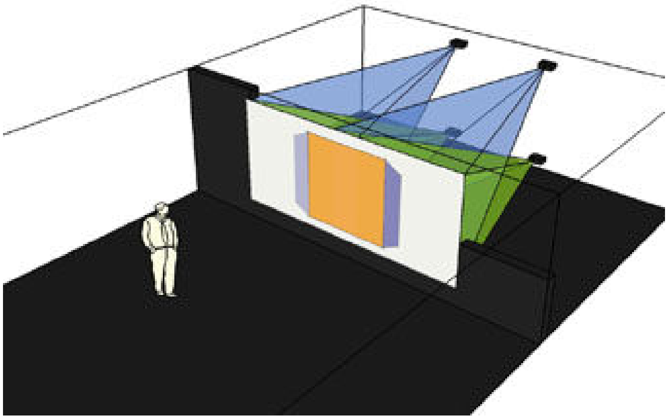


Figure 1.3. Visualisation Lab / Digital Space Lab Configuration
Source: Department of Architecture, NUS

The final VR architectural visualisation is presented in a visualisation lab as shown in Figure 1.3, which uses the 4.5m X 2.5m flat rear projection screen (horizontal 1280X2 pixels – 10% overlapping area and vertical 1040 pixels). It is projected by 4 bright high resolution Christie DS30 1280 X 1024 pixels SXGA DLP 3000 Lumen projectors via the Cyviz xpo.2 active to passive stereo 3D converters powered by 2 DELL^{TM} Precision^{TM} 650 workstation computers. These two computers are connected via a Gigabit network using the Master-Slave Model which is used often for VR visualisations by synchronization of rendering and events (Ryu *et. al.*, 2006). It is equipped with a 7.1 Dolby Surround THX sound system and uses of the mouse and keyboard as input devices. The viewers wear passive stereo glasses to get the stereoscopic effect of the visualisations.

The visual aspect portion of the research requires a survey to be conducted. Subjects who participated are students from different faculties within the university. The 3D models used are all done after the knowledge and experience obtained from optimisation techniques and independent variables tests. The survey was done in the same lab in stereoscopic mode. It was done to determine subjects' quality acceptance limit for triangle count, texture resolution and frame rate amount.

Finally, results of both the quantitative and qualitative portions are combined to determine how an efficient real-time VR architectural visualisation can be achieved..The findings will serve as a guideline for users who plan to build and present their designs in VR simulations. This guide will predict the expected frame rate count for the VR simulation using the value of each variable.

The thesis is organised in the following manner:-

Chapter one introduces the research problem, objectives, scope of research, hypotheses and methodology. Chapter two presents the findings of literature reviews which provide information, tips and guidelines on how to perform the research. Chapter three introduces optimisation techniques from literature reviews and observations as well as testing them out in application. The experiments conducted in this chapter will help us in choosing the most effective optimisation techniques to improve VR visualisation performance. Suitable techniques will be used for the preparation of VR visualisations and experiments in subsequent chapters. Chapter four explains the utilization of optimisation techniques to perform independent variable tests of each variable and the relationship against frame rate. The experiments done in this chapter is to verify each variable’s independent relationship with frame rate. Chapter five uses the basic four basic variables together to run against frame rate in an experiment using the simple multiple regression method on different hardware specifications. Unlike the previous chapter, the four fundamental variables chosen are tested at one go instead of separately. The other part is a survey conducted to study humans’ acceptance of three variables. Chapter six discusses the results from both the experiments and survey. Chapter seven summarizes the findings and gives a conclusion to the entire research. Finally, chapter eight discusses possible future research directions. The entire thesis framework is summarized in Figure 1.4.

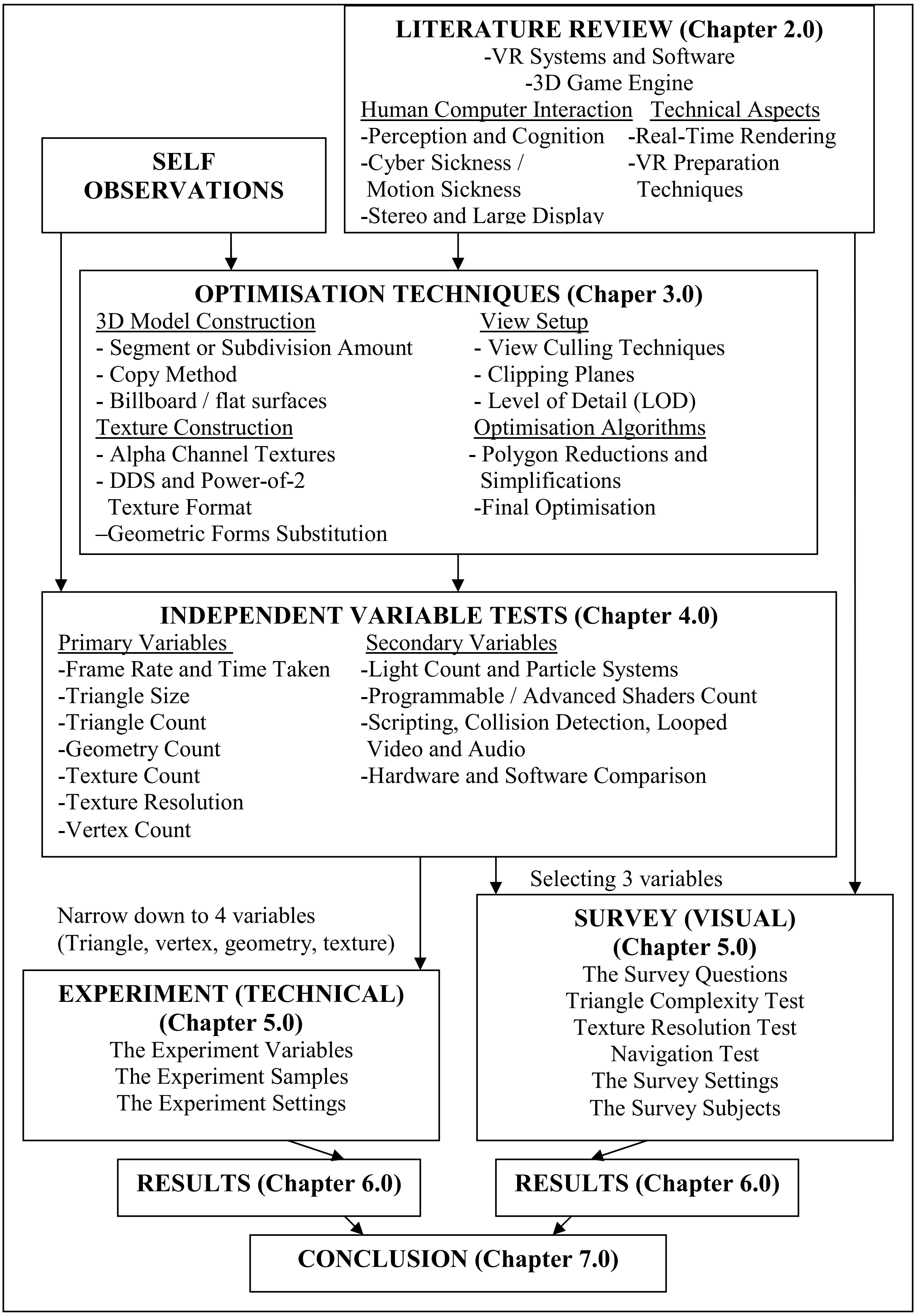


Figure 1.4. Thesis Framework

CHAPTER 2.0: LITERATURE REVIEW

This chapter reviews the literature used in the research by providing initially the historical accounts of visualisation, how it evolved to what it is today, the types of VR systems and software, as well as human-computer interaction and technical aspects. The human-computer interaction involves perception and cognition, cybersickness / motion sickness and stereo and large display. It explores the human response to VR visualisations and understands how they are affected physically, emotionally, psychologically and mentally. The technical aspects cover real-time and VR preparation techniques. It investigates the variables and factors involved in real-time VR visualisation. The fields of perception, cognition, cyber-sickness / motion sickness, and stereo and large display settings are covered, not because they are part of the research scope but because a knowledge of them will help in setting up the visual aspects for the survey portion of the research later. This chapter will serve as a foundation to the overall understanding of all aspects of VR. It will also focus the research in the area of interest by using findings by others as a guide. The literature review will greatly help towards the aim of achieving optimisation in the technical and visual aspects to improve performance.

2.1 History

Visualisation technology as how it is today did not come out of a vacuum. It is a result of continuous effort by men to find the best way to represent our real world with the mediums invented. From as early as the time when men drew on caves, beaches and tree barks to the invention of paper and computers, men have always wanted to draw realistic scenes of the world, objects and living beings. In architecture, representation of the real world is important because it will help us understand spaces before they are built, or if built, to understand it even without us being there at the real place. Finally, it will help us understand how destroyed or totally disappeared architectural ruins originally looked like. All these have value in all the related architectural disciplines of history, construction, M&E (mechanical & electrical) services, technology, structure, material, design and details. The interest in visualisation started with 2D drawings, drafting and perspectives on traditional mediums to current advanced mediums which can create still renderings, animations and real-time renderings.

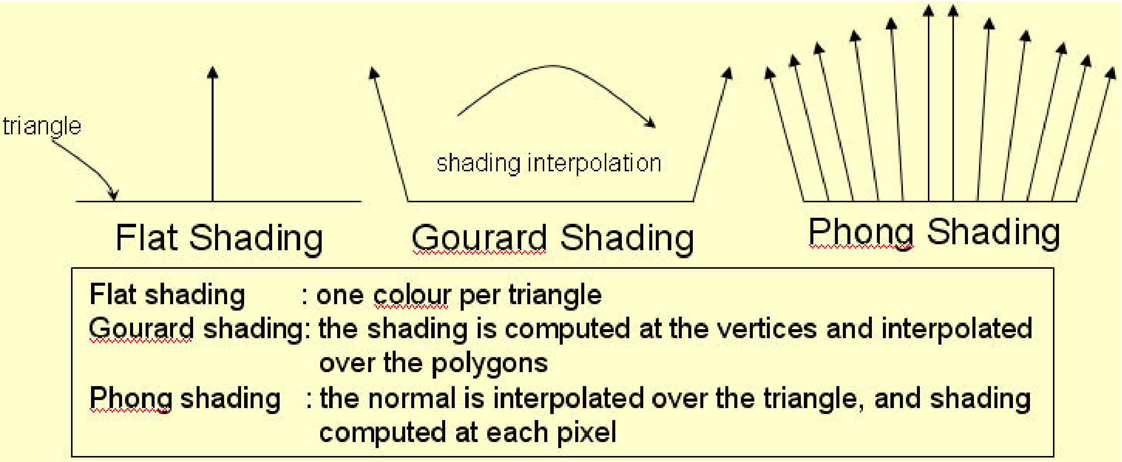
The Computing Era

Generally, it is thanks to flight simulation, urban warfare simulation and 3D gaming that VR visualisation has become more accessible and affordable to the masses. It is because of the research and advances done in these fields that we are able to benefit from the technologies used to achieve realistic VR presentation.

The first real-time computer is ‘Whirlwind’, which was under development at the Massachusetts Institute of Technology (MIT) since 1944. The machine has 1024 bytes X 2 banks of memory and weighed 10 tones. It aimed to respond instantly to whatever the user did at the console. It was part of Project SAGE, a programme to create a computer-based air-defence system against Soviet long-range bombers. It was first planned as a flight simulator and later evolved into the world's first real-time computer. Graphical interfaces were not very well-developed yet even though Cathode Ray Tube (CRT) displays with keyboards controls are available.

In the 1960s, the first Computer Aided Design (CAD) application called Sketchpad was developed. Then, there was the military-funded and upgraded Sketchpad III, which was the first 3D CAD application by Sutherland, a MIT computer scientist. This version enabled vector lines to be drawn on the computer screen with a light pen. The concept of an immersive 3D computer environment like the Head Mounted Display (HMD) was developed by Sutherland around 1965-1968. It was around the same time that interfaces and peripherals for human-computer interaction were developed. The mouse was first described by Englebart and colleagues. The touch sensitive glove was pioneered by Brooks who conducted research into haptic feedback.

Sutherland and his research students in Utah explored the rendering of 3D objects, which helped improve computer graphics greatly in the 1970s. Hidden line removal, colours, textures, lights and shading can be done to generate a rendering, which is the process of creating a final image from a set of geometrical data. Flat, Blinn, Phong, Lambertian and Gouraud shading were created one after another to improve the realistic look of objects generated on screen. Among them, Gouraud is used mostly in real-time VR while Phong shading is too costly to render. Flat shading is the most basic of them all, consuming the least processing power when used to render scenes but also the least appealing. The differences between traditional shaders are shown in Figure 2.1.



Flat shading : one colour per triangle
Gourard shading: the shading is computed at the vertices and interpolated over the polygons
Phong shading : the normal is interpolated over the triangle, and shading computed at each pixel

Figure 2.1. Traditional Shaders

Network computers became the next thing in research to get more processing power for complex 3D graphics. The first interactive architectural walkthrough system was developed at the University of North Carolina (UNC). The development of video projection methods, given the name 'artificial reality' in the mid-1970s was done by Myron Krueger. In the late 1980s, Jaron Lanier, the founder of VPL Research was credited as the person who coined the term "virtual reality". They pioneered development of hardware and software for VR systems. Interactive 3D became possible on the personal computer and applications were designed for low-end systems. During SIGGRAPH 1989, AutoDesk®, Inc. showcased their PC-based VR CAD system, Cyberspace. By the 1990s, the games market, especially computer games, continued to push developments on low-end systems. The first 3D first-person game, Wolfenstein 3D, was released in 1992. It was the period of Intel® 386 32-bit computers. Computer graphic cards were further developed at a faster pace, pushing low-end systems further with capabilities to update 3D scenes rapidly.

Later, CAD tools started to boom in the industry. Hence, the transfer of files between different CAD packages and between CAD and other design software became crucial. Early CAD tools could only do 2D drafting. By the 1990s, they were overtaken by those which enabled 3D design. From just drawing lines, CAD evolved to object-oriented CAD. This allows the manipulation of objects so you can do anything with it like moving, rotating and scaling. There was also the creation of parametric modelling, which involves the use of mathematical variables or parameters to control, modify or manipulate design. Many famous commercial CAD software began to hit the market from 2D drafting to 3D visualisations, which are used in all kinds of fields. They include AutoCAD®, Microstation®, Form Z®, Lightscape®, Revit®, ArchiCAD®, CATIA (Computer Aided Three Dimensional Interactive), Pro/E®, Maya®, 3ds Max®, Cinema4D, Lightwave 3D®, and SketchUp®.

In the 1990s, a VR hardware supplier called Fakespace introduced CAVE (CAVE Automatic Virtual Environment) and Immersadesk®, which enables large scale display of information as well as being inside the display system. On the software side, standard procedural models for 3D were developed based on the non-proprietary Open Graphics Library (OpenGL™). Open Inventor and Iris Performer, which provided further functionality, allowing the programmer to concentrate on world creation, were created by Silicon Graphics (SGI®), the VR hardware and software supplier. Microsoft® on the other hand, brought out a proprietary standard for Windows® called Direct3D™. At the end of the 1990s, most PC-based games and hardware developers based their programs on the Direct3D™'s DirectX®. Still, many others supported both and much later, nVidia® and Microsoft® jointly developed Cg (C for graphics). By the mid-1990s, the Virtual Reality Modelling Language (VRML) was developed based on the Open Inventor to provide visualisation of virtual worlds networked via the Internet. Soon, it became an internal standard called VRML 97 after further development. There was much competition later from Sun® Microsystems' Java™3D and Microsoft®'s abortive Fahrenheit. VRML97 was later updated to the Web 3D consortium's open standard X3D thanks to progress in 3D graphics.

By the mid 90s to 2000, mobile computing became possible. CAD to VR and GIS to VR data translation were possible through direct plug-in exports, VRML, 3DS and many other file formats supported by the VR software. On top of that, 3D game engines became available for users to import CAD into them as well. Since Doom™, game engine has been popularized for use in creating new games with models, scenery, texture and sound. Many game engines including CryENGINES2, Gamebryo Element™, RenderWare®, Source™ and Unreal Engine 3 became popular after that. These game engines were pushed to the limit because of extreme high-end 3D games like Quake™, Unreal Tournament, Doom™, Half-Life®, Counter-Strike™ and Oblivion™. They benefited the computer graphical world by introducing many realistic features which enhanced the experience of navigating in the 3D world. In the early 2000s, data input techniques have also been improving. We can import 3D laser scanning and geometry capture from images and film into CAD and VR software.

Displays also improved with volume visualisation, auto-stereoscopic displays and more portable and flexible screen technologies. In addition, the technology improved in terms of the quality of the monitor displays as well as the sizes. CRT, TFT LCD, plasma and HD are the standards that evolved through time with resolution standards of VGA, SVGA, XGA, SXGA+, UXGA, WXGA. Nowadays, displays are going into the wide ranges with resolutions of up to 1600 X 1200. Display sizes of computer monitors also increased from just 13" all the way to 43" currently. nVidia®, ATI™ and 3dfx have all contributed to the development of the graphics we have today with their famous models like RivaTnT, GeForce, Radeon and Voodoo. The graphics supplied to the display also improved from an integrated type to a PCI dedicated card to an AGPX4 card. Later, they went to a faster AGPX8 and to the current PCI Express cards and SLI cards. When they became a dedicated card, they had their own graphics processing unit (GPU) as well as video memory ranging from 4MB all the way to 1.5 GB (nVidia® Quadro® FX5600) nowadays.

Graphic cards also evolved to support more advanced and programmable shaders languages like Cg (C for Graphics), HLSL (High Level Shader Language) and GLSL (OpenGL™ Shader Language) which runs on both DirectX® and OpenGL™ Application Programming Interface (API). All the shaders are further divided into two types called vertex and pixel shaders. In real-time rendering, vertex shaders manipulate geometry vertices and their attributes while pixel shaders manipulate rendered pixels. The stereoscopic glass also evolved from the first generation red/blue glasses which had red and blue filters on the glass for the left and right eye. After that, the stereoscope began to appear. It enabled the users to view stereoscopic effects through it or to view directly with cross eyes. Finally, the current stereoscopic glasses are with polarized light, with vertical and horizontal filters on the glasses for the left and right eye (Faustner, L. 2007).

It is beginning to be possible for shadows to be generated real-time as volume hard and volume soft shadows and the amount of real-time lights that can be generated in real-time has increased too to around 8. New file formats like the High Dynamic Range Image (HDRI) enables more realistic rendering of spaces according to real world light quality. DirectDraw Surfaces (DDS) was created first with DirectX® 7.0 and later in OpenGL™ to efficiently generate textures in real-time visualisations using mipmap levels. At the moment, graphic cards can support up to 4096 pixels of texture images running in real-time too. Progress is ongoing to find better techniques to achieve more realistic graphics for better quality visualisations especially with the flourishing gaming market of the current generation of 128bit consoles of Nintendo™ Wii™, Microsoft® Xbox 360™ and Sony® Playstation®3.

In the entertainment industry, VR was already used as early as 1922 in the movie ‘The Power of Love’. In the 1980s, IMAX (large format-sideways running, 70mm) started offering many 3D films, among them ‘Jaws 3-D’ in 1983 was the most renown. In the music industry, the band U2 produced the first 3D concert idea from their seven different concerts of the Vertigo Tour in South America in 2006. ‘U23D’ (U2 2008) was made possible by 3ality Digital LLC. (3ality Digital LLC. 2008) using nine different cameras.

As we are currently moving into the interactive multimedia age, VR and mixed reality (MR) / augmented reality (AR) are going to be used more and integrated into multiple fields and industries. Serious Games Institute (SGI) based in Coventry University's Technology Park is one example which incorporates virtual worlds into different industries like medical training, military simulations, business enterprise modelling, primary and secondary education, tourism and cultural heritage, location independent working and virtual conferencing. (Serious Games Institute 2008)

Our lives are already slowly being changed by the virtual worlds with online communities created. Among the famous few are Second Life by Linden Research, Inc., Google Earth by Google, OLIVE^{TM} (On-Line Interactive Virtual Environment) by Forterra Systems Inc. and World of Warcraft (WoW) by Blizzard Entertainment. The real world is merging with the virtual world in many ways. One of the projects available to relate them is the Metaverse Roadmap (MVR) by Acceleration Studies Foundation (ASF) and its supporting foresight partners to explore the virtual and 3D future of the World Wide Web (Metaverse Roadmap 2008).

2.2 Types of VR Systems

There are many types of VR systems which differ depending on the economic values and scale. Each of them has their own advantages and disadvantages and they fulfil needs according to different research interests. If more than a computer is used for the system, synchronization methods are used to make sure the merged parts of the viewport move together during simulation. It can be done with addition hardware or with software instead (Koike *et. al.*, 2005).

(i) Auto-Stereoscopic display screens

The holographic stereo screen does not require stereoscopic glasses to be worn by the users. More advanced screens enable touch activated capabilities on screen and do not confine a user to a certain distance and position to be able to view the 3D stereoscopic depth effects.

(ii) Large flat wall

One or two computers generate the VR scene. They are connected to projectors behind the screen. The projectors will separate the images projected on screen and the users are required to wear stereoscopic glasses to visualise the effect. The advantage of this system is the ability to accommodate more users at a time and has better definition displays.

(iii) Large curved wall

Normally consisting of three computers or more, these projections from the back of the curved screen which surrounds the user give the effect of immersion in space. This system can also accommodate a big audience and requires stereoscopic glasses to display.

(iv) Large Flat Multiwall (CAVE – Cave Automatic Virtual Environment)

Four, five or six walls of a room-sized cube space where users can visualise the spaces inside. Projectors will project from outside the cube space onto each of the surfaces. The downside is the limitations of the user to view the space at a time. One of the most advanced systems to date is a 8 projectors X 6 walls system which has a grand total of 48 computers projecting through 48 projectors. This solves the problem of low resolution of each wall panel of the CAVE.

(v) Immersive Reality System

Head mounted display (HMD) that allows individual users to immerse themselves fully in the virtual environment through 3D imagery involving sight and motion. The user can navigate in the environment simply by hand and head movements. There will be gloves or camera sensors to detect users' movements. The advanced sense of touch can be visualised too. The advantage of this system is that it is mobile and does not require a lot of space to set up. The disadvantage is that it is for individuals. In the future, perhaps networked HMDs can allow multiple users in the same environment to see each other as avatars or realistic representations of the real person.

(vi) Augmented Reality System

(rephrase – this is VR that superimposes the real world with imagined images projected on goggles worn on the head.) This is a system which enables mobile users to interact with the real world. The difficult task at hand is to enable wireless connections so that the user can benefit from the use of this system to roam around a huge space, such as a city, and still be fed with information about the space they are in. Currently, it is more possible to use augmented VR in small spaces, such as inside a museum space.

(vii) Hologram

This system uses huge 50” wide screens with High Definition TV and PC inputs, which are projected from below so that users can sit around it and visualise 3D images in midair. It is more effective to show objects than spaces as users are looking at the thing from the outside.

In another related research in holograms, an updatable holographic 3D display has been developed at College of Optical Sciences, University of Arizona, Tucson, AZ (Greenemeier L. 2008). It is based on photorefractive polymers capable of recording and displaying new images every few minutes. This is the largest photorefractive 3D display to date (4 times 4 inches in size); it can be recorded within a few minutes, viewed for several hours without the need for refreshing, and can be completely erased (Flavelle C. 2008) updated with new images when desired.

There are companies which provide the service of producing VR and mixed reality (MR) / augmented reality (AR) visualisations. They normally have the whole package of hardware and software solutions to cater to the different needs and budgets of their various clients. They include:

- EON Reality, Inc. (http://www.eonreality.com)

- Forterra Systems Inc. (http://www.forterrainc.com)
- Inition (http://www.inition.co.uk)
- Mechdyne Corporation (http://www.mechdyne.com/index.htm)
- RTT AG (http://www.rtt.ag/cms/en)

2.3 Types of VR Software

There are many types of VR software available in the market. Common commercial ones include:

(i) Act-3D^{TM} Quest3D®

(ii) Blaxxun^{TM} Place Builder

(iii) Cubicspace® rtre^{TM}

(iv) Cycore System AB Cult3D

(v) Dassault Systèmes Virtools^{TM}

(vi) DX Studio^{TM}

(vii) Electric Rain Swift3D®

(viii) EONReality^{TM} EON^{TM} Studio^{TM} Professional^{TM}

(ix) NavisWorks® JetStream

(x) QuadriSpace^{TM} Publisher3D^{TM}

(xi) TurnTool^{TM} TurnToolBox, TurnToolViewer

(xii) Wildtagent^{TM}

(xiii) Skyline® TerraExplorer® Pro^{TM}

(xiv) VR4MAX®

(xv) Octaga Octagon Player

(xvi) Presagis Engenuity, MultiGen-Paradigm, TERREX, S-Mission

Some of them are meant for web 3D publishing, which include Demicron Wirefusion® and Pulse3D Pulse Mobile™.

For an architecture school which does not provide computer programming lessons, most students would lack such a background. Therefore, the ease of using commercial software for exporting 3D models to achieve VR simulation presentations would be critical. From the list, EONReality™ EON™ Studio™ Professional™ is the primary VR software used in this research. Act-3D™ Quest3D® is also used for some comparisons. Both software have enough tools to provide the functions required in the field of architecture to produce simulations of various interests which include historical, structures, artificial lighting, sun studies, reconstruction and detailing.

2.4 3D Game Engines

Game engines have played a big role in influencing hardware and software development so as to achieve good 3D visualisation. If not for the great progress done on game engines, research into VR technology would still be backward. It also caused hardware to become more economical because of the fast upgrades.

A game engine is the software component of a computer game, console game or interactive application with real-time graphics. 3D real-time rendering in computer games started in 1984 with the 8-bit computer, ZX Spectrum with Zig-Zag game and later on 16-bit computer, Amiga and the Atari. The first game to use textures was id Software's Wolfenstein 3D in 1992, programmed mostly by John Carmack (Valient 2004). The objects were painted with the billboard method so it would not be too complicated for computers back then to run.

3D game engines can run on multiple computer platforms including Linux, Mac and Microsoft® Windows® to the latest game consoles of Nintendo Wii, Sony Playstation 3 and Microsoft® Xbox360. In the case of VR, 3D game engines are referred to specifically. These game engines are used as the final stage of game creation and have features of a rendering engine, 3D graphics, physics engine, collision detection, 3D sound, scripting, animation, artificial intelligence, streaming, memory management, a scene graph and networking. Most 3D game engines are built upon a graphics API such as Direct3D™ or OpenGL™ while the rest will have their own creations.

Game engines used in the market are as follows:-

(i) Unreal Engine

(ii) NetImmerse

(iii) Oblivion™

(iv) Half-Life®

(v) RenderWare®

(vi) Gamebryo Element™

(vii) DX Studio™

(viii) ChaosEngine

(ix) 3DGameStudio

(x) DarkBASIC Professional

(xi) LawMaker

(xii) Garagegames®Torque Game Engine

(xiii) TV3D SDK 6

(xiv) C4 Engine

(xv) Unity

(xvi) 3Impact

(xvii) Deep Creator

(xviii) Source™ (Half-Life® 2: Episode 1, Counter-Strike™: Source, Lost Coast, Day of Defeat: Source)

(xix) Axiom

(xx) Ogre

There have been architectural and urban visualisations done using game engines such as Quake™ (So-Yeon and Tutar, 2004), Half-Life® and Oblivion™ (Digital Urban, UCL 2006). It is always important to know how commercial software stand against game engines (So-Yeon and Tutar 2004). It is comparing advantages and disadvantages in representing a building with the VR functions between the Quake™ III game engine prepared in Discreet GMAX and EONReality™ Studio™ prepared with 3ds Max® / Maya®.

As a result of the comparison tests, Quake^{TM} III was found to have more transparent interfaces to use, and has better environmental graphic and sound effects. It gives closer to real life experience in environments with realistic movements and many changeable views. However, it has no human scale eye level as it was designed for giant figures in virtual battlefields. The aim of game engines is to produce realistic spatial experience and therefore, they are good for architectural representation tools.

On the other hand, EON^{TM} Studio^{TM} allows more diverse interactive features, and provides more design alternatives with many high-level interactive options. It is easy to convert to VR environment formats such as CAVE and HMD (head mounted display) or stereoscopic displays to web publishing. It does not require as high a level of programming skill as game engines because the most one needs to do is some Java^{TM} and Visual Basic scripting. This gives the commercial software an edge for use in architecture schools where programming skills are scarce.

Another study was done to compare between several methods using game engines and VRML file format through an internet browser to visualise the ancient Maya city of Calakmul (Mexico) (Ruiz-Rodrate *et. al.*, 2005). The model was initially built in AutoCAD with topographic measurements and GPS data. Texture mapping is done in 3ds Max® before exporting to internet browsers and game engines. The project uses Unreal, Unreal Tournament and Nebula game engines while the web version uses the common VRML format to run on internet browsers.

The Nebula game engine can take 3ds Max® 3D models using a third-party plug-in converter. It only takes bicubical images (base 2 dimensions). C++ programming is required to build interactivity into the virtual environment. As for Unreal, the ASE format needs to be exported out from 3ds Max® with images of only PCX or TIF format accepted. Character height needs to be enlarged 65 to 1 in the game engine. Unreal's environment is very realistic as it provides torches, moving water and grass as some features to use. Even without programming background, it can still be used.

The VRML file format is required to run on internet browsers for the web version visualisation. Optimisation of the performance of real-time rendering by any internet connection is done by changing common 3D geometries with texture details. Buildings are plain cubes mapped with photographs from the real buildings they represent. Therefore, it requires more photograph retouching than with the model construction. The other method is to divide the whole scene into separate sections which are called by the inline VRML node as the user approaches certain points. This study shows that some modifications in programming are required when using game engines while optimisations are needed to run in internet browsers, taking into account the various internet navigation speeds of all possible users. It does give a good account of how to start if users are interested in using game engines.

In the end, we have excluded using game engines in our experiments as they are meant for game developers to develop 3D games and the architecture community has limited access to them. To use them also requires programming skills which most architecture students lack. There are also no game engines in the school to test on, so they are totally neglected in the research.

2.5 Human Computer Interaction

2.5.1 Perception and Cognition

This topic will help in the preparation of VR visualisation projects, especially for the virtual acceptance survey later. It is crucial to understand how humans react to VR visualisations so that the simulations will effectively achieve their aims when showcased to the audience. It will be a waste if all the advantages that VR can provide are not utilized and the disadvantages ruin the presentation in the end. There are still many promising areas for research with regards to the human factors involved in the perception of the VR visualisation as it is still in its infancy stage (Tory, M., and Moller, T. 2004).

Related research done to understand human preferences includes comparing between just using colours and applying materials to surfaces in 3D scenes. Colour was found to have a weaker impact on human perception than material (Chiu-Shui 2007). Others look into understanding human behaviours and strategies of wayfinding in VR (Darken and Sibert 1996). It was found that when there are no sources of directional cues, disorientation will occur both in wayfinding performance and spatial knowledge acquisition. Subjects were uncomfortable with completely free and unrestricted movement. Another interesting finding is that path-following is a natural spatial behaviour where coastlines or grid lines are used by subjects as if they were paths. If a map is given, it will optimise searching strategies by supplementing it.

Feeling of presence in VR simulation is also attracting research interest. This is normally related to our senses as cumulative effects (Dinh et. al., 1999) while others are interested in looking at the effective orchestrations of these multiple sensory stimuli to further the bounds of virtual realism (Morie et. al., 2003). These sensorial responses can be enhanced by more interactivity that will improve the user's engaging experience. The strength of interactivity is studied as well by using two systems, Boom Chameleon used for evaluating virtual models, and StyleCam for developing online product marketing and advertising as they have real-time response and provides interactive narrative with game-like experience (Tsang *et. al*., 2003). They adapt and use elements of video and computer gaming to create great user experience in virtual design and product marketing. The use of game concepts in non-game application domains is effective in improving interactivity.

The feeling of presence will determine how realistic a virtual environment is (Chiu-Shui and Chien-Hui 2005). If there are more sensory inputs and more activities added, the feeling of 'being there', which is the sense of presence will improve. However, more activities may cause motion sickness. It is suggested that the sensory stimuli of sight, sound and touch, with more cognition occurring in the experience improves the overall sense of presence. A study was done and it was found that the perceived level of presence correlated positively with feelings of warmth, comfort, simplicity, uniformity and spacious space for the monocular conditions displayed on the HMD (Mania and Chalmers 2001). Others go one step further by looking at the effect of rendering quality on subjective impressions of illumination and perceived presence after exposure to a virtual environment (Mania and Robinson 2004).

From the literature review, it is very clear that interactivity and our senses play a major role in helping the VR visualisation to be more realistic to the audience. The findings in this category will help plan the creation of our projects in a more meaningful way so that the advantages of this technology is maximized. Many findings are helpful, for example choosing between colour or textures in scenes and how crucial tools like maps are to help navigation.

2.5.2 Cyber Sickness / Motion Sickness

Cyber sickness or motion sickness is a serious issue that needs to be discussed because VR technology will be rendered useless if it occurs frequently during presentation. For whatever research in the VR field, this aspect has to be looked into seriously because if a VR presentation causes cyber sickness, all the advantages it can provide will fail. It defeats the purpose of visualisation if people cannot visualise things the way they are supposed to be. Cyber sickness can be caused by hardware setup and software to issues relating to the individual such as age, background, gender characteristics (Czerwinski *et. al.*, 2002) and 3D graphics experience. It is considered a crucial issue to resolve so that VR visualisations created are viewable for humans. It will be useless and a total disaster to create the best quality projects but which are totally uncomfortable, causing nausea and dizziness in people who are viewing them. At the end of the day, all the advantages of a VR presentation will amount to nothing and all the hard work put in to create the simulation will be meaningless if people cannot digest it well.

Cyber sickness or motion sickness caused by viewing VR simulations is discussed in a research to understand it in detail (LaViola 2000). The symptoms of the sickness include eye strain, headache, pallor, sweating, dryness of mouth, fullness of stomach, disorientation, vertigo, ataxia, nausea and vomiting. The cyber sickness theories available include the sensory conflict theory, the poison theory and the postural instability theory. All of them are not perfect and there are situations where a theory does not hold. They also cannot explain why in identical conditions, not everyone gets sick. A unified theory is still being sought for.

The causes of cyber sickness are divided into display and technology issues as well as individual issues. Position tracking error, lag and flicker are identified as the display and technology factors. Individual factors include gender, age, illness and position in the simulator. It is claimed that women appear to be more susceptible to cyber sickness than men because one of the reasons is that they have wider fields of view. People between the ages of 2 to 12 years have the greatest chance of getting cyber sickness. It greatly decreases from 12 to 21 years and more thereafter.

Cyber sickness is claimed to be nonexistent for people of around 50 years of age. People who are suffering from illness, fatigue, sleep loss, hangover, upset stomach, periods of emotional stress, head colds, flu, ear infection or upper respiratory illness will get sick more easily. Stting during the simulation is a better position than standing. Users who control the simulation are less susceptible to cyber sickness than passive participants who just watch the simulation. Cyber sickness reduction methods are proposed but nothing is claimed to be able to solve the sickness completely. These methods include motion platforms, direct vestibular stimulation and rest frames. For the purposes of this research, an understanding of the reasons that cause cyber sickness will help in the understanding of how humans will react to VR simulation depending on their characteristics and systems used.

A possible procedure to reduce simulator sickness and virtual environment sickness is proposed (Duh *et. al.*, 2001). The procedure involves using an "independent visual background" (IVB) to examine the difference in postural disturbance evoked by visual scene motion at different frequencies.

The results clearly show that IVB reduces balance disturbance and therefore can be used in conditions where conflicting visual and inertial cues are likely to result in simulator sickness. At high frequencies (0.8 Hz), there was little balance dispersion between IVB and without IVB conditions. At low frequency (0.05 Hz) scene motion, there is greater balance disturbance. An IVB will lower the balance disturbance. The bright IVB condition will also improve the balance performance of subjects as well as their ability to follow instructions of focusing on the scene or IVB during the experiment. A potential problem seen with using the IVB is that it will lower the sense of presence by taking away too much attention. Explorations are done to make the IVB unnoticeable or even sub-threshold.

There is another research which uses the same method but added a new procedure called the Virtual Guiding Avatar (VGA). It combines self-motion prediction cues and an independent visual background (IVB) to alleviate simulator sickness (Lin et. al., 2004).

The effects of the VGA were examined using the Revised Simulator Sickness Questionnaire and The Enjoyment Engagement and Immersion Questionnaire. The results showed that simulator sickness was significantly reduced either by a VGA that was earth-fixed and coupled with rotational prediction cues or by a non-earth fixed VGA that provided rotational and translational prediction cues. VGA also enriched the positive aspects of user experiences as they reported more presence and enjoyment. The ability to predict upcoming motion because of the motion prediction cues which permit subjects to maintain gaze at the centre of radial visual flow, in the direction of the car's heading significantly reduced simulator sickness as well.

Another study on cyber sickness was done in 450 visitors to a VR lab in Iowa State University (Knight and Arns 2006). They were invited to experience an immersive virtual environment in a 4-wall projection VR system. The survey looked into their experience playing video games, experience with VR, general susceptibility to motion sickness and experience of level of presence on a scale of 5. Background questionnaires about age and gender as well as the Simulator Sickness Questionnaire (SSQ) (Kennedy, *et. al.*, 1993) were used during the survey.

In the study, 18.9% reported no sickness while the majority of them reported low post-test sickness values. Total sickness severity scores show that it will decrease as level of game play and presence increase,. On the other hand, total sickness will increase as self-reported susceptibility and age increase. The score remains constant across genders. The research findings, especially on subjects' profiles, will help much in understanding the number of human factors that will affect simulation results.

Another important aspect of cyber sickness is to understand the effects of field-of-view, display type and user role on the experience of simulator sickness and presence in users when viewing VR simulations (Seay *et. al.*, 2002). One hundred and fifty-six undergraduate students comprising 133 males and 23 females between the ages of 17 and 38 participated in the study. Assessments were done using the Presence (PQ), Immersive Tendencies (ITQ) and Simulator Sickness Questionnaire (SSQ) (Kennedy, *et. al.*, 1993). The NAVE (Non-expensive Automatic Virtual Environment), a rear-projected display system was used for the experiment. It has three 8' X 6' screens with two side screens positioned at 120 degree angles to the central screen. The central screen is 16' wide and approximately 7' deep.

The experiment was conducted by dividing the subjects into groups to assess three independent variables – field of view [60 degrees (central screen) vs. 180 degrees (all three screens)], display type (stereoscopic vs. monoscopic) and user role (driver vs. passenger). The results show that the drivers feel more immersed and present than passengers. Passengers' sense of presence is not enhanced by the passive nature of their role. Participants experiencing the simulation with a high field-of-view reported higher feelings of presence than those in the low field-of-view conditions. In the low field-of-view case, the two blank side screens remind users that they are in a simulator.

However, those in the high field-of-view conditions reported higher levels of nausea than those in the lower field-of-view conditions. Passengers in high field-of-view conditions reported the highest nausea ratings. The stereoscopic mode added to a low field-of-view simulation creates the same level of disorientation as a high field-of-view monoscopic mode. Overall, field-of-view seems to be the biggest factor in determining the experiences of sickness and presence during simulations. A large field-of-view is a double-edged sword as the participants feel immersed, but would like to leave as soon as possible.

The literature review in this category helps further the understanding of cyber sickness and the ways to reduce it. Impacts of field-of-view, stereoscopic or non-stereoscopic displays, and roles of driver or passenger on the degree of cyber sickness are also important. The knowledge gained is crucial for the research, especially in the qualitative aspect later where the survey deals with human subjects directly. Efforts have to be in place to understand how cyber sickness happens and how to minimize its impact on the audience.

2.5.3 Stereo and Large Display

An understanding of the capabilities and advantages of stereo and large displays will help in the survey part of this research. It will provide an understanding of the benefits of having large and stereo displays and the settings can be applied in all the VR projects. A large display is essential to project 3D models on an approximately 1:1 scale, so that they look more realistic. It is expected also to improve task performance. Stereoscopic display will add depth to the flat displays. This will give a sense of distance between objects in the scene and is expected to increase realism as well. Both are critical in making the final visualisation as real as navigating in the real world.

For viewing VR in stereoscopic mode, research has been done to determine the different responses between two groups of human subjects to visual factors of illumination direction, light source numbers and view point position (Lo and Chalmers 2003). Human judgement of the visual realism of computer-generated images is investigated. Subjects are to decide whether the visual stimulus given to them is real and how much time they take to respond. One group views in stereo while the other does not.

The results showed that subjects involved in stereo-viewing took more time to respond than those who were not. This holds true for responses in all three experiments examining different illumination directions, the light source numbers from one to five as well as view point variances at 15, 30, 45, 60, 75 and 90 degrees. At least 20 seconds was required to decide the realism of a scene when in stereo. This means that people will not be able to determine much detail when viewing dynamic scenes in stereo. Therefore, more processing power can be saved from using more realistic qualities in the VR scenes like not using shadows and still maintaining the same perceptual response. The paper is crucial in understanding that human beings will require more time to adjust to stereo vision. Therefore, in the proposed survey, participants should be allowed more time to view each simulation choice before moving to another.

The impact of large displays is always important in VR to understand how it affects tasks and users' characteristics. Some tasks studied include objects manipulation and navigation in the environment and visual attention abilities of users (Tyndiuk *et. al.*, 2004). Forty subjects go through the experiment viewing in a large wall display with video projection system (3.3m X 2.8m) and a desktop monitor (0.23m X 0.30m). They are asked to sit at 5.5m in front of the large display as well as 0.5m in front of the desktop monitor.

The results showed that performances of users were better whatever the tasks involved in the large display. In the large display, the interaction showed performances gain for the slower subjects while it remained the same for faster subjects. For visual attention abilities test, the performances improved for some users when viewing with large displays. This paper's results are crucial in justifying the use of big displays for VR visualisation as it will be able to present architectural designs in the best possible way.

A similar research looking into increased display size and resolution in task performance improvement in virtual environments was done (Bowman and Chen 2006). The interest was in how users work with various displays when using a wayfind aid to help spatial information acquisition and mental map construction. Thirty-two subjects from intermediate to experienced video gaming backgrounds participated in the experiment. The experiment used two levels of resolution, at 1280 X 720 and 2560 X 1440 pixels in two display sizes of 48.0" X 27" and 18.8" X 10.6". The distance of viewing is 24", so for the large display, a 90 degrees horizontal field-of-view can be achieved while 42.8 degrees can be achieved for the small display.

The results showed that increased display size and resolution improved subjects' performance for all the varieties of navigation, search and comparison tasks. Large displays do help in facilitating both spatial navigation and information gathering. In the comparison of resolution test, even a large rear-projected screen outperformed a regular-sized monitor with a higher resolution. Therefore, size matters more than resolution if one must choose between the two. Wayfinding aid, even in the small display, did help improve navigation performance significantly. Users did move their heads to cover the big displays, leaning forward in some cases. Users felt more present when experiencing large displays, especially when they were seated close to the screen. The results reinforced the findings of the previous paper to choose large displays for VR visualisation but high resolution is optional if economic reasons do not permit it. It also shows that the distance of the audience to the screen should be sufficiently short.

Knowing the pros and cons of stereoscopic visualisation and big displays from the literature review in this category will help greatly in preparing VR projects as well as obtaining the correct settings to show the audience. It is crucial to know the impact of display resolutions and how different displays affect tasks.

2.6 Technical Aspects

2.6.1 Real-Time Rendering

Real-time is defined as maintaining a continuous frame exchange process which is fast enough so as not to be detected by our human eyes. It imitates navigation in the real world. If it lags, which causes ghosting and jerks (Helman 1995), the delay will defeat the purpose of a VR visualisation. Demands for bigger and more complicated models are always present but the hardware is always not good enough to fully meet the demands. Therefore, in order to maintain the quality of real-time, many things need to be done at many levels to sustain it.

Research was done to examine the problem of estimating the rendering time for a real-time simulation (Wimmer and Wonka 2003). The different factors that contribute to rendering time were studied and several algorithms that could give reasonable upper limits for the rendering time on consumer hardware were proposed. The four components (tasks) that were looked into in their rendering time estimation function were system tasks, CPU tasks, idle time and GPU tasks. Then, comparisons of several mathematical heuristics for the rendering rime estimation were done. New rendering time estimation heuristics were also proposed. During these experiments, a few variables which included triangles, objects or geometries, vertices and textures count were handled. These variables are selected and tested in the independent variable tests in the research later.

Current hardware systems are limited by the semiconductor material as well as the system architecture. These have limited the ability to perform real-time renderings which is why the interest of this research is to find ways to efficiently run real-time renderings on the current available hardware systems. There are proposals on future real-time rendering systems which possess some great features (Mark and Fussell 2005) like having ray tracing as the visibility algorithm of choice. Some of these features are already being used in game consoles, such as integrating scene management with rendering and thus having a parallel hardware architecture. It is also believed that a unified system of MIMD multithreaded, multicored machine with cache-coherent shared memory will be commonly used in the future.

This literature review will help in constructing the hypotheses as well as looking into the core variables that a 3D model is constructed of. Understanding the limits of hardware here has opened up investigation opportunities on the current hardware systems available in the market which are commonly used for VR visualisation.

2.6.2 VR Preparation Techniques

There are many preparation techniques of creating a 3D model for VR visualisation, starting at the modelling process. All the research done under this category will greatly help to kick start the investigation of optimisation techniques in the next chapter.

The modelling process for VR is different from modelling for still perspective renderings and animation. Dealing with 3D models requires a different approach as all computer processing power and memory consumption cannot be spent when running VR simulations. Therefore, all the techniques must be used beforehand to make sure that the final presentation will not be too slow for visualisation.

A research is done to show a report from a surveying course called “Modelling for VR in Architecture”, which is given during the spring semester at the CAAD (Computer Aided Architectural Design) division of TU-Lund (Klercker 2000). Practising architects with experience in using object modelling CAAD were the students for the course. The aims were to survey different ways of using available hard- and software to create VR-models of pieces of architecture and evaluate them in desktop and CAVE environments. The architect was to do as much preparation work as possible with his CAAD program and only the final adjustments were done with the special VR tool.

The few things done by the students were as follows:-
(i) simplifying the geometry by minimizing the amount of polygons

(ii) using textures to substitute geometric forms

(iii) texture type is square with the side size with the multiple factor of 2

(iv) adjust them to VR formats

(v) LOD (Level of Detail)

(vi) behaviours – opening doors, moving objects

(vii) camera with headlight

(viii) move your “eyes” around and manoeuvre in a naturalistic way

All these findings are important in helping to understand the troubles and techniques that are needed to go through as well as use to model for VR visualisation. Another paper which used almost the same techniques explored how performance of the Medieval Norwich model could be improved. The Charismatic project, which was combined with the 401 by 401 height map to produce a new model comprising 544,002 polygons, (Willmott *et. al.*, 2001) made full use of four techniques of optimisation. The researchers used View Frustum culling, Occlusion culling, a view-dependent level of detail algorithm for large terrain data sets called Real-time Optionally Adapting Meshes (ROAM) and their own specialized house level of detail which helped push their frame rate from just 1.60fps to 35.54fps.

As can be seen, LOD and culling are some of the most famous techniques of optimising models so that the entire 3D model is not modelled out, especially when you cannot even see those surfaces. Most VR software have their own LOD and culling techniques built into them while others create algorithms to improve it, whether it is a LOD algorithm or a

View Frustum culling algorithm or an occlusion culling algorithm (Coorg and Teller 1997).

Texture mapping is another important aspect of VR visualisation that is important to be looked into for optimisation. It is even crucial when dealing with huge scenes. Another LOD technique called Continuous level of detail (CLOD) is integrated with a texture representation called Texture Mipmap Quadtree (TMQ) to represent large-scale textures by a research to map large-scale terrain (Hua *et. al.*, (2004). The research focuses on using the TMQ and dynamic texture management to trade-off between the quality and efficiency of texture mapping using their algorithm. It is also in the hope that texture compression can be accelerated using image processing and texture synthesis algorithms. There is another research done to use images to replace distance 3D models (Aliaga and Lastra 1999). They present a pre-processing algorithm and run-time system for rendering 3D geometric models at a guaranteed frame rate. Therefore, higher frame rate can be achieved by reducing the maximum number of primitives to render. The research will look at some of the optimisation techniques mentioned above to help improve performance of VR visualisation so that they can be used in the 3D models preparation process to the final stages of VR presentation.

CHAPTER 3.0: OPTIMISATION TECHNIQUES

Continuing from the previous chapter on literature review, some techniques of optimisation are identified and explored in this chapter. Some of the techniques are available as functions of the CAD and VR software. There are many traditional techniques of optimisation which can help to efficiently prepare 3D models for VR presentation. These methods are also useful for 2D drafting, 3D model construction for rendering and animation. All these methods will help speed up the process of completing all the tasks mentioned above. Their use is crucial and even compulsory in order to produce the final presentation smartly and efficiently. All the methods are used in all experiments and survey models wherever applicable in the research. Some of the most common ones include using texture maps to reduce geometric detail, using primitive solids, using distance dependent levels of details (LODs), using billboards and selectively loading objects (Whyte 2002).

This chapter is divided into four sections where optimisation can be executed. The first section is during model construction, the second is texture construction, the third is during view setup and finally the last is optimisation algorithms. All these stages are different processes that need to be gone through before getting a project ready in VR visualisation presentations. This is to ensure that whatever is prepared for the visualisation is in the most efficient form possible.

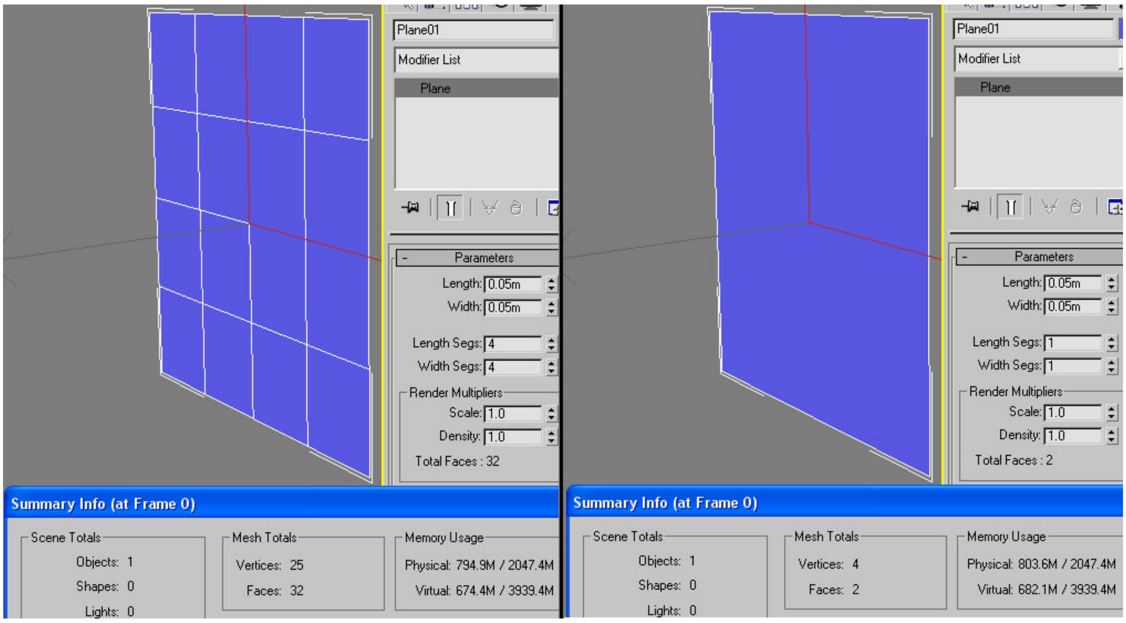
3.1 3D Model Construction

The 3D model construction process will require some careful preparation depending on the requirements of the projects. The designer should know what they really want to build and therefore can plan what is required. Creating 3D models which are too complicated and which will not enhance the final architectural visualisation will be a waste of resources and time.

3.1.1 Segment or Subdivision Amount

A segment or subdivision, as it is sometimes called, is a method of dividing a surface into many parts. By default, 3ds Max normally gives a plane of 4x4 segments during construction. All other 3D models and their parts have segments or subdivisions, regardless of whether it is linear or organic, from a cube, cuboid, pyramid or cylinder to a torus, oil tank or teapot.

Figure 3.1 shows a plane with four segments each for both the width and length in comparison with the same plane with just one segment for both the width and length. It is very clear that the vertices count is a difference between 25 and 4. The faces, which also means triangles amount is a difference between 32 and 2. There is only one object here, which also means geometry. The physical memory used is a difference between 794.9M and 803.6M and the virtual memory consumed is a difference between 674.4M and 682.1M.



Plane01
Modifier List
Plane
Parameters
Length: 0.05m
Width: 0.05m
Length Segs: 4
Width Segs: 4
Render Multipliers
Scale: 1.0
Density: 1.0
Total Faces : 32

Summary Info (at Frame 0)
Scene Totals
Objects: 1
Shapes: 0
Lights: 0
Mesh Totals
Vertices: 25
Faces: 32
Memory Usage
Physical: 794.9M / 2047.4M
Virtual: 674.4M / 3939.4M

Plane01
Modifier List
Plane
Parameters
Length: 0.05m
Width: 0.05m
Length Segs: 1
Width Segs: 1
Render Multipliers
Scale: 1.0
Density: 1.0
Total Faces : 2

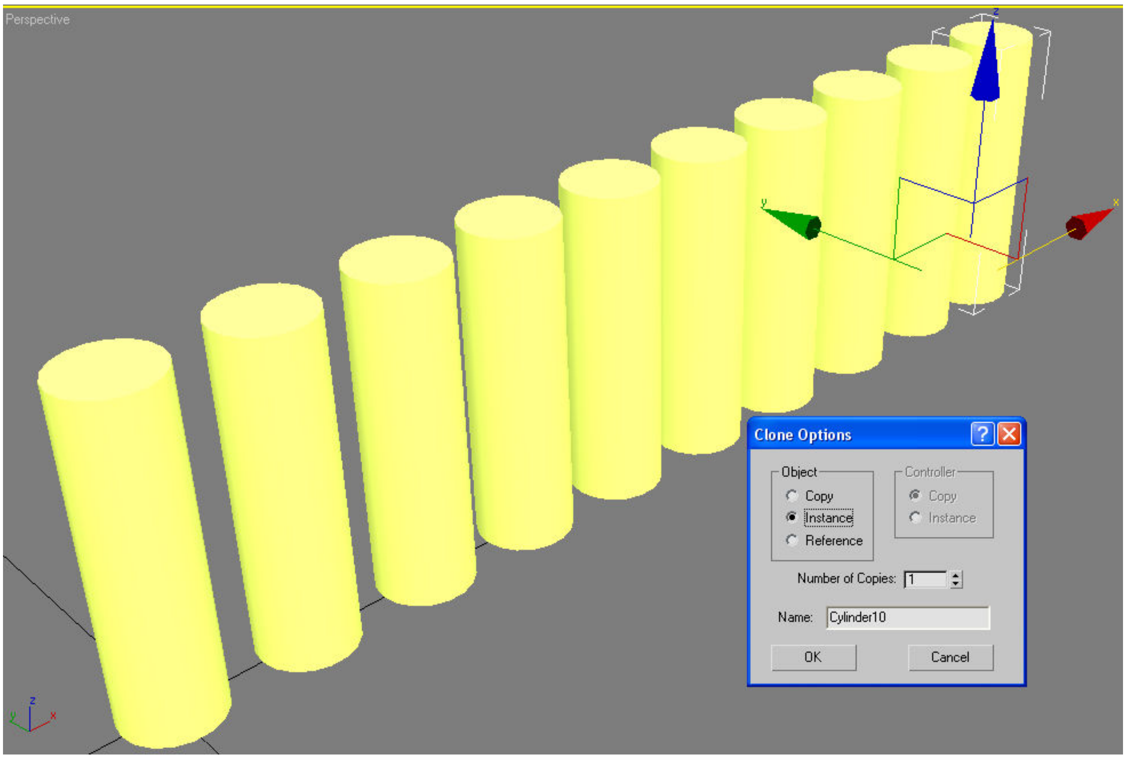
Summary Info (at Frame 0)
Scene Totals
Objects: 1
Shapes: 0
Lights: 0
Mesh Totals
Vertices: 4
Faces: 2
Memory Usage
Physical: 803.6M / 2047.4M
Virtual: 682.1M / 3939.4M

Figure3.1. Comparison of a Plane with 4 Segments and 1 Segment

A simple comparison shows that it is so much more efficient to run a plane with just 1 X 1 segments compared to anything higher. Therefore, a scene with thousands of these planes could make a difference in terms of triangles count, vertices count and memory consumption. If there are no reasons for additional segments and vertices, there is no need to waste resources. There must be reasons to subdivide surfaces which may include curving them, applying high definition textures or achieving better effects when rendering with light sources and shadows on it.

3.1.2 Copy Method

The copy method is exactly like how Blocks are represented in AutoCAD®. A single 3D model which is to be repeated many times in the scene can be represented by just a single object. In 3ds Max®, the CLONE method with INSTANCE choice is used as shown in Figure 3.2. The possibility to have several model transforms associated with the same model includes different locations, orientations, and sizes in the same scene, without requiring replication of the basic geometry. This is because the model vertices and normals are transformed by the model transform (Möller 2002).



Object
○ Copy
● Instance
○ Reference

Controller
● Copy
○ Instance

Number of Copies: 1

Name: Cylinder10

Figure 3.2. Clone Instance of 10 Columns

This method is extremely crucial because it will save a lot of triangle, geometry count and file size. A column, for example, can still be repeated even though there are many variations of it in the scene with different positions, scales and rotations as shown in Figure 3.3.

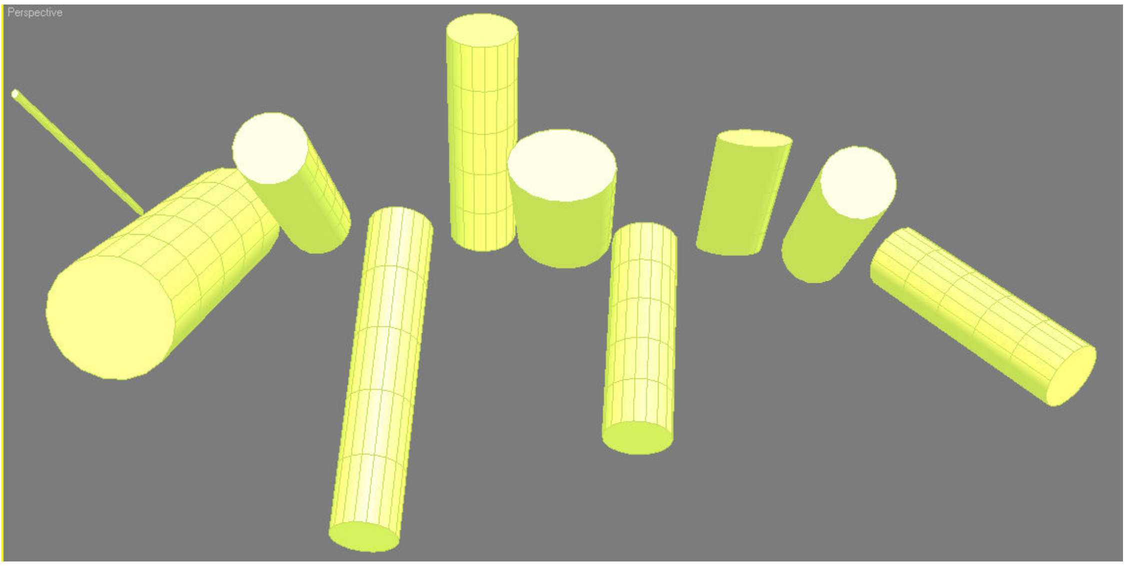
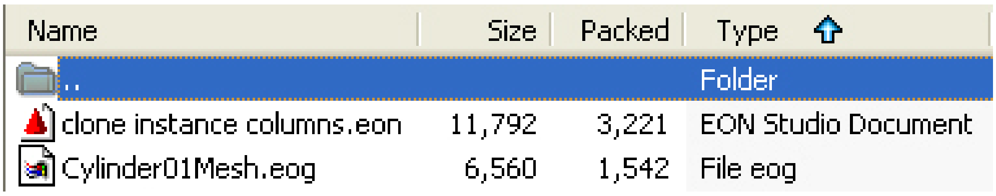


Figure 3.3. Columns Altered in Position, Scale or Rotation

It is clear that the file size and amount of columns remained the same even though transformations of position, scale and rotation are applied to them. Both clone instances with or without transformation end up with just one column and same file size.

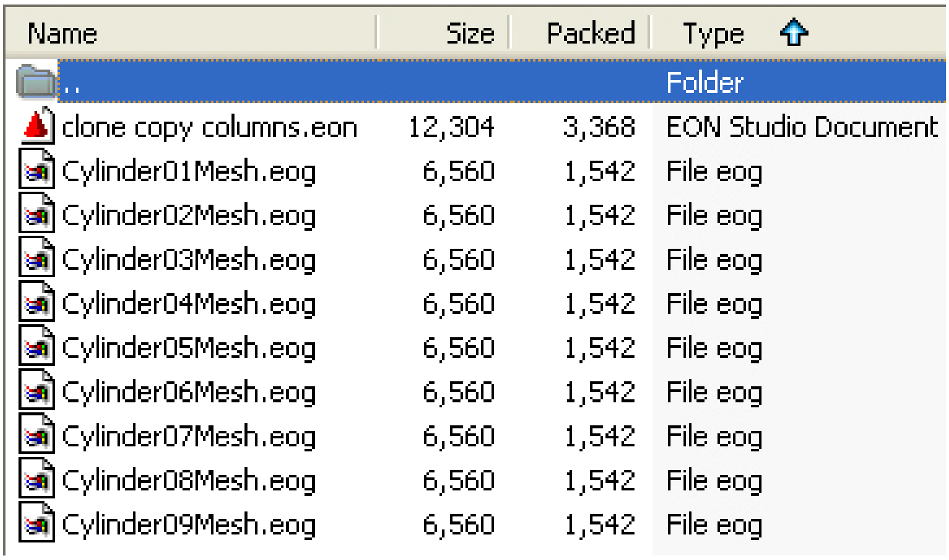


Name	Size	Packed	Type
..			Folder
clone instance columns.eon	11,792	3,221	EON Studio Document
Cylinder01Mesh.eog	6,560	1,542	File eog

Figure 3.4. Clone Instance of 10 Columns

It is different when the CLONE COPY option is used. Even with the same columns, they will come out as 10 separate columns and the file size will increase according to the number of copies created. If the first method is used, the file size of the 10 columns is equivalent to just 6,560 bytes as shown in Figure 3.4. If the second method is used, the file size of the 10 columns will become (6,560 X 9) 59,040 bytes as shown in Figure 3.5.

The difference is 9 times in this case. The EON^{TM} Studio^{TM} document file, EON is also slightly bigger because of the increase of geometry to a total of 10 instead of just one which requires more information to be recorded.



Name	Size	Packed	Type
..			Folder
clone copy columns.eon	12,304	3,368	EON Studio Document
Cylinder01Mesh.eog	6,560	1,542	File eog
Cylinder02Mesh.eog	6,560	1,542	File eog
Cylinder03Mesh.eog	6,560	1,542	File eog
Cylinder04Mesh.eog	6,560	1,542	File eog
Cylinder05Mesh.eog	6,560	1,542	File eog
Cylinder06Mesh.eog	6,560	1,542	File eog
Cylinder07Mesh.eog	6,560	1,542	File eog
Cylinder08Mesh.eog	6,560	1,542	File eog
Cylinder09Mesh.eog	6,560	1,542	File eog

Figure 3.5. Clone Copy of 10 Columns

3.1.3 Billboard / Flat surfaces

This is a method used to represent 3D models in VR simulation. There are two types which are most common. One is always static while the other always faces the camera wherever it moves in the scene. There are a few ways to do billboards depending on the details required based on the chances of passing through it in the simulation navigation. It can be a combination of just one, two, four or more planes to give more volume to the 3D model. Sometimes, a simple geometry shape is added to make it look even more volumetric, For example, to model a pine tree, a cone is added in the middle with all the intersecting planes to provide more volume to its 3D version.

This is sometimes called image-based photorealistic surfaces for people who do research in this field. People always use photorealistic images to replace 3D models of the same thing if it is meant to be for the background, if there is availability of high detailed images to represent the 3D models and if there is not much chance to go near such 3D models during navigation. Some research even provides interactivity to manipulate the object in the distant image in the background (Yoshida *et. al.*, 2002). A higher level usage is image-based rendering which generates many views from a set of reference images. It requires a large number of reference images or normally has geometry information available (Siu *et. al.*, 2004).

3.2 Texture Construction

Texture plays a major role in making 3D models look realistic. Sometimes a very simple 3D model can be enhanced dramatically with good quality textures. There must be a balance between the complexity of 3D models and texture resolution and size depending on the demands and requirements of the final VR visualisation presentation.

3.2.1 Alpha channel textures

Alpha channel textures are there to replace 3D models which have too many holes or empty spaces in between and around it like wire meshes and a tree full of leaves. This is because such details will require a lot more triangles to construct and thus will result in an extremely huge model that is almost impossible to run. Most texture formats like TARGA and DDS can support alpha channel on top of their own basic RGB channels.

The disadvantage of using it is in surfaces with holes that has depth. There is a need to fill the surfaces of the holes extrusion through the two surfaces with surfaces to cover the gaps. This is especially crucial when the holes are deep and people can easily see the usage of alpha channel textures from close by. If the holes are of no or minimum depth like wire meshes, you can afford to use it without any problem as people will not notice it even if they go extremely close in detail to the surfaces.

3.2.2 DirectDraw Surface (DDS) and Power-of-2 Texture Format

A texture format comparison is done to determine the most efficient type to be used in the model for real-time simulation. The common texture formats used in texture mapping are JPEG, PNG and TARGA. A less common format, DDS (DirectDraw Surface) texture created in power-of-2 resolution (example: 512X512, 1024X1024, 2048X2048) and mipmap is found to be the most efficient texture format for real-time rendering. It is an uncompressed format, therefore the application does not need to decompress it every time the simulation starts. The other formats are compressed and this means that they are generally smaller in size in comparison to DDS.

A power-of-2 resolution texture is needed to run VR visualisations faster and efficiently because of the limitations of the graphic card. Therefore, it is advisable to use resolutions of 64, 128, 256, 512, 1024, 2048 or 4096 for both width and height. A texture of 500 X 500 pixel resolution is used for a simulation; it will consume as much memory as the 512 X 512 pixel resolution image. In such a situation, 12 pixels of resolution in each dimension is wasted. For a 513 X 513 pixels resolution map, it will end up using as much memory as the next chunk, which is 1024 X 1024 pixels resolution, meaning almost 3/4 of the memory devoted to that texture is wasted completely.

The graphic card can uncompress in runtime and therefore can load textures faster since it does not need to take time to uncompress. It is possible to have more textures in the scene since they use less memory. Therefore, higher detailed textures (with better resolution) can be afforded. It will benefit the field of architecture because it will improve realism of materials.

There is a format created for the sake of real-time rendering thanks to 3D games progress. The Microsoft® DirectDraw Surface (DDS) is a texture file format introduced with DirectX® 7.0 and later in OpenGL™. It can store uncompressed and compressed pixel formats with mipmap or without mipmap levels. Mipmap is a texture mapping technique using multiple texture maps as shown in Figure 3.6. Every MIP map is half the size of the previous one, providing several texture maps of various levels of depth. MIP stands for "multum in parvo", which is Latin for "many in a small place."

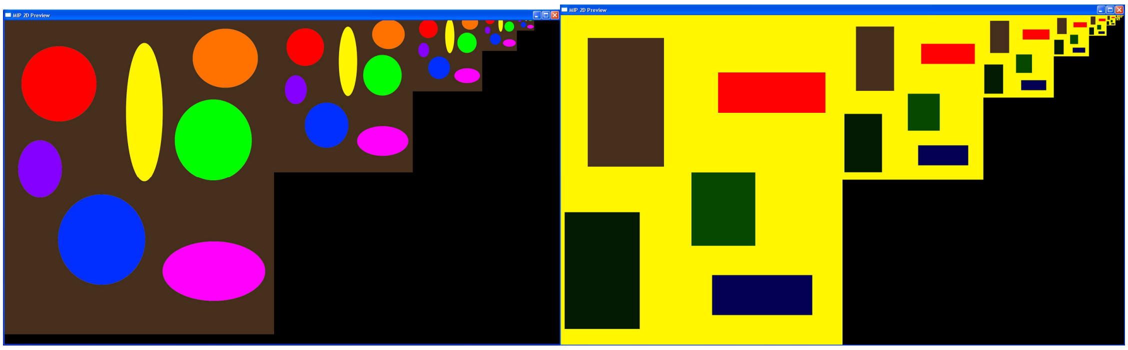


Figure 3.6. Mipmapping of Textures

A simple test is done by creating four surfaces as shown in Figure 3.7. They are run in the simulation without any textures applied. Later, two textures of 2048 X 2048 72 dpi and 1024 X 1024 72dpi in both JPEG and DDS formats are created. It can be seen that the DDS file format is huge because it has mipmaps and in this case, it is created into 10 levels. Therefore, it will smartly project the smallest map when navigating furthest away from the texture and it will project the biggest and best resolution map when navigating closest to it and all the different levels in between as the distance varies. The memory usage here is a vital reason why DDS format is used. The jpeg format requires 53MB to run the simulation while the DDS format only requires 8MB, which is almost 7 times the difference.

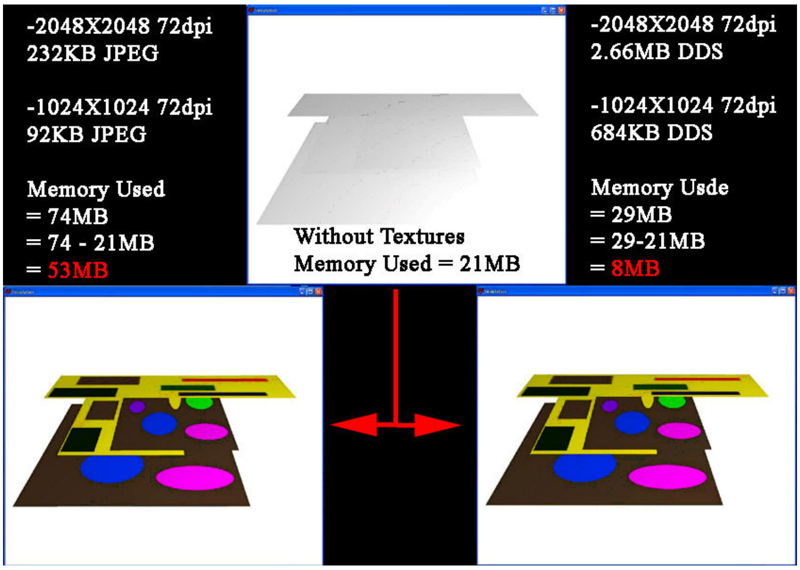


Figure 3.7. Comparison between JPEG and DDS

The replacement of textures from JPEG to DDS is done for two projects which are running at extremely low frame rates. The first project is Ming De of Chang'An (1,148,005 triangles) which has an average frame rate of 2.5 when it originally had JPEG textures as shown in Figure 3.8. Chang'An is a historical reconstruction project done by research in the architecture department and Ming De is a part of the whole city. The replacement to DDS textures pushed the frame rate up to 6. The other project tested is Warren Housing (2,623,190 triangles) which runs with a frame rate of 4 with JPEG textures but improved to 7 after being replaced with DDS textures. Warren Housing is a proposed NUS future hostel construction which is constructed for research visualisation by the department of architecture as shown in Figure 3.9.

The frame rate for both projects is pushed above the minimum requirement of 6 fps because of changing to DDS textures. It is therefore concluded that using DDS textures is better than traditional textures as they are more efficient in memory consumption as well as help improve performance in the overall run.

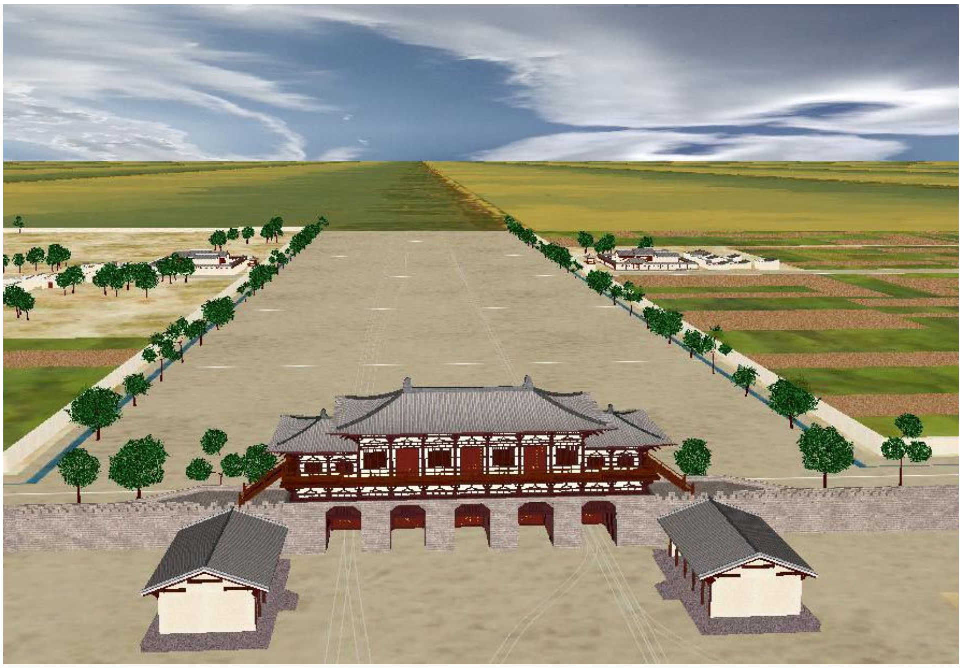


Figure 3.8. Chang An's Ming De

Figure 3.9. Warren Housing

3.2.3 Texture Substituting Technique

It is crucial to have geometric forms substitution with textures to represent them if a high quality texture can do the job. A bump map can even be applied to the surface to make it look rough. This technique is different than the billboard method mentioned earlier as it targets mostly flat surface details. For example, a building façade with doors and window panels can be represented by just a flat image with all the doors and windows being part of the whole image integrated with the wall material. This will greatly reduce the triangle counts for all the walls by building only blocks of cubes or cuboids with texture maps to represent facades. This is done only if a high resolution texture image is available or if there is not much chance for a user to go near such facades or if the facades serve as backdrops at a far distance during simulation. Only in the situations mentioned above will this technique be really useful and effective. As of today, the maximum recommended resolution is 4096 X 4096 pixels.

3.3 View Setup

View setup is the next most important aspect to control apart from 3D model construction and texture construction. This is especially crucial when the scene has a very complicated and huge 3D model with high resolution textures. The main intention is to only show what can be seen by the camera. This will free up a lot of computational power and memory.

3.3.1 View Culling Techniques

View culling is the method used to only show what can be seen through the camera during navigation. It is meant not to process data that will not contribute to the final image. There is no point in wasting computer processing power in rendering everything even if you will not see them on the screen. It is highly recommended to be used in 3D VR visualisations to cut down on unseen triangles in the scene. There are many types of culling techniques which include view frustum, backface, portal and occlusion culling as shown in Figure 3.10.

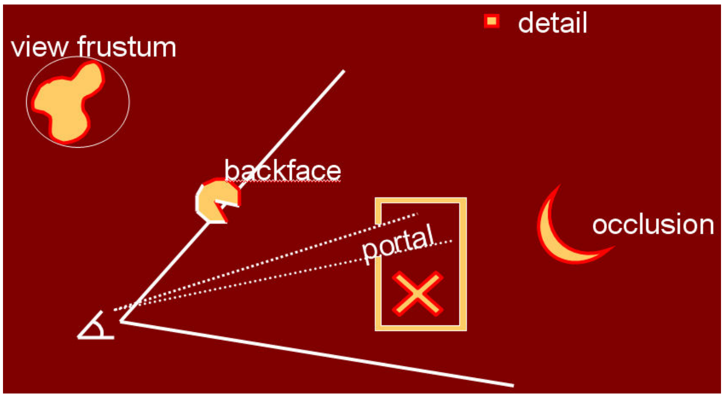
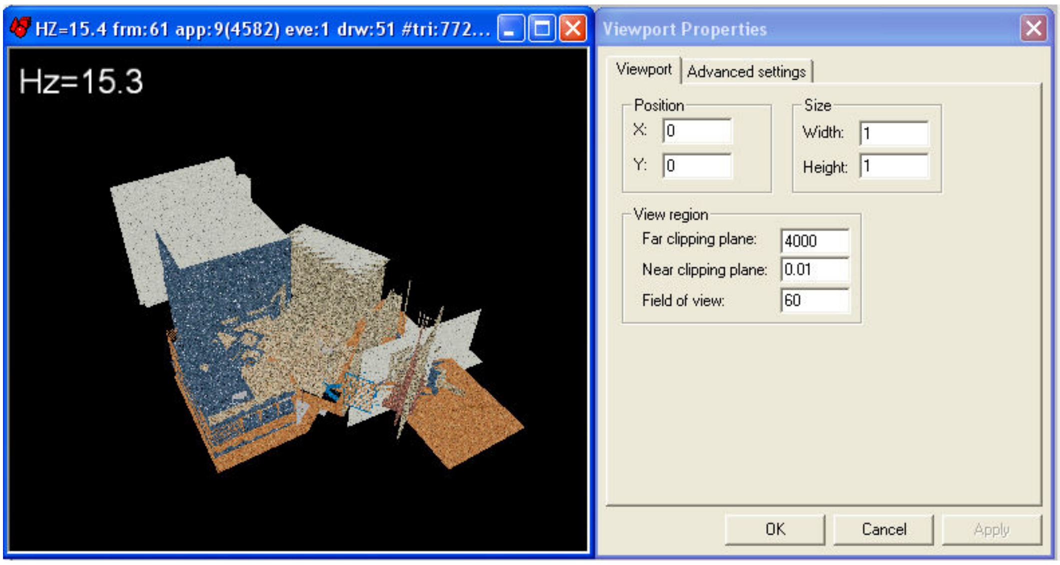


Figure 3.10. Culling techniques
Source: http://www.realtimerendering.com

For view frustum culling, if a bounding volume is outside the view frustum, then the contents will not be visible and not rendered. For backface culling, all the surfaces that are backfacing the viewport will not be rendered. Occlusion culling is almost similar by not rendering objects that lie completely behind another object or set of objects. Portal culling will systematically not render objects that are blocked by surfaces (normally walls) in the view while navigating through rooms.

3.3.2 Clipping Planes

The near and far clipping planes are the lower and upper limits of where 3D models are contained a shown in Figure 3.11. It is crucial to set the planes just nice enough to cover the whole model so that resources are not wasted by forcing the system to calculate beyond what is necessary. It will smooth up the navigation because no added calculations are done in the backdrop to find any 3D models in the empty space behind or in front of the actual 3D model. Good adjustment of the clipping planes will also make sure that no parts of the 3D model are left outside the two planes and thus ending up not seeing them in the scene.



Position
X: 0
Y: 0

Size
Width: 1
Height: 1

View region
Far clipping plane: 4000
Near clipping plane: 0.01
Field of view: 60

Figure 3.11. The Near and Far Clipping Planes Setting

In some projects where only walkthroughs will be done, there is no point in setting the clipping planes to see the whole model if during walkthrough, many parts of the model are beyond our eyesight. In a situation like that, there is a need to set it far enough to a distance where our eyes will not be able to see any further. A good example is navigating indoors where there are many walls blocking the view of the whole 3D scene. In situations like that, the surrounding walls should be within the clipping planes range and whatever is beyond the walls will be outside the clipping planes.

3.3.3 Level of Detail (LOD)

Level of Detail (LOD) is a famous method used to optimise simulations. It is used in many aspects of VR simulations including in the advanced or programmer shaders (Simmons and Shreiner 2003). Shader simplification with a system is used as well to automatically creates shader levels of detail to reduce the number of texture accesses (Olano et. al., 2003). Traditionally, LOD is used to reduce the geometric complexity of 3D models (Xia et. al., 1997) in cases where the hardware is struggling to render those complex models. It normally has 3 different complexities of the same model to be projected depending on the distance of the viewer to the 3D model.

It is usually used when there is a need to navigate through large virtual environments and people have different ways of using it, either by using different techniques or creating algorithms (Chhugani et. al., 2005). The nearer the viewer is to the 3D model, the higher the quality of the 3D model that will be shown. It is done by many others depending on their research interest, which includes plant motion (Beaudoin and Keyser 2004). The difference in human perception between supra-threshold LOD control and LOD control at threshold (Watson et. al., 2004) is examined by others to understand human feedback on it and how contrast plays a role in it.

In most software including EON^{TM} Studio^{TM} Professional, the transition between a higher quality model to the next is sometimes not smooth and thus, people can sense that LOD technique is used in the simulation. Therefore, the transition between a higher quality model to a lower one cannot be too drastic. The purpose of using this technique will be lost if more triangle counts cannot be saved. Another disadvantage of using this technique is the requirement to have three to four versions of the same model created at different quality levels. This will actually require more time to build the 3D model and thus ending up with a file with bigger size.

3.4 Optimisation Algorithms

This is the final step to take if the 3D model prepared is still not performing smoothly during navigation after going through all the stages above. This step is normally taken when one does not have enough time to go back to the earlier stages and if just a bit more tweaking is needed to enable the visualisation to perform better. It will help especially in cases where a 3D model is just short of achieving a smooth navigation speed. The risk of heavy optimisation here will result in a 3D model with missing triangles and surfaces.

3.4.1 Polygon Reductions and Simplifications

Many papers and software have different algorithm methods of reducing modelling size (either its vertices, faces, edges and triangle counts). A prominent figure who did a lot of work in this area is Michael Garland who explored many techniques to optimise and simplify 3D models. The interest of the architecture school is to find the best reduction / simplification software that best suits different CAD models from all the techniques created. Geoffrey Bawa's Singapore Botanical Garden Cloud Forest Biosphere 3D terrain model is used for this exercise with the canal project, Mario Botta's The Cymbalista Synagogue and Jewish Heritage Center and Mies Van der Rohe Farnsworth's House. All the models used are in 3DS and DXF file format. The Cloud Forest Biosphere is an unbuilt project which was constructed as part of this research. The canal project and Farnsworth's House are 3D models constructed by fellow students in their design and independent study module. The Cymbalista Synagogue and Jewish Heritage Centre 3D model was constructed for an independent study module.

Software chosen to perform the experiments are:

(i) **Okino Polytrans / Nugraf** (Michael Garland's Quadric Error Metrics Algorithm) as shown in Figure 3.13.

This technique uses what Garland calls quadric error metrics (QEM) and vertex pair contraction, which has been restricted to edge contraction in this application (Garland and Heckbert 1997 & 1998). Edge contraction simply means that the two end vertices of a model edge are replaced by a single new vertex as shown in Figure 3.12. This target vertex is usually somewhere in between the other two, in a place where it best approximates the original model. This edge contraction step removes a vertex and one, two, or more faces from the model, depending on the mesh neighbourhood.

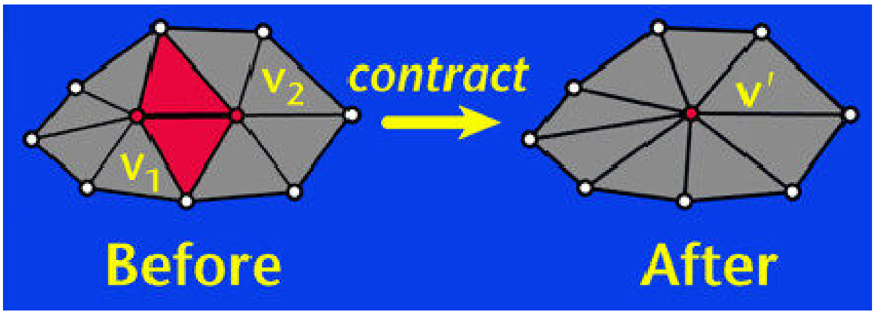


Figure 3.12. Garland's Iterative Edge Contraction
Source: http://graphics.cs.uiuc.edu~/garland/home.html

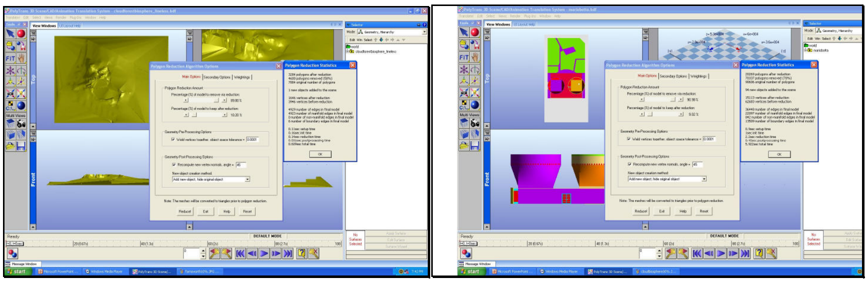
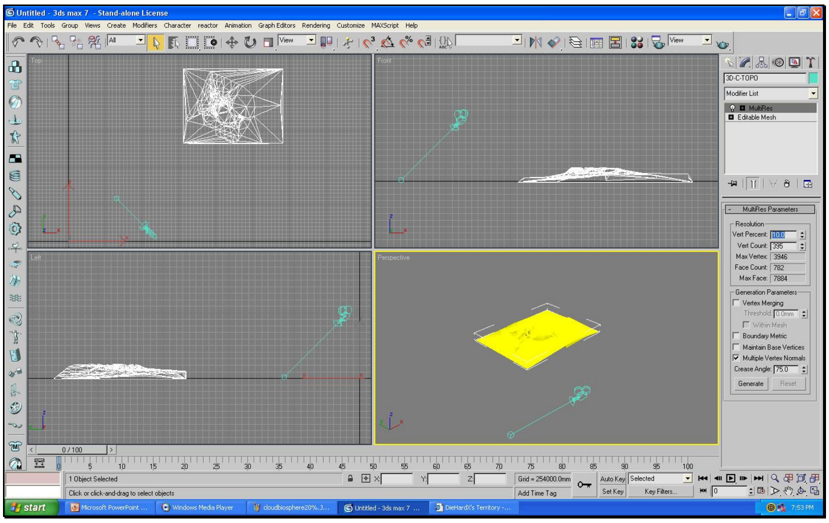


Figure 3.13. Okino Polytrans (90% Reduction) Cloud Forest Biosphere and Jewish Synagogue

(ii) **Autodesk® 3ds Max®** (Multiresolution Mesh Decimation Algorithm) as shown in Figure 3.13.



Resolution
Vert Percent: 10.0
Vert Count: 395
Max Vertex: 3946
Face Count: 782
Max Face: 7884

Generation Parameters
☐ Vertex Merging
Threshold: 0.0mm
☐ Within Mesh
☐ Boundary Metric
☐ Maintain Base Vertices
☑ Multiple Vertex Normals
Crease Angle: 75.0

Figure 3.14. 3ds Max® Cloud Forest Biosphere (90% reduction)
(only mesh models can be reduced so only this model can be tested)

(iii) **Right Hemisphere Deep Exploration** (EON^{TM} CAD – triangles reduction algorithm) as shown in Figure 3.15.

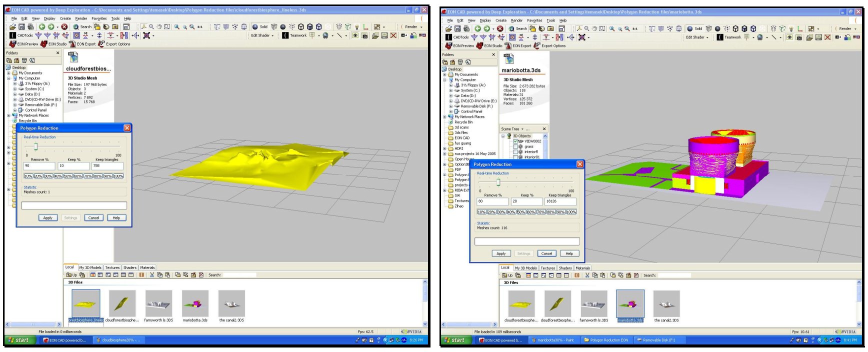


Figure 3.15. Right Hemisphere Deep Exploration (90% & 80% Reduction)
Cloud Forest Biosphere and Jewish Synagogue

(iv) **Raindrop Geomagic** (3D Scanner editing software –triangles reduction based upon the deviation of new vertices to the original polygonal model surface) as shown in Figure 3.16.

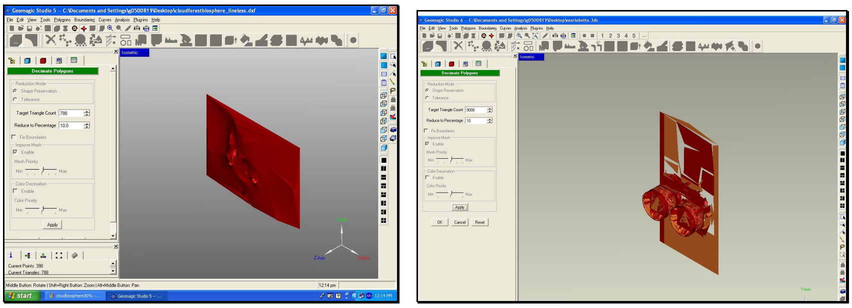
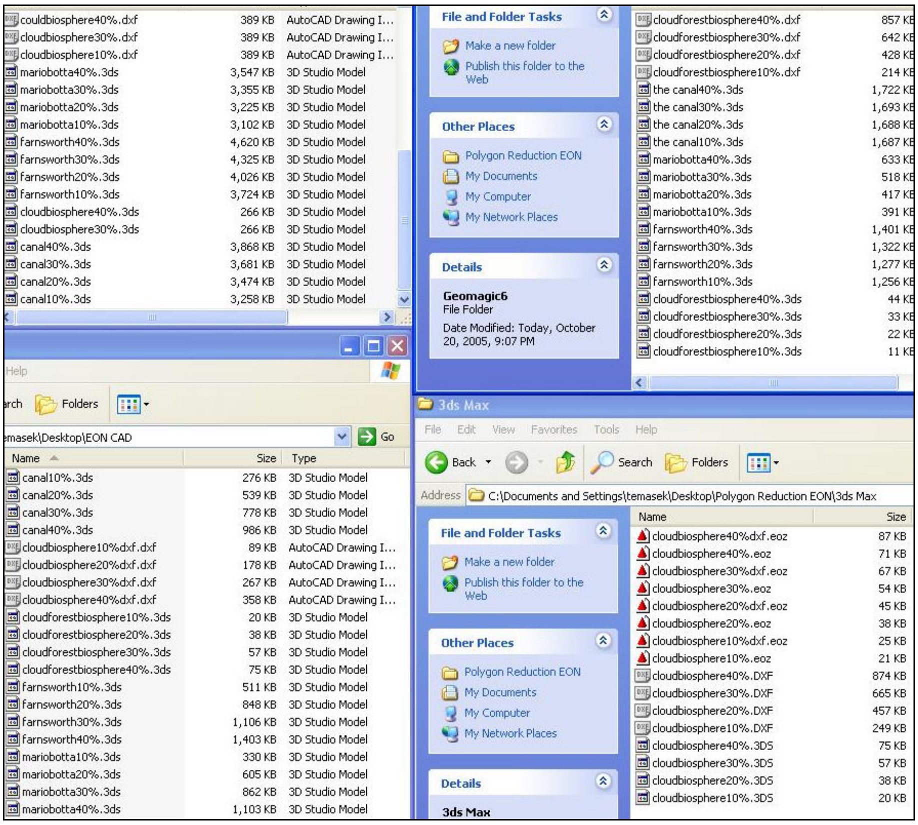


Figure 3.16. Raindrop Geomagic (90% reduction) and Cloud Forest Biosphere & Jewish Synagogue

All four software simplification and reduction from 100% to the range of 10% - 40% are shown in Figure 3.17.



couldbiosphere40%.dxf	389 KB	AutoCAD Drawing I...
cloudbiosphere30%.dxf	389 KB	AutoCAD Drawing I...
cloudbiosphere10%.dxf	389 KB	AutoCAD Drawing I...
mariobotta40%.3ds	3,547 KB	3D Studio Model
mariobotta30%.3ds	3,355 KB	3D Studio Model
mariobotta20%.3ds	3,225 KB	3D Studio Model
mariobotta10%.3ds	3,102 KB	3D Studio Model
farnsworth40%.3ds	4,620 KB	3D Studio Model
farnsworth30%.3ds	4,325 KB	3D Studio Model
farnsworth20%.3ds	4,026 KB	3D Studio Model
farnsworth10%.3ds	3,724 KB	3D Studio Model
cloudbiosphere40%.3ds	266 KB	3D Studio Model
cloudbiosphere30%.3ds	266 KB	3D Studio Model
canal40%.3ds	3,868 KB	3D Studio Model
canal30%.3ds	3,681 KB	3D Studio Model
canal20%.3ds	3,474 KB	3D Studio Model
canal10%.3ds	3,258 KB	3D Studio Model

Geomagic6
File Folder
Date Modified: Today, October 20, 2005, 9:07 PM

cloudforestbiosphere40%.dxf	857 K
cloudforestbiosphere30%.dxf	642 K
cloudforestbiosphere20%.dxf	428 K
cloudforestbiosphere10%.dxf	214 K
the canal40%.3ds	1,722 K
the canal30%.3ds	1,693 K
the canal20%.3ds	1,688 K
the canal10%.3ds	1,687 K
mariobotta40%.3ds	633 K
mariobotta30%.3ds	518 K
mariobotta20%.3ds	417 K
mariobotta10%.3ds	391 K
farnsworth40%.3ds	1,401 K
farnsworth30%.3ds	1,322 K
farnsworth20%.3ds	1,277 K
farnsworth10%.3ds	1,256 K
cloudforestbiosphere40%.3ds	44 K
cloudforestbiosphere30%.3ds	33 K
cloudforestbiosphere20%.3ds	22 K
cloudforestbiosphere10%.3ds	11 K

emasek\Desktop\EON CAD

Name	Size	Type
canal10%.3ds	276 KB	3D Studio Model
canal20%.3ds	539 KB	3D Studio Model
canal30%.3ds	778 KB	3D Studio Model
canal40%.3ds	986 KB	3D Studio Model
cloudbiosphere10%dxf.dxf	89 KB	AutoCAD Drawing I...
cloudbiosphere20%dxf.dxf	178 KB	AutoCAD Drawing I...
cloudbiosphere30%dxf.dxf	267 KB	AutoCAD Drawing I...
cloudbiosphere40%dxf.dxf	358 KB	AutoCAD Drawing I...
cloudforestbiosphere10%.3ds	20 KB	3D Studio Model
cloudforestbiosphere20%.3ds	38 KB	3D Studio Model
cloudforestbiosphere30%.3ds	57 KB	3D Studio Model
cloudforestbiosphere40%.3ds	75 KB	3D Studio Model
farnsworth10%.3ds	511 KB	3D Studio Model
farnsworth20%.3ds	848 KB	3D Studio Model
farnsworth30%.3ds	1,106 KB	3D Studio Model
farnsworth40%.3ds	1,403 KB	3D Studio Model
mariobotta10%.3ds	330 KB	3D Studio Model
mariobotta20%.3ds	605 KB	3D Studio Model
mariobotta30%.3ds	862 KB	3D Studio Model
mariobotta40%.3ds	1,103 KB	3D Studio Model

Address C:\Documents and Settings\temasek\Desktop\Polygon Reduction EON\3ds Max

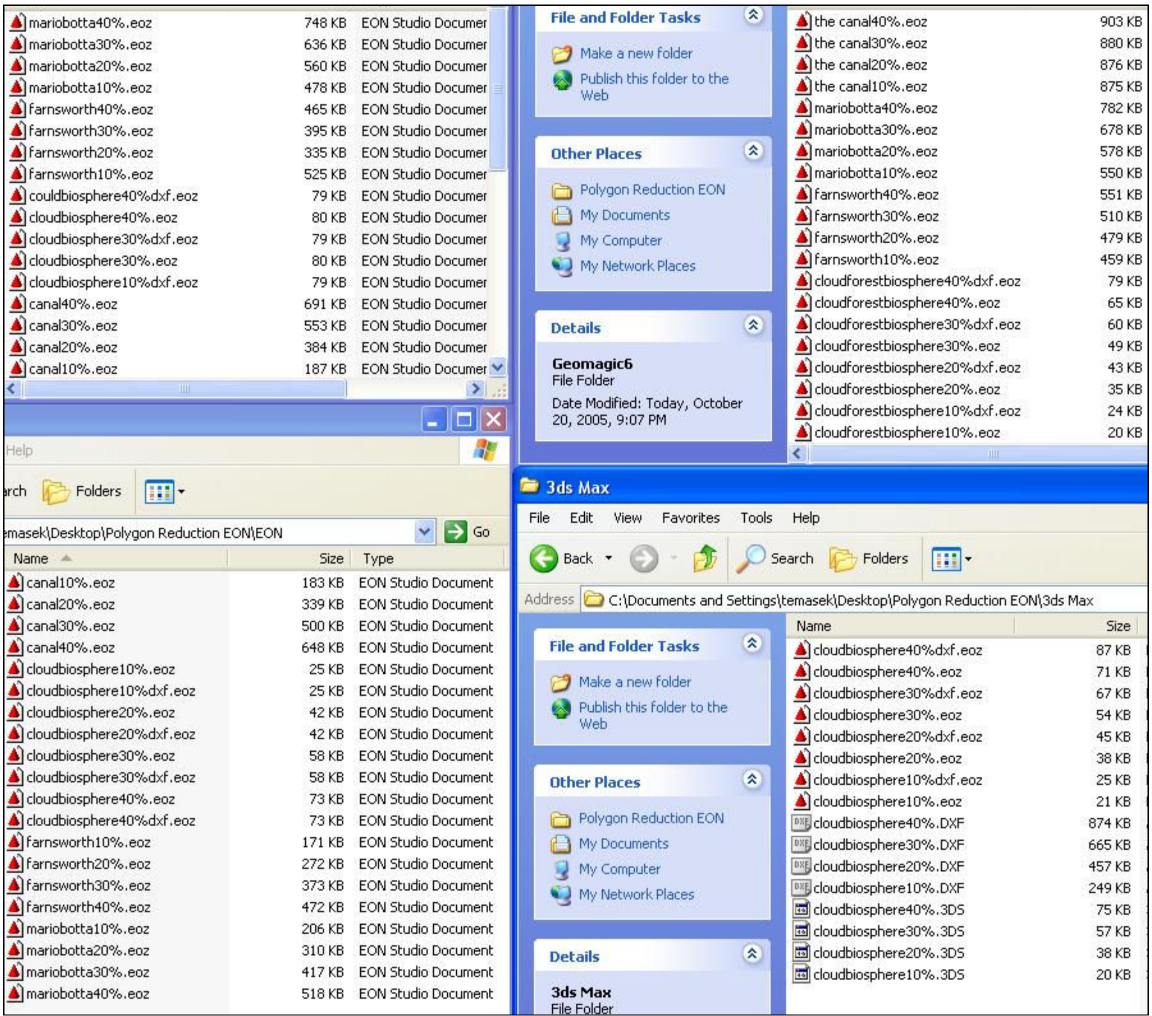
Name	Size
cloudbiosphere40%dxf.eoz	87 KB
cloudbiosphere40%.eoz	71 KB
cloudbiosphere30%dxf.eoz	67 KB
cloudbiosphere30%.eoz	54 KB
cloudbiosphere20%dxf.eoz	45 KB
cloudbiosphere20%.eoz	38 KB
cloudbiosphere10%dxf.eoz	25 KB
cloudbiosphere10%.eoz	21 KB
cloudbiosphere40%.DXF	874 KB
cloudbiosphere30%.DXF	665 KB
cloudbiosphere20%.DXF	457 KB
cloudbiosphere10%.DXF	249 KB
cloudbiosphere40%.3DS	75 KB
cloudbiosphere30%.3DS	57 KB
cloudbiosphere20%.3DS	38 KB
cloudbiosphere10%.3DS	20 KB

Figure 3.17. All the CAD files Simplification / Reduction from 100% to 10% - 40% (UpperLeft: Polytrans UpperRight: Geomagic LowerRight: 3ds Max® LowerLeft:EON^{TM} CAD)

There are many other algorithm methods in the market which are either developed individually by computer science students and lecturers or commercially released.

(i) Action3D Inc. Action3D (Only supports X and 3DS file formats)
(ii) Trapezium Chisel
(iii) Antonio Cortés Carrillo Progressive Fans
(iv) Erik Pojar's Progressive Meshes (Maya® plug-in)
(v) Paralelo Computação's ProgMesh
(vi) System In Motion Rational Reducer (3ds Max® plug-in)
(vii) Realax RXpolyred
(viii) Sergei Zawelsky VIZup (only supports VRML WRL file format)

The next step is to export all the 3D models into EON™ CAD and EON™ Studio™ Professional™. The entire list of them are shown in Figure 3.18.



Name	Size	Type
mariobotta40%.eoz	748 KB	EON Studio Documer
mariobotta30%.eoz	636 KB	EON Studio Documer
mariobotta20%.eoz	560 KB	EON Studio Documer
mariobotta10%.eoz	478 KB	EON Studio Documer
farnsworth40%.eoz	465 KB	EON Studio Documer
farnsworth30%.eoz	395 KB	EON Studio Documer
farnsworth20%.eoz	335 KB	EON Studio Documer
farnsworth10%.eoz	525 KB	EON Studio Documer
couldbiosphere40%dxf.eoz	79 KB	EON Studio Documer
cloudbiosphere40%.eoz	80 KB	EON Studio Documer
cloudbiosphere30%dxf.eoz	79 KB	EON Studio Documer
cloudbiosphere30%.eoz	80 KB	EON Studio Documer
cloudbiosphere10%dxf.eoz	79 KB	EON Studio Documer
canal40%.eoz	691 KB	EON Studio Documer
canal30%.eoz	553 KB	EON Studio Documer
canal20%.eoz	384 KB	EON Studio Documer
canal10%.eoz	187 KB	EON Studio Documer

File and Folder Tasks
Make a new folder
Publish this folder to the Web

Other Places
Polygon Reduction EON
My Documents
My Computer
My Network Places

Details
Geomagic6
File Folder
Date Modified: Today, October 20, 2005, 9:07 PM

Name	Size
the canal40%.eoz	903 KB
the canal30%.eoz	880 KB
the canal20%.eoz	876 KB
the canal10%.eoz	875 KB
mariobotta40%.eoz	782 KB
mariobotta30%.eoz	678 KB
mariobotta20%.eoz	578 KB
mariobotta10%.eoz	550 KB
farnsworth40%.eoz	551 KB
farnsworth30%.eoz	510 KB
farnsworth20%.eoz	479 KB
farnsworth10%.eoz	459 KB
cloudforestbiosphere40%dxf.eoz	79 KB
cloudforestbiosphere40%.eoz	65 KB
cloudforestbiosphere30%dxf.eoz	60 KB
cloudforestbiosphere30%.eoz	49 KB
cloudforestbiosphere20%dxf.eoz	43 KB
cloudforestbiosphere20%.eoz	35 KB
cloudforestbiosphere10%dxf.eoz	24 KB
cloudforestbiosphere10%.eoz	20 KB

Address C:\Documents and Settings\temasek\Desktop\Polygon Reduction EON\3ds Max

File and Folder Tasks
Make a new folder
Publish this folder to the Web

Other Places
Polygon Reduction EON
My Documents
My Computer
My Network Places

Details
3ds Max
File Folder

Name	Size
cloudbiosphere40%dxf.eoz	87 KB
cloudbiosphere40%.eoz	71 KB
cloudbiosphere30%dxf.eoz	67 KB
cloudbiosphere30%.eoz	54 KB
cloudbiosphere20%dxf.eoz	45 KB
cloudbiosphere20%.eoz	38 KB
cloudbiosphere10%dxf.eoz	25 KB
cloudbiosphere10%.eoz	21 KB
cloudbiosphere40%.DXF	874 KB
cloudbiosphere30%.DXF	665 KB
cloudbiosphere20%.DXF	457 KB
cloudbiosphere10%.DXF	249 KB
cloudbiosphere40%.3DS	75 KB
cloudbiosphere30%.3DS	57 KB
cloudbiosphere20%.3DS	38 KB
cloudbiosphere10%.3DS	20 KB

emasek\Desktop\Polygon Reduction EON\EON

Name	Size	Type
canal10%.eoz	183 KB	EON Studio Document
canal20%.eoz	339 KB	EON Studio Document
canal30%.eoz	500 KB	EON Studio Document
canal40%.eoz	648 KB	EON Studio Document
cloudbiosphere10%.eoz	25 KB	EON Studio Document
cloudbiosphere10%dxf.eoz	25 KB	EON Studio Document
cloudbiosphere20%.eoz	42 KB	EON Studio Document
cloudbiosphere20%dxf.eoz	42 KB	EON Studio Document
cloudbiosphere30%.eoz	58 KB	EON Studio Document
cloudbiosphere30%dxf.eoz	58 KB	EON Studio Document
cloudbiosphere40%.eoz	73 KB	EON Studio Document
cloudbiosphere40%dxf.eoz	73 KB	EON Studio Document
farnsworth10%.eoz	171 KB	EON Studio Document
farnsworth20%.eoz	272 KB	EON Studio Document
farnsworth30%.eoz	373 KB	EON Studio Document
farnsworth40%.eoz	472 KB	EON Studio Document
mariobotta10%.eoz	206 KB	EON Studio Document
mariobotta20%.eoz	310 KB	EON Studio Document
mariobotta30%.eoz	417 KB	EON Studio Document
mariobotta40%.eoz	518 KB	EON Studio Document

Figure 3.18. All the output EON^{TM} Studio^{TM} Professional^{TM} files from above. (UpperLeft: UpperRight: LowerRight: LowerLeft)

From the comparisons done, a few conclusions are obvious:

(i) Parameters to control reduction are useful (boundary reduction, average vertices reduction, vertex clustering (average of two vertices), vertices stitching), and the more the better.

(ii) There is limited reduction as the percentage goes beyond 70% and above; some algorithms will prevent further reductions.

(iii) Each software has its strength and weakness on reduction depending on different modelling types (landscape, organic to linear models).

(iv) Smaller files may be too ugly to show at all, sometimes the reduction will make some surfaces totally disappear.

(v) 3ds Max®'s Multires modifier only works for mesh models, not spline / patch or NURBS.

(vi) EON^{TM} CAD seems most efficient without tweaking any parameters for all four software probably because it is the best medium of preparation before going into EON^{TM} Studio^{TM} Professional

Therefore, this method is only for very complicated terrains / organic objects because the loss of vertices will not be very visible to the naked eye. If for any reason, there are many details to be drawn in linear modelling, polygon reductions / simplifications have to be done layer by layer to maintain its form so that it will not be too distorted or have the 'after destruction/explosion' feeling.

3.4.2 Final Optimisation

VR software have their own optimisation functions which are normally at the mesh / polygon, geometry and texture levels. EON^{TM} CAD and EON^{TM} Studio^{TM} Professional^{TM} have all three of them and final optimisations can be done if one is really running out of time and there are no other choices at the final stage of design. Polygon reduction, texture and geometry compressions can help improve performance of the final simulation.

The reduction or compression is done in percentages and levels for all three of them (mesh/polygon, geometry and texture levels). The user can change the percentages and levels to a point that is still acceptable before saving the file for the final simulation presentation. This method will also decrease the geometry and texture file sizes used in the whole 3D simulation and will push down the overall size of the EON^{TM} Studio^{TM} Professional^{TM}'s EOZ file format.

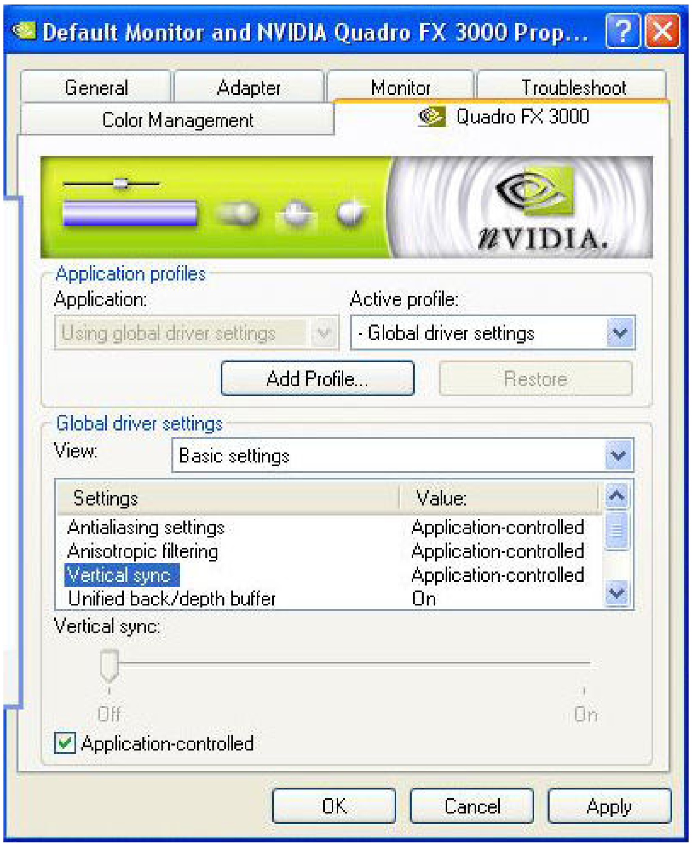
In conclusion, there are many optimisation techniques which can be used in VR architectural visualisation projects. Some of them are crucial and considered compulsory, like using the right copy method, using alpha channel textures, DDS textures, view culling technique and clipping planes. However, some of them require more preparation time in addition to producing poorer quality 3D models and textures like LOD, polygon reductions and simplifications, geometric forms substitution and final optimisations by VR visualisation software. Therefore, some of the techniques are optional unless the user is left with no more choices at the end of the day. The understanding of the pros and cons of each technique is crucial for the next step of the research, which is to be used in the independent variable tests in the next chapter.

CHAPTER 4.0: INDEPENDENT VARIABLE TESTS

From the previous chapter on optimisation techniques, a series of tests were done to verify their usefulness in application for VR visualisation projects. In this chapter, the techniques will be applied in all the independent variable tests wherever applicable. The output performance of VR presentations will be monitored, which are measured by frame rate, texture memory and vertex memory.

From the literature review, four basic variables that contribute to the decline in performance were identified. The tests carried out in this chapter help prove whether these variables do affect frame rate and memory consumption directly and how they are related. Apart from that, many other variables are tested because they are used in specific projects created as well as speculations by some observations done in VR projects. In addition, the investigation into the time taken to navigate from the starting point to the ending point is done at the initial stage of the independent variable test. All the experiments are done in stereoscopic mode and with a DELL^{TM} Precision^{TM} 650 workstation except the vertex count and triangle size, which are done with the DELL^{TM} Inspiron^{TM} 9300. All 3D models used in the experiment are related to architecture. It does not matter what objects they are since all objects consist of triangles (more information in Chapter 4.1.3).

The vertical sync (VSYNC) setting for the graphic card is application controlled during the whole experiment as shown in Figure 4.1. With VSYNC enabled, frame rates will not exceed the monitor's refresh rate for that particular resolution. If the monitor refresh rate is set to run at 60Hz, the VSYNC when enabled will never exceed 60fps. If it is enabled, it can cause image tearing and flashing polygons. (Agi 2000) Therefore, disabling the VSYNC can help push frame rate up and this is crucial when you are running extremely low frame rate.



Application: Using global driver settings

Active profile: - Global driver settings

View: Basic settings

Settings	Value:
Antialiasing settings	Application-controlled
Anisotropic filtering	Application-controlled
Vertical sync	Application-controlled
Unified back/depth buffer	On

Vertical sync: Off — On

☑ Application-controlled

Figure 4.1. Graphic Card VSYNC Setting

The final part is to investigate how much hardware and software will influence the outcome of the projects' performances. Initial guesses are that hardware and software will influence the performance and therefore, there is a need to test them out to prove it.

For triangle count, geometry count and advanced / programmable shader count, graphs were plotted to identify the limit that can be put in the scene at 6fps. This is the minimum frame rate for real-time navigation identified from the literature review.

4.1 Primary Variables

This section will look into the first generation variables or primary variables. These are fundamental variables which are basic elements used to construct a VR visualisation. These core variables will be measured against frame rate. Initially, the relationship between time taken and triangle size will be explored and determined. Apart from the four fundamental variables, exploration into other related aspects like triangle size and texture resolution will be conducted to find out whether they have any relationship with frame rate.

4.1.1 Frame Rate and Time Taken

This test was done to observe the frame rate and time taken to navigate through a scene planted with trees. It is because of Geoffrey Bawa’s unbuilt Cloud Forest Biosphere VR project that the test for tree count is done. The project is for display of cloud forest trees and plants and therefore the focus is to find out how to model trees in a 3D scene in the most efficient way possible. Therefore, the main objective of this experiment is to know the maximum quantity of different tree types that can be modelled in the 3D scene before the simulation is no longer real-time.

Four types of trees were used in the experiment and for each type, a representation of the most common complex trees was chosen. They are the most common trees used by architecture students. The frame rate and time taken for a distance travelled were recorded. This was done with a number of trees from 10 to 1,600. A graph was later plotted for the number of triangles of the trees against frame rate. From the graph, at the frame rate of 6fps, the maximum amount of trees for each method on each tree type used can be obtained. The time taken for the navigation from the starting point to the ending point of the full view of all the trees was recorded.

(i) 3D 3ds Max® trees

Two 3D 3ds Max® trees of different complexity were used in the experiment as shown in Figure 4.2 and 4.3. These are purely 3D trees without any flat surfaces which are available in the 3ds Max® library.

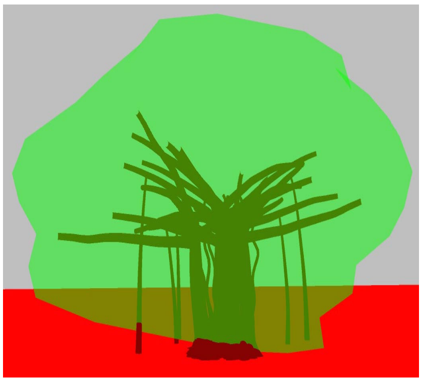


Figure 4.2. Banyan Tree (3D 3ds Max® tree) 10, 50, 100, 200 and 400 trees

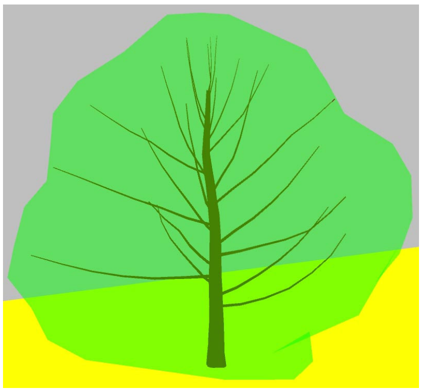


Figure 4.3. American Elm (3D 3ds Max® tree) 10, 50, 100, 200 and 400 trees

The graph in Figure 4.4 shows the trees in triangles count plotted against frame rate. As the amount increases from 10, 50, 100, 200 and 400 trees, the frame rate drops significantly. At frame rate 6, the American Elm recorded 4,941,765 triangles and the Banyan Tree recorded.4,971,279 triangles. At a frame rate of 6, the limit of display is 1703 number of American Elms (4,941,765 triangles) and 75 number of Banyan Trees (4,971,279 triangles). The formula enables the prediction of frame rate (Y) with the amount of triangles known.

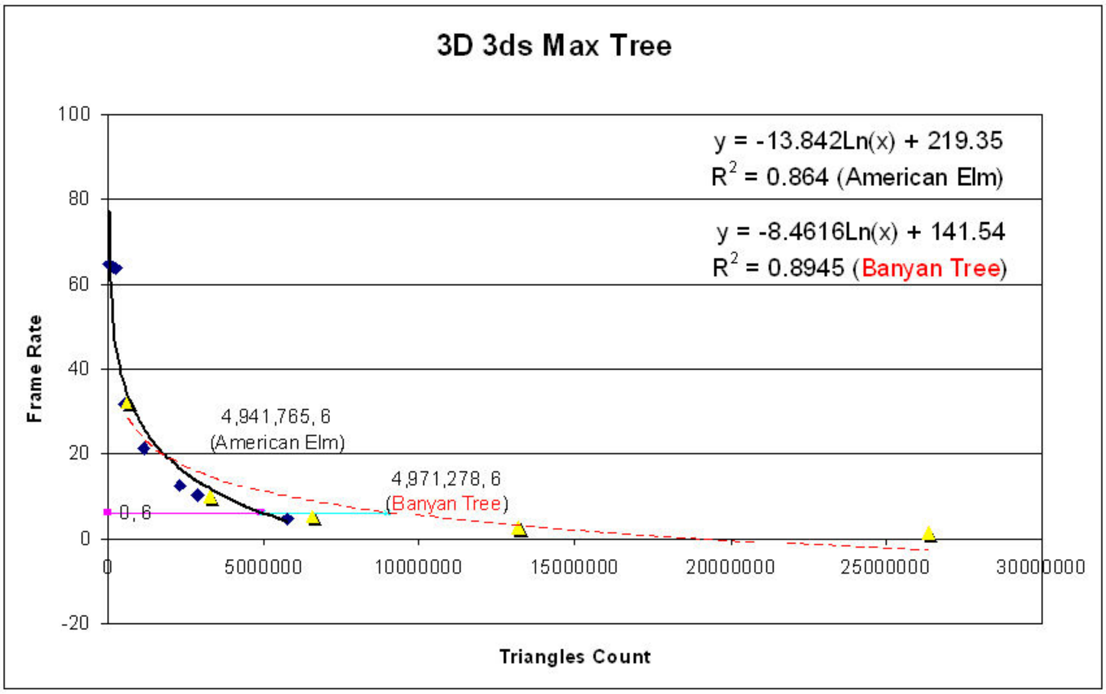


Figure 4.4. 3D 3ds Max® Tree 10, 50, 100, 200 and 400 Trees

(ii) 2-plane and 4-plane billboard trees

Two and four surfaces crisscrossed with different complexity are used in the experiment as shown in Figure 4.5 and 4.6. This method uses a tree image applied to surfaces or planes which are commonly called billboards.

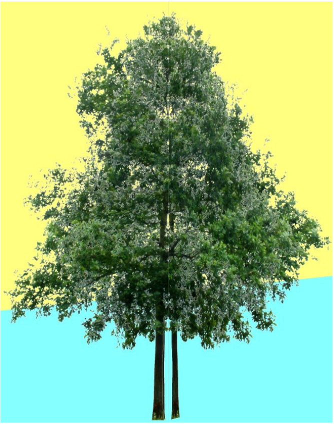


Figure 4.5. Tree A 4-Plane (Billboard Tree)

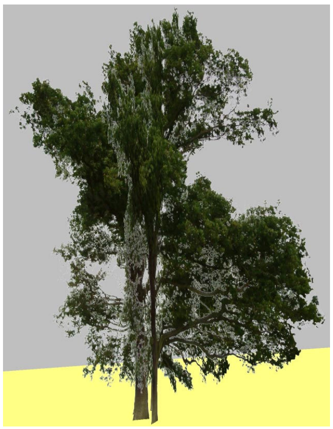


Figure 4.6. Tree B 4-Plane (Billboard Tree)

The graph in Figure 4.7 shows the trees in triangles count plotted against frame rate. As the amount increases from 10, 50, 100, 400, 800 and 1,600 trees, the frame rate drops significantly. At frame rate 6, the limit is 646 number of Tree A 4-Plane (26,339 triangles), 592 number of Tree B 4-Plane (24,142 triangles), 960 number of Tree A 2-Plane (23,816 triangles) and 781 number of Tree B 2-Plane (19,357 triangles). The formula enables the prediction of frame rate (Y) with the amount of triangles known.

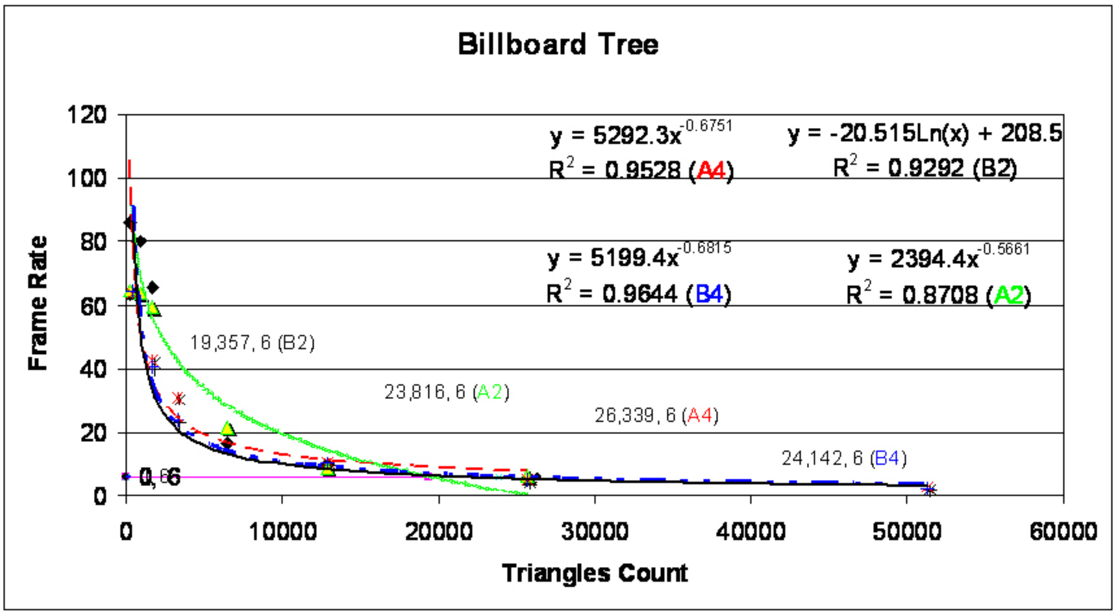


Figure 4.7. Billboard Tree 10, 50, 100, 400, 800 and 1,600 Trees

(iii) 3D trees with billboard leaves

Three trees of different complexity were used in the experiment, with two of them shown in Figure 4.8 and 4.9. The trunks and branches were in 3D while only the leaves were flat surfaces applied with images of leaves which are normally repeated all over the trees.

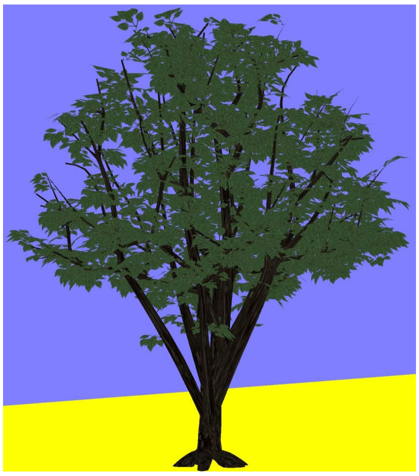


Figure 4.8. Bush (3D Tree + Billboard Leaves)

Figure 4.9. Leaf (3D Tree + Billboard Leaves)

The graph in Figure 4.10 shows the trees in triangles count plotted against frame rate. As the amount increases from 10, 50, and 100 trees, the frame rate drops significantly. At frame rate 6, the limit is 43 number of Bush trees (1,890,550 triangles), 22 number of Leaf trees (1,832,374 triangles) and 58 number of Palm trees (1,804,856 triangles). The formula enables the prediction of frame rate (Y) with the amount of triangles known.

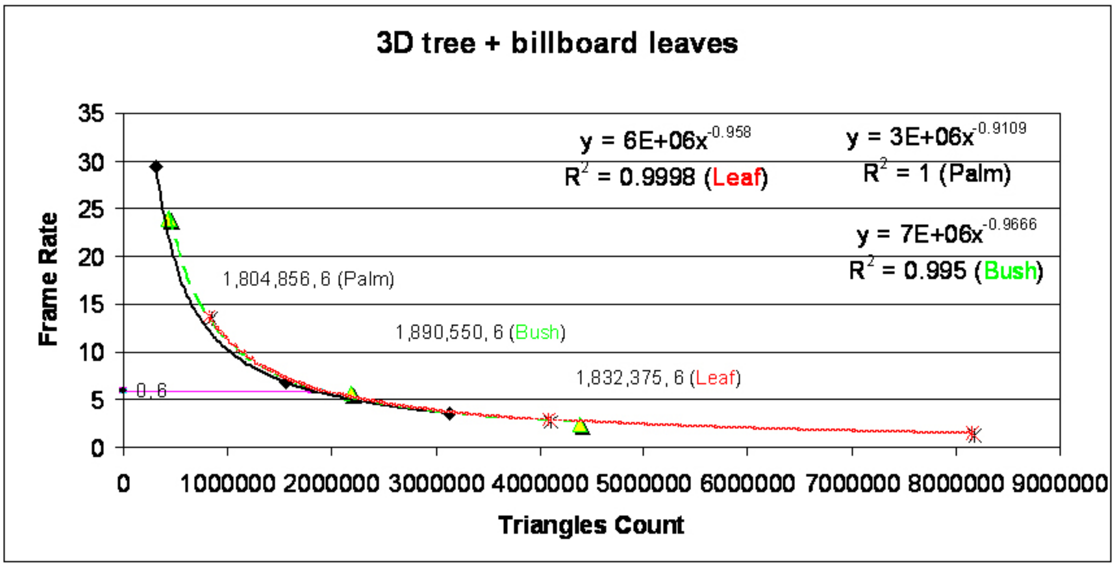


Figure 4.10. 3D Tree + Billboard Leaves 10, 50 and 100 Trees

(iv) Archvision Rich Photorealistic Content (RPC) trees

Three trees of different tree types and complexity were used as shown in Figure 4.11, 4.12 and 4.13. This method uses a series of flat images of the whole tree taken from all 360 degrees and will generate out the correct view based on the position of the camera to the tree.

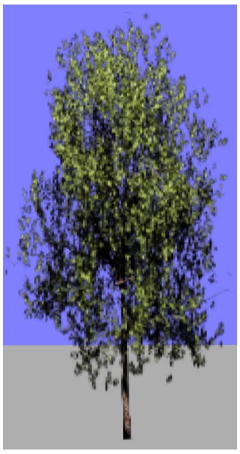


Figure 4.11. Acer Oliverianum RPC

Figure 4.12. Crape Myrtle RPC

Figure 4.13. Golden Malay Palm RPC

The graph in Figure 4.14 shows the trees in triangles count plotted against frame rate. As the amount increases from 10, 50, 100, 200, 400, 800 and 1,200 trees, the frame rate drops significantly. At frame rate 6, the limit is 284 number of Crape Myrtle trees (3,635 triangles), 289 number of Acer Oliverianum trees (3,702 triangles), 311 number of Golden Malay Palm trees (3,975 triangles) and 299 number of Hurricane Palm trees (3,829 triangles). The formula enables the prediction of frame rate (Y) with the amount of triangles known.

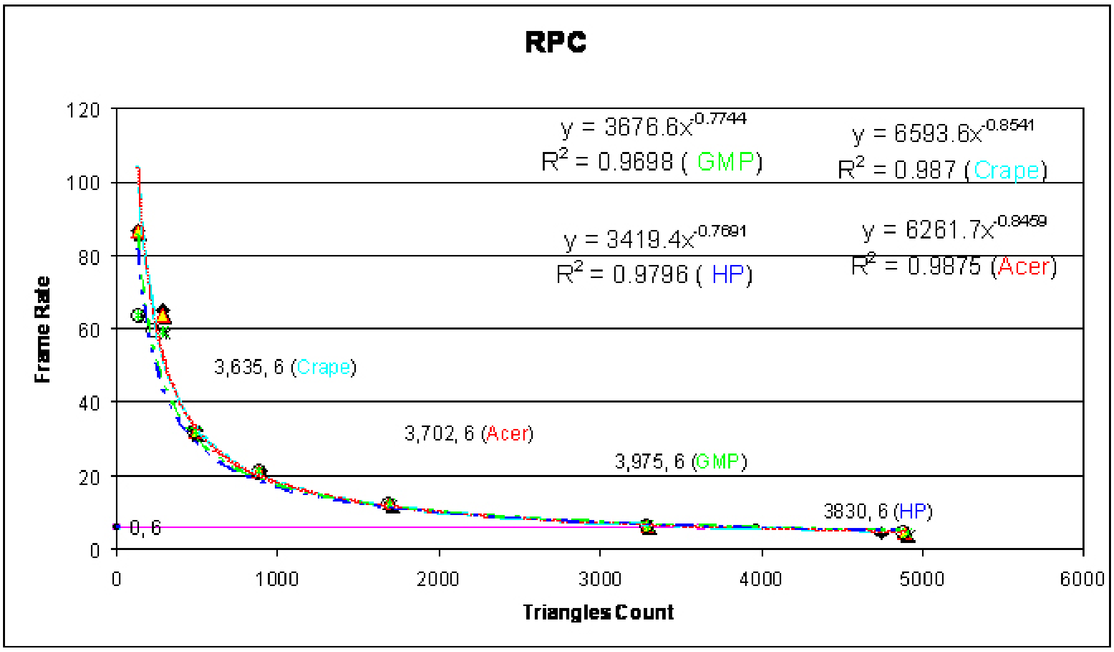


Figure 4.14. RPC 10, 50, 100, 200, 400, 800, 1,200 Trees

The time taken for travelling the same distances regardless of the increase of trees will not change. Table 4.1 proves that no matter the quantity of trees or tree types, the time taken is roughly between 33.5 - 33.6 seconds. The experiment setup is shown in Figure 4.15 and 4.16. Therefore, the time taken can be ignored and dropped from any further measurement. This has proven one of the hypotheses wrong. The hypothesis is that time taken to cover a distance is different depending on the complexity of the scene. It is not true and the only difference is the drop in frame rate. This drop makes the simulation feel increasingly slower as the complexity increases.

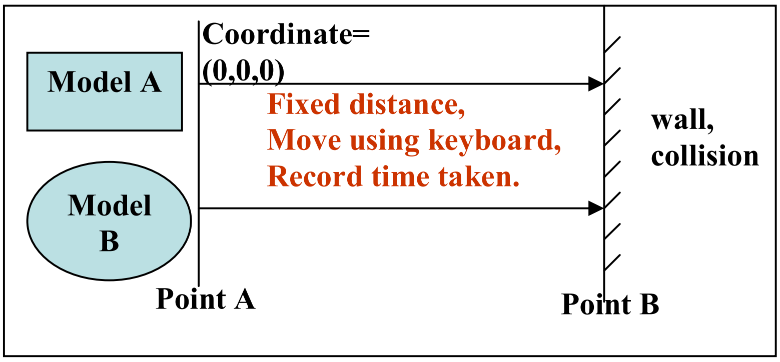


Figure 4.15. Time Taken to Cover the Distance from Point A to Point B

TABLE 4.1. Time Taken For The Same Distance Travelled

Tree Types:	**Time Taken (seconds)**	**Tree Types:**	**Time Taken (seconds)**
400 Golden Malay Palm	33.5	10 Banyan Tree	33.5
800 Golden Malay Palm	33.5	100 Banyan Tree	33.5
1200 Golden Malay Palm	33.5	200 Banyan Tree	33.6
400 Hurricane Palm	33.6	400 Banyan Tree	33.6
800 Hurricane Palm	33.6	400 Tree1 2 Planes	33.5
1200 Hurricane Palm	33.5	1600 Tree1 2 Planes	33.5
400 Acer Oliverianum	33.6	400 Tree2 2 Planes	33.5
800 Acer Oliverianum	33.6	1600 Tree2 2 Planes	33.5
1200 Acer Oliverianum	33.6	400 Tree1 4 Planes	33.6
10 American Elm	33.5	1600 Tree1 4 Planes	33.5
100 American Elm	33.5	400 Tree2 4 Planes	33.6
2000 American Elm	33.5	1600 Tree2 4 Planes	33.5

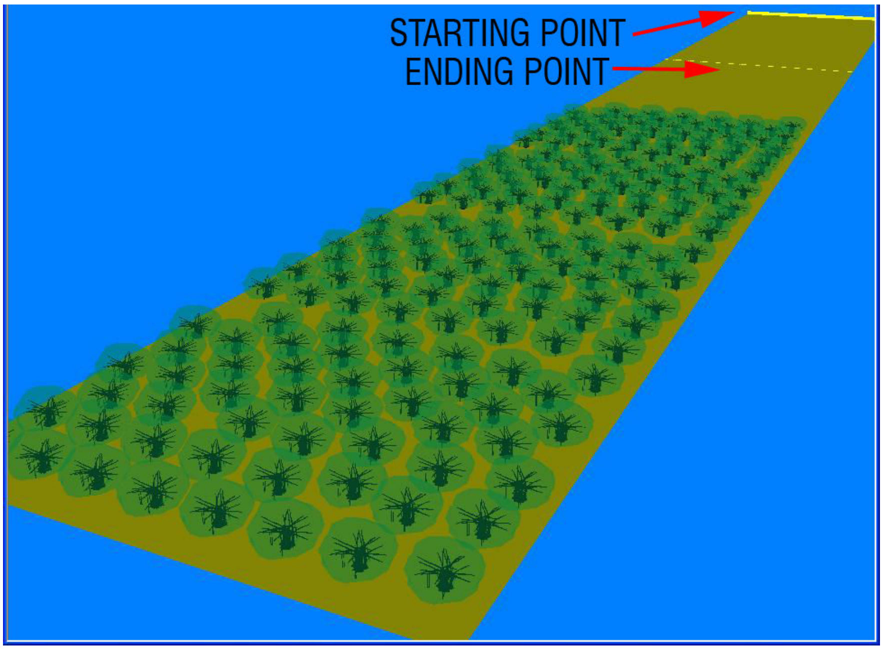


Figure 4.16. Starting and Ending Point of Navigation

From the equation gathered from each method and tree ranges, at the threshold frame rate of 6 (the minimum requirement for real-time), the number of triangles can be identified. From the average taken, the maximum number of trees in each method that a scene can support at frame rate of 6 can be found.

(i) 3D Banyan Tree -75 trees

(ii) 3D American Elm - 1708 trees

(iii) 2-Plane Billboard tree A - 1326 trees

(iv) 4 Plane Billboard tree A -775 trees

(v) 2-Plane Billboard tree B -1077 trees

(vi) 4 Plane Billboard tree B 4 - 711 trees

(vii) 3D + billboard Bush - 43 trees

(viii) 3D + billboard Palm - 58 trees

(ix) 3D + billboard Leaf - 22 trees

(x) Crape Myrtle RPC - 633 trees

(xi) Acer Oliverianum RPC - 643 trees

(xii) Golden Malay Palm RPC - 690 trees

(xiii) Hurricane Palm RPC - 686 trees

From the methods analyzed, a few things can be concluded:

(a) 3D trees with billboard leaves are definitely the most volumetric of all and therefore are suitable for close range navigation. However, because of its complexity, this kind of tree should not be used extensively in a scene. From the experiments, it can be easily deduced that any number between 22 to 58 trees will give the borderline frame rate of 6. Therefore, one should use this method wisely in a scene where detail is essential.

(b) 3D 3ds Max® tree is a sketch-like tree shape representation. This kind of tree is more suitable for a symbolic representation and therefore has the least realism. It will be good for an urban design site where the details are not so critical. From the results, they can be in the range of 75-1708 trees depending on the tree type and can be selected depending on the size and scale of the urban site.

(c) Billboard trees of 2 planes and 4 planes are considered to be the most basic tree versions. Therefore, a scene can accommodate a large number of them before the frame rate drops significantly. It can accommodate easily between 1077-1326 numbers of 2-

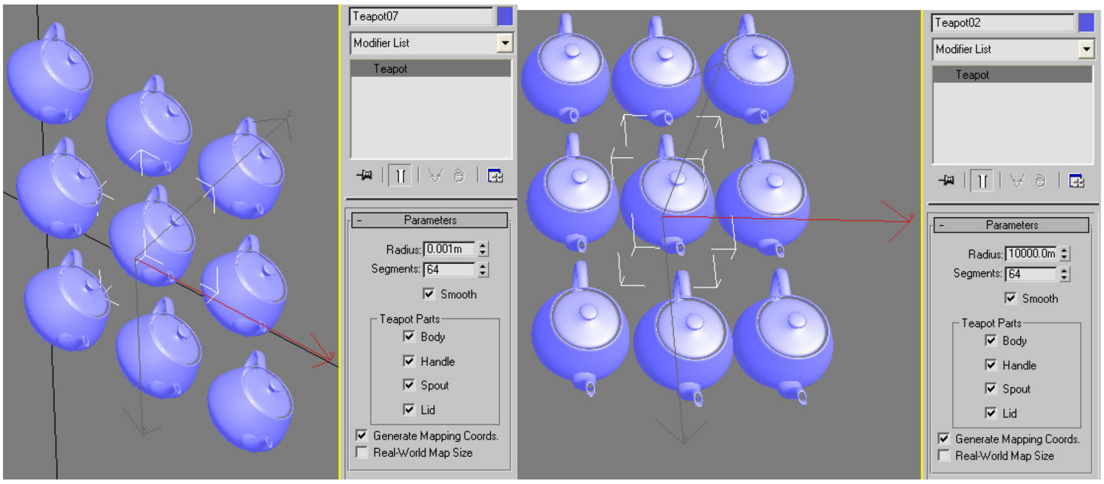
plane trees and between 711-775 numbers of a 4-plane trees. This type of tree is good for background planting as well as long-range backdrop.

(d) RPC trees are the most surprising of them all. Before the experiment, it was expected that the RPC will not be able to perform close to the billboard trees as each tree is represented by 95 textures taken from a rotation of 360 degrees around it. However, it was proven wrong as up to 633 - 690 trees can be inserted. RPC can be very detailed so they are suitable for close range to background tree planting without degradation of real-time performance. It is not a popular choice of usage because its library content is very limited and will require a lot of work by the user to create the RPC file themselves. All the library contents need to be purchased.

With these experiment results, graphs and equations, the correct method of tree representation can be used, based on the requirements of different projects. The estimation of how many trees and the tree method to use can be done with the knowledge of the number of triangles and required frame rate of a project before any trees are inserted.

4.1.2 Triangle Size

The experiment looks at how different triangle sizes affect frame rate. Nine teapots with radius lengths varying from 0.0001 metres to 10,000 metres were constructed. The increase in scale means the triangle size is increased. Each teapot was increased in size for every scale level increase consistently. Figure 4.17 shows the two different sets of teapot created at 0.0001m in radius on the left and 10,000m in radius on the right.



Teapot07

Radius: 0.001m
Segments: 64
☑ Smooth
Teapot Parts
☑ Body
☑ Handle
☑ Spout
☑ Lid
☑ Generate Mapping Coords.
☐ Real-World Map Size

Teapot02

Radius: 10000.0m
Segments: 64
☑ Smooth
Teapot Parts
☑ Body
☑ Handle
☑ Spout
☑ Lid
☑ Generate Mapping Coords.
☐ Real-World Map Size

Figure 4.17. 9 Teapots from 0.0001m to 10,000m in radius

It is proven that the frame rate is consistent regardless of the increase in triangle size. The frame rate of 29.1 Hz is maintained from nine teapots of 0.0001m in radius to 10,000m in radius. Table 4.2 shows a consistent result regardless of the difference in teapot radius. Therefore, it is concluded that triangle size does not affect the performance of VR visualisation.

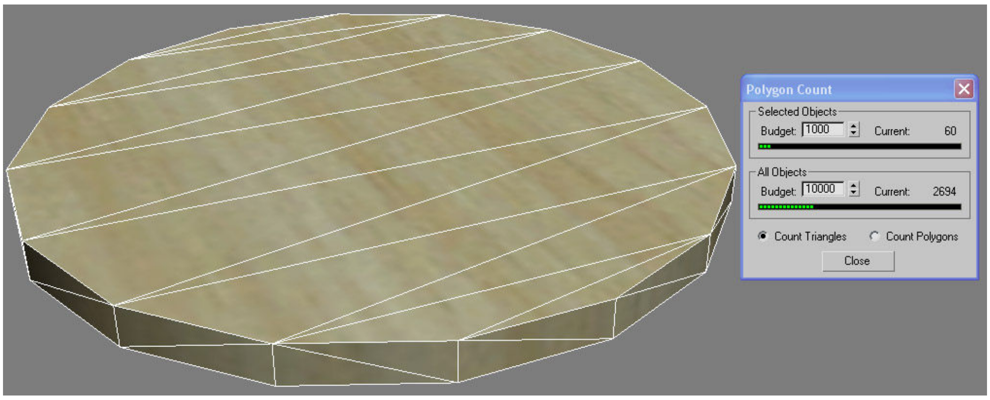
TABLE 4.2. Variation of Triangle Size

Teapot Radius (m) (triangle size)	**Triangles Count**	**Vertex Memory (KB)**	**Frame Rate (Hz)**
0.0001	2,359,296	27,702	29.1
0.001	2,359,296	27,702	29.1
0.01	2,359,296	27,702	29.1
0.1	2,359,296	27,702	29.1
1	2,359,296	27,702	29.1
10	2,359,296	27,702	29.1

100	2,359,296	27,702	29.1
1,000	2,359,296	27,702	29.1
10,000	2,359,296	27,702	29.1

4.1.3 Triangle Count

A simple square or rectangular surface is made up of two triangles. This is how computers calculate surfaces and this is how they are generated. The more triangles a surface has, the more details can be placed in it. However, it will also mean that more computer processing power is needed. A good example to show how surfaces are represented by triangles is a chair as shown in Figure 4.19. The seat of the chair in Figure 4.18 has 14 triangles at the top, 32 triangles at the side and 14 triangles at the bottom. Therefore, the chair seat is constructed of 60 triangles.



Budget: 1000 Current: 60

Budget: 10000 Current: 2694

● Count Triangles ○ Count Polygons

Figure 4.18. Chair seat

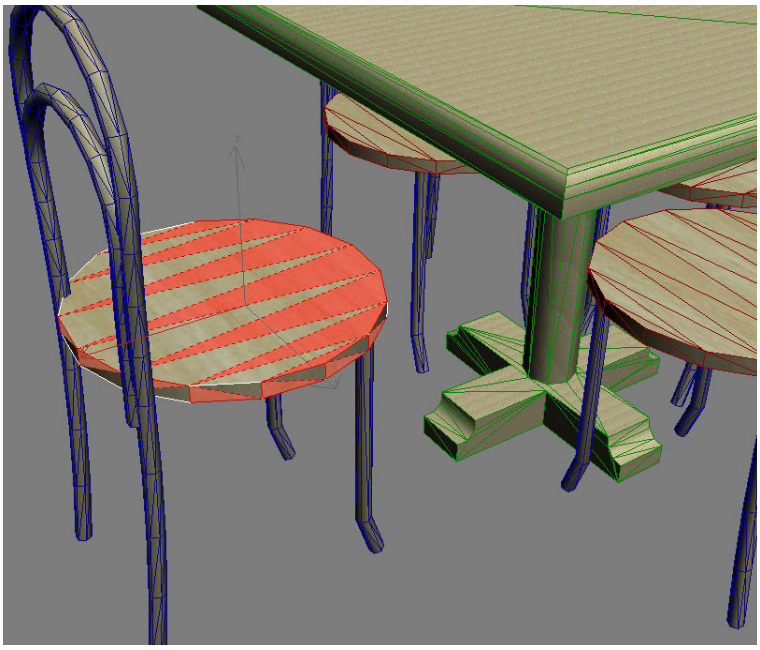


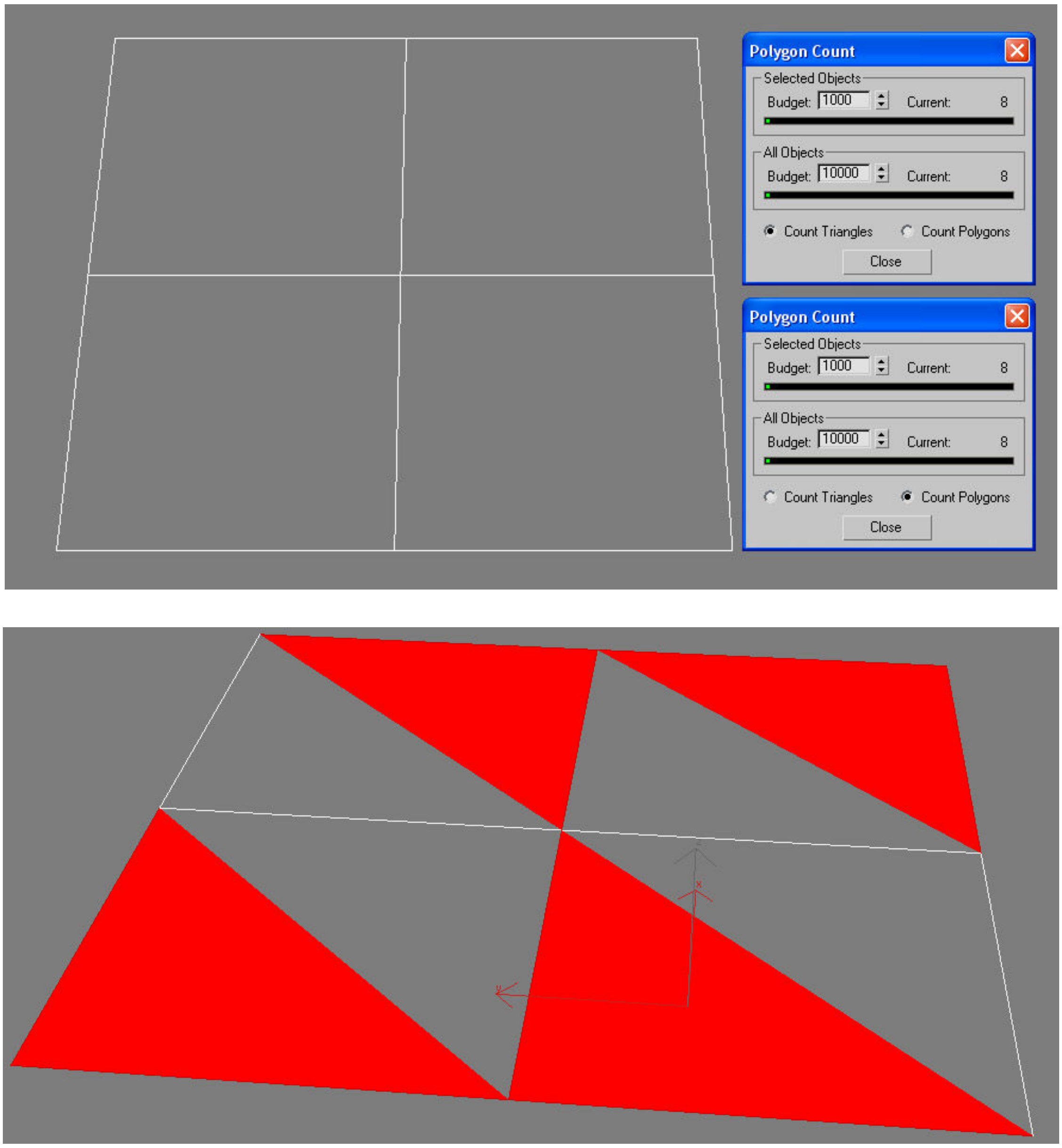
Figure 4.19. Chair

There is always confusion arising from the difference between triangles and polygons.

> "It may interest you to know why polygons are generally defined as triangles in common rendering architectures. First of all, triangles are the simplest shape that can be defined with an actual surface area, and thus it serves as a fundamental building block for the creation of meshes. Although you could use more complex polygons, and some unusual architecture do, you are bound to encounter two problems. The first significant issue comes from the fact that by using a more complex shape as your fundamental primitive, such as a square, you may not be able to define more complex shapes as accurately. In addition, the mathematics for rasterizing more complex polygons, which mean determining the pixel coverage of a polygon, become much more complex because you would have to deal with non-convex polygons."

(Sebastien2004)

A simple test was done in 3ds Max® to confirm the similarities between triangle and polygon count. A plane was constructed as shown in Figure 4.20 to have eight triangles or eight polygons.



Selected Objects
Budget: 1000 Current: 8
All Objects
Budget: 10000 Current: 8
● Count Triangles ○ Count Polygons

Selected Objects
Budget: 1000 Current: 8
All Objects
Budget: 10000 Current: 8
○ Count Triangles ● Count Polygons

Figure 4.20. Triangle and Polygon Comparison

No matter what the 3D models look like, they are primarily constructed from triangles. Therefore, it does not matter whether it is a tree or a piece of furniture or a vehicle or an architecture element, they basically consist of triangles / polygons in the computer. Thus, the number of triangles matters in the experiment rather than the form of the 3D objects. The computer will just basically treat them as triangles. However, since this research is in the field of architecture, all 3D models used are related to architecture.

This experiment showed that the increase of triangles does not translate to increase use of vertex memory. The increase of triangles count does not increase the use vertex memory much, from 3342 Kb to 4171 Kb only, which translates to only 19.93% increase. The frame rate however, drops significantly to below 0.9 for the largest number of triangles. That is a huge drop of 96.96%. The graph in Figure 4.21 shows that at 818,958 triangles, the frame rate will reach 6 fps. The formula enables the prediction of frame rate (Y) with the amount of triangles known.

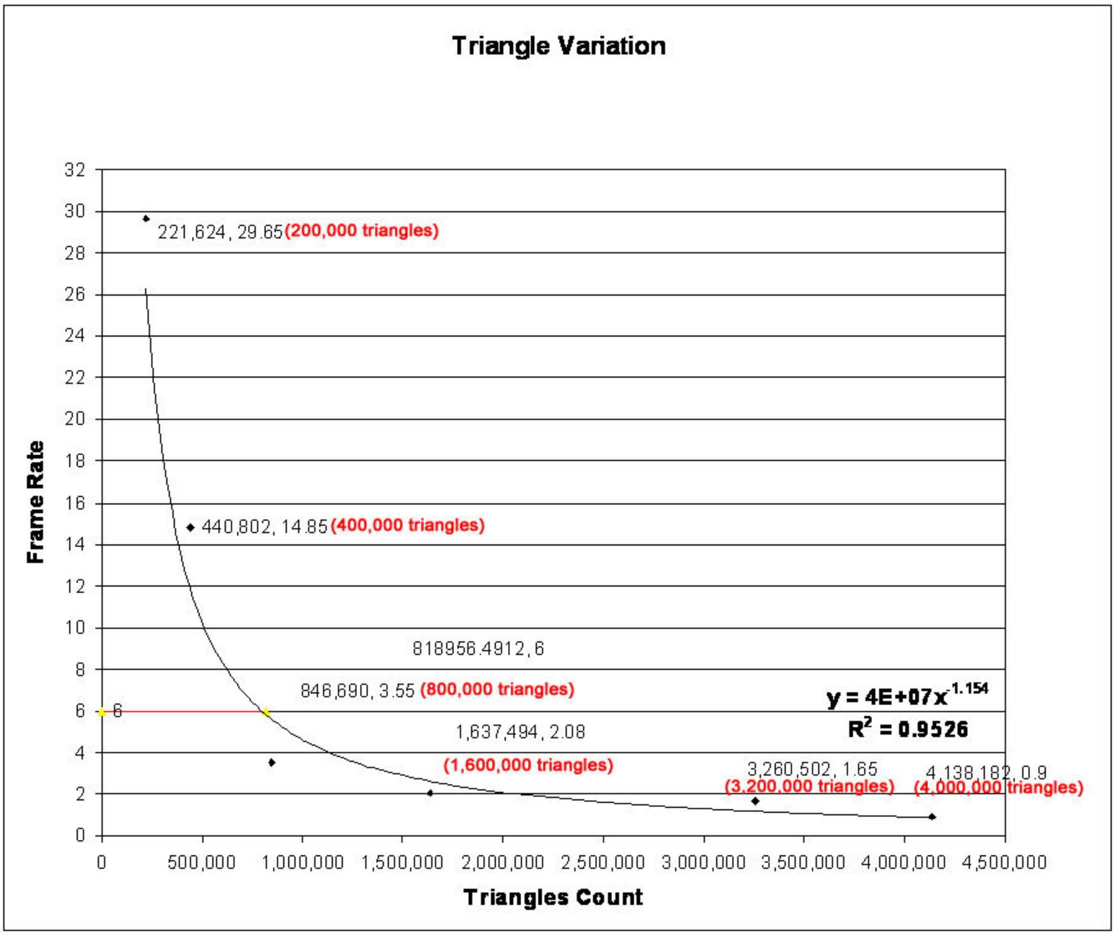


Figure 4.21. Triangle Count (200,000, 400,000, 800,000, 1,600,000, 3,200,000, 4,000,000 triangles)

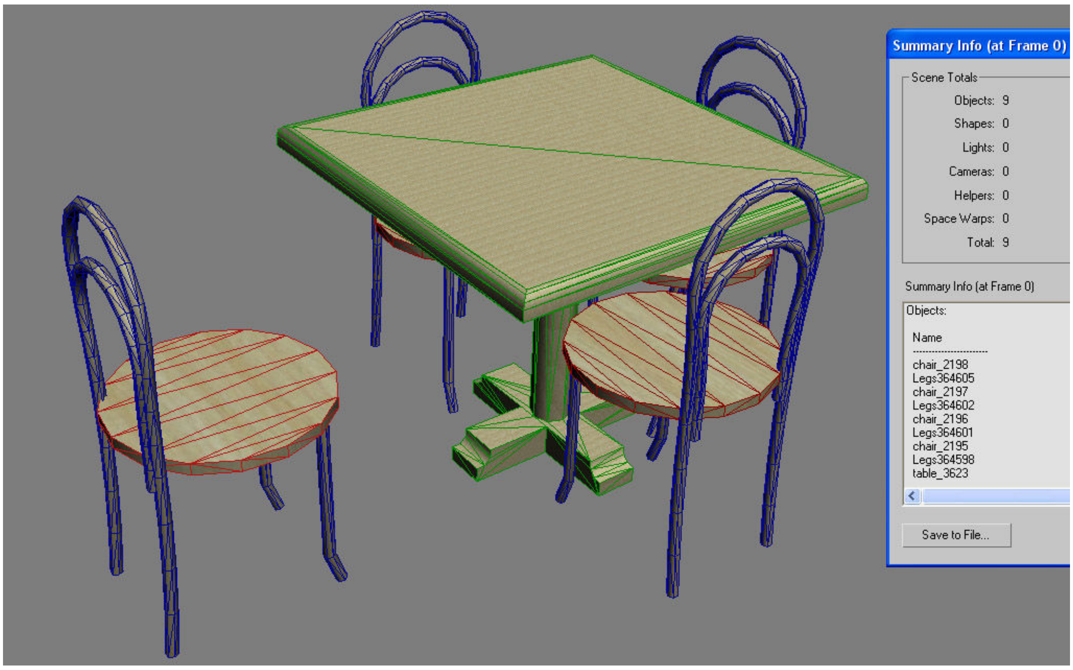
TABLE 4.3 . Triangle Count with Vertex Memory

Models	**Tri 200k**	**Tri 400k**	**Tri 800k**	**Tri 1600k**	**Tri 3200k**	**Tri 4000k**
Vertex Memory (KB)	3342	4169	4174	4171	4174	4174

It is clear that triangle count is very much related to frame rate. Although vertex memory does increase when triangle count increases as shown in Table 4.3, it is too minimal to give much impact to the amount of memory available in the graphic card and system memory. Therefore, it is concluded that it does not affect performance. Most graphic cards nowadays have 256 – 512MB memory and most system memory possesses around 2GB – 4GB RAM.

4.1.4 Geometry Count

A 3D model can be divided into many parts, most of the time depending on different textures used. These parts are called geometry or sometimes called object in some software. For example, the chair is textured with a timber for the seat and steel material for the frames and legs. The table shown in Figure 4.22 comes in just one timber texture and therefore, there is no reason to subdivide it into more parts. Multi texture method can be sometimes used in less complicated 3D models in an effort to avoid dividing a 3D model into too many parts. In Figure 4.22, each chair has two parts, sometimes called objects or geometries. Hence, the whole set has nine objects altogether, taking into account that each of the four chairs is divided into two objects while one more is contributed by the whole table.



```
Scene Totals
        Objects: 9
         Shapes: 0
         Lights: 0
        Cameras: 0
        Helpers: 0
    Space Warps: 0
          Total: 9

Summary Info (at Frame 0)
Objects:

 Name
 ----------------------
 chair_2198
 Legs364605
 chair_2197
 Legs364602
 chair_2196
 Legs364601
 chair_2195
 Legs364598
 table_3623
```

Figure 4.22. Chair = 2 objects (2 textures), Table = 1 object, TOTAL = (2x4)+1 = 9 Objects

This experiment was done to determine whether the increase of geometry will affect the frame rate as well as the vertex memory. It showed that an increase in the number of geometries will increase the vertex memory used. The increase from 50 to 3200 geometries in the scene increases the vertex memory from 807 Kb to 4174 Kb, which is an increase of 80.67%. The frame rate for the number of geometries from 50 to 3,200 decreased from 64.5 to 6.25, a drastic drop of 90.31%. The graph in Figure 4.23 shows that at 1,216,542 triangles, the frame rate will reach 6fps. The formula enables the prediction of frame rate (Y) with the amount of geometries known.

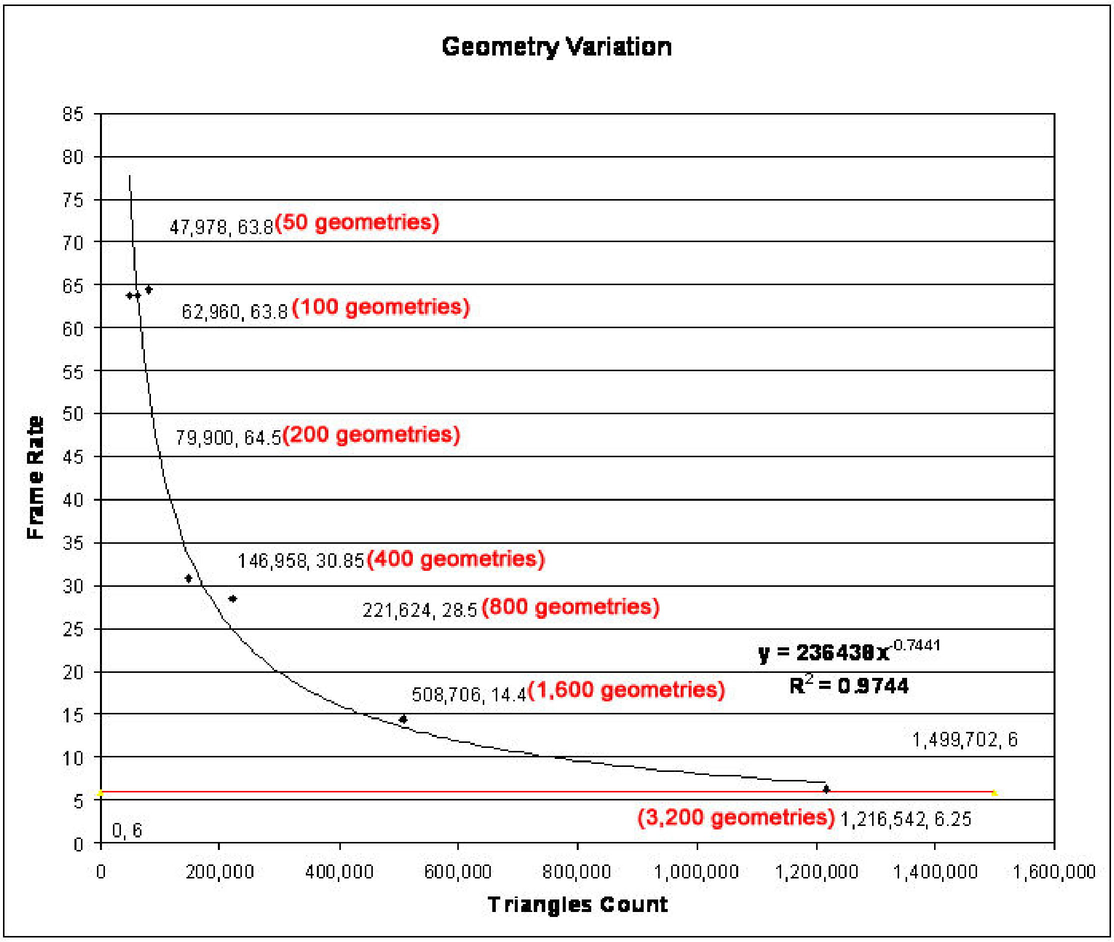


Figure 4.23. Geometry Count (50, 100, 200, 400, 800, 1,600, 3,200 geometries)

TABLE 4.4. Geometry Count with Vertex Memory

Models	**Geo 50**	**Geo 100**	**Geo 200**	**Geo 400**	**Geo 800**	**Geo 1600**	**Geo 3200**
Vertex Memory (KB)	807	1061	1345	2292	3342	4174	4174

It is clear that geometry count is very much related to frame rate. Similar to triangle count, vertex memory does increase when geometry count increases as shown in Table 4.4. The increase again is too minimal to give much impact to the amount of memory available in the graphic card and system memory. Therefore, it can be concluded that it does not affect performance. In comparison to triangle count, the increase of geometry in the scene will require more memory than the increase of triangle counts. Since cloning is used for many copies of the same parts of the 3D models, a lot of vertex memory can be saved.

4.1.5 Texture Count

Experiments were conducted to determine whether the number of textures used in the scene affect the frame rate. The table in Figure 4.5 shows application of textures ranging from 10 to 100 textures to 3D models ranging from 100,000 to 2,000,000 triangles. The frame rate difference from 10 to 100 textures is only 0.9 for 100,000 triangles, 1.9 for 500,000 triangles, 2.8 for 1,000,000 triangles and 0.1 for 2,000,000 triangles. The overall difference of frame rate drop from 0.1 to 2.8 is not very significant. In other words, an increase in the number of textures used in the scene will not affect frame rate much. It will however, result in an increase in texture memory and vertex memory used.

TABLE 4.5. Texture Count (10, 50, 100 textures)

Models \ Details	**vertex memory (KB)**	**texture memory (KB)**	**triangle count**	**frame rate**
Tri 100k with 10 textures	4,650	32	1,887,142	21.2
Tri 100k with 50 textures	4,892	8,228	221,624	21.4
Tri 100k with 100 textures	4,925	13,754	221,624	20.3
Tri 500k with 10 textures	6,622	32	995,650	2.5
Tri 500k with 50 textures	7,481	9,039	1,009,230	3.5
Tri 500k with 100 textures	10,449	13,853	993,400	4.4
Tri 1000k with 10 textures	7,552	32	1,887,142	3.5

Tri 1000k with 50 textures	17,713	8,952	2,422,854	2.4
Tri1000k with 100 textures	24,315	15,133	1,907,084	0.7
Tri 2000k with 10 textures	6,275	32	4,135,686	1
Tri 2000k with 50 textures	17,713	10,360	4,259,430	0.9
Tri 2000k with 100 textures	N/A	N/A	N/A	N/A

*N/A because software crashed

It is clear that texture count does affect performance as much as triangle count and geometry count. Even though the consumption of memory, especially texture memory does increase by a lot, it is still too minimal to affect the total available in the graphics memory and system memory. Therefore, there is no need to worry about them and both of them can be discarded in the final experiment.

4.1.6 Texture Resolution

The texture resolution test was done to understand how much impact it has on the final performance. The experiment was done using DDS file format in the resolution quality of 128 X 128, 256 X 256, 512 X 512, 1024 X 1024, 2048 X 2048 and 4096 X 4096 and the frame rate and memory consumed are shown in Table 4.6.

TABLE 4.6. Texture Resolution (128 X 128 pixel to 4096 X 4096 pixel)

Resolution: 128		**Resolution: 1024**	
FR	**12.9**	**FR**	**12.7**
TRI	**347,832**	**TRI**	**347,832**
TM	**290**	**TM**	**330**

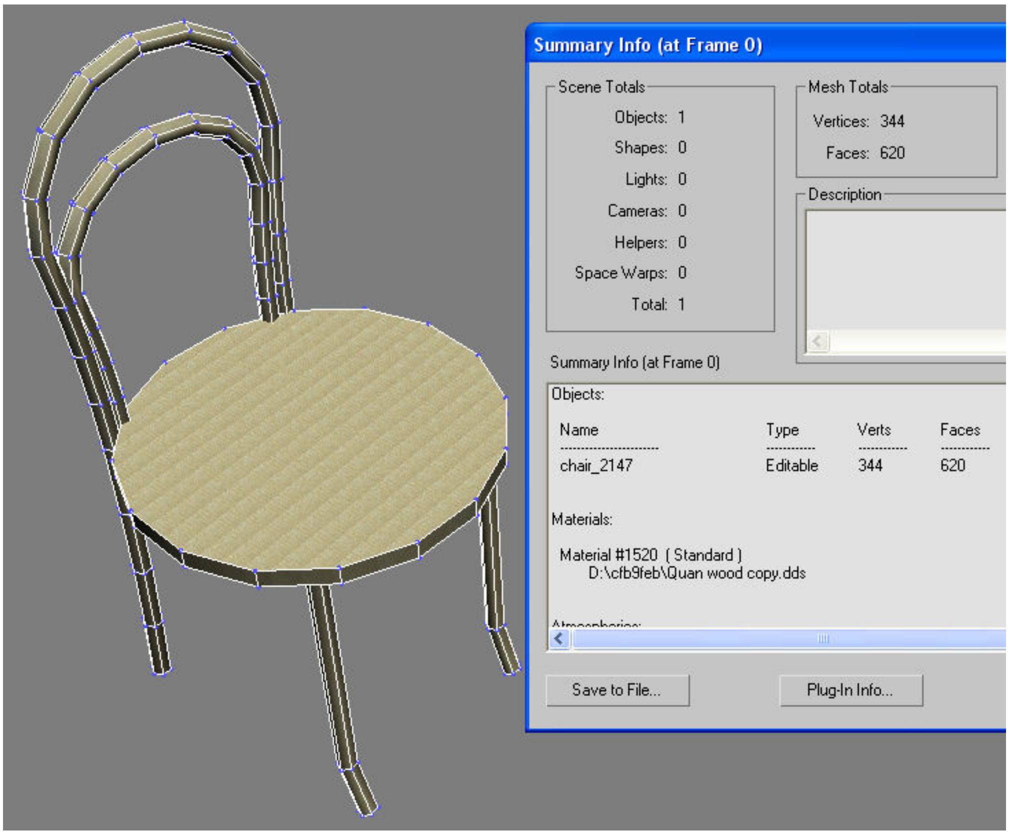
VM	6,467	VM	6,469
Resolution: 256		Resolution: 2048	
FR	12.8	FR	12.5
TRI	347,832	TRI	347,832
TM	170	TM	1,098
VM	6,469	VM	6,469
Resolution: 512		Resolution: 4096	
FR	12.8	FR	12.4
TRI	347,832	TRI	347,832
TM	68	TM	3,018
VM	6,469	VM	6,469

(FR: Frame Rate, TRI: Triangle, TM: Texture memory, VM: Vertex memory)

It is clear that texture resolution does not impact performance much as shown in Table 4.6. However, it is just one texture, and in a real architectural visualisation; it could reach up to 50 textures. Therefore, a survey is done later in the research to understand what the human acceptance of texture resolution is so that a good management of using them correctly can be done.

4.1.7 Vertex Count

Each basic triangle has 3 vertices for each corner. Normally, more vertices are added to a simple surface to make the shaping of it easier, especially for organic models. More vertices around a surface will give the user more ability to shape it at any preferred angle without it looking too triangulated. The chair frame in Figure 4.24 has vertices at positions where the curving or bending takes place for the frames as well as the chair seat. If more vertices are present, the curve and bending can be shaped much more smoothly. However, like triangles count and geometries count, more vertices count will mean slower performance.



Objects: 1
Shapes: 0
Lights: 0
Cameras: 0
Helpers: 0
Space Warps: 0
Total: 1

Vertices: 344
Faces: 620

```
Objects:

 Name                      Type        Verts       Faces
 -----------------------   ----------- ----------- -----------
 chair_2147                Editable    344         620

Materials:

 Material #1520  ( Standard )
      D:\cfb9feb\Quan wood copy.dds
```

Figure 4.24. Vertices Count for Chair

After predicting the amount of triangles at 6 frames per second for both triangle count and geometry count, the vertices count test was done just to re-confirm the relationship between frame rate and vertices count. It is clear that the relationship is not linear as the best fit equation proves. The graph in Figure 4.25 shows that frame rate drops significantly as vertices count increases. The formula enables the prediction of frame rate (Y) with the amount of vertices known.

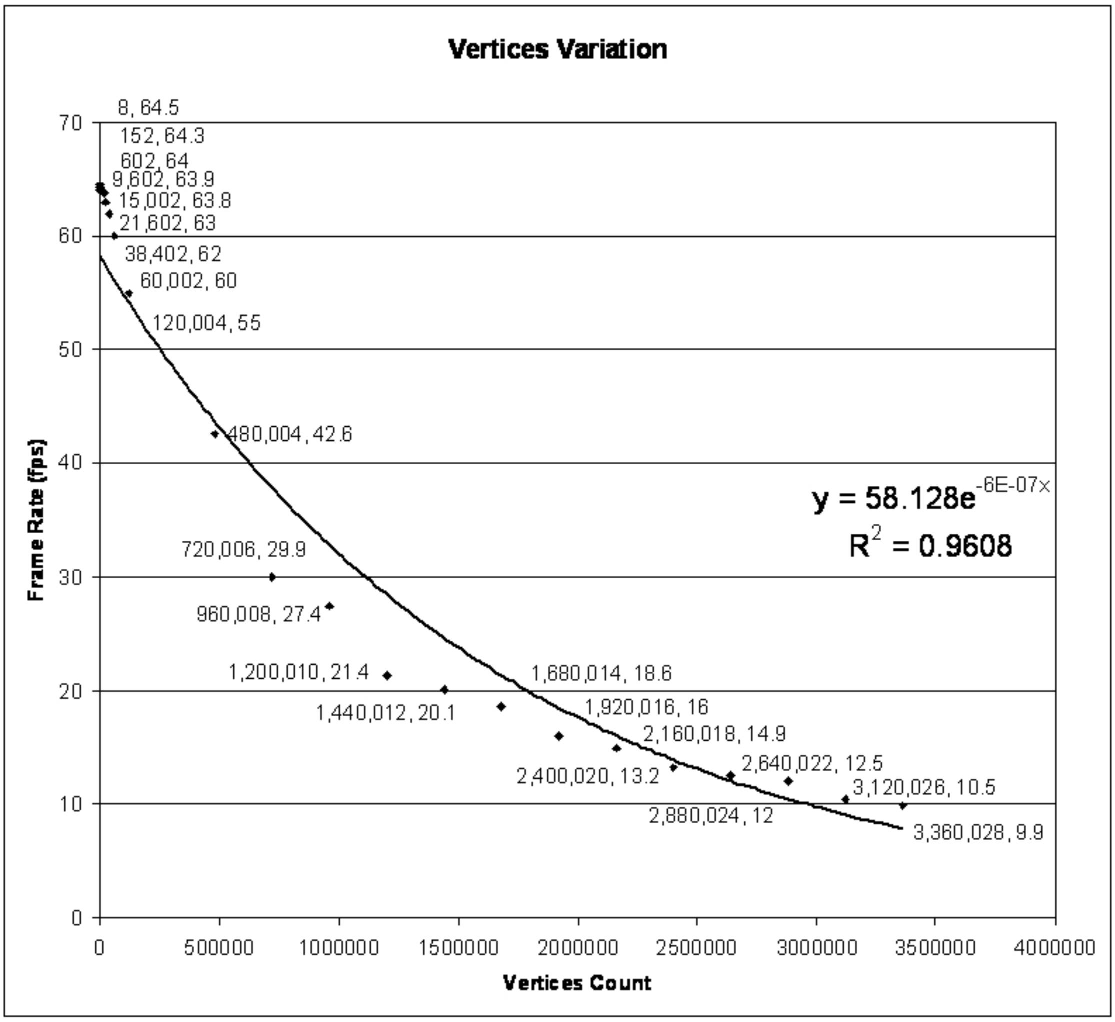


Figure 4.25. Vertices Count (8, 152, 602, 9602, 15002, 21602, 38,420, 60,002, 120,004, 480,004, 720,006, 960,008, 1,200,010, 1,440,012, 1,680,014, 1,920,016, 2,160,018, 2,400,020, 2,640,022, 2,880,024, 3,120,026, 3,360,028 vertices)

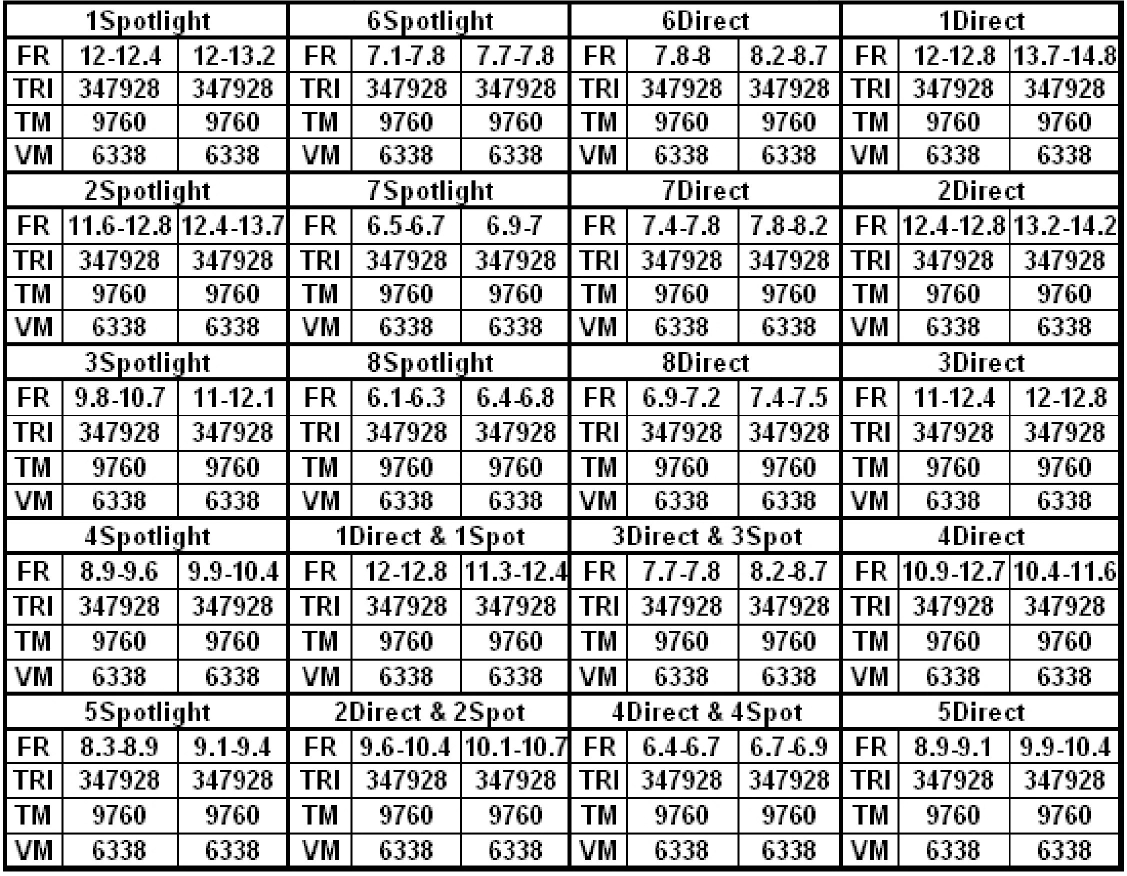
4.2 Secondary Variables

Secondary variables, also known as second generation variables, exist because of the advancement in graphics technology. They are additional features which can be used to enhance VR visualisations to make them far more realistic. Their impact on performance will be explored to better understand how to use them efficiently.

4.2.1 Light Count and Particle Systems

The test for light count is to push the limits of using eight lights in a simulation. Therefore, an experiment was done to test the combination of direct lights and spot lights in the simulation. The computer calculates lighting at vertices of a triangle and then interpolates over it depending on the colours applicable.

TABLE 4.7. Light Count Combinations in still and navigation mode



1Spotlight			6Spotlight			6Direct			1Direct		
FR	12-12.4	12-13.2	FR	7.1-7.8	7.7-7.8	FR	7.8-8	8.2-8.7	FR	12-12.8	13.7-14.8
TRI	347928	347928	TRI	347928	347928	TRI	347928	347928	TRI	347928	347928
TM	9760	9760	TM	9760	9760	TM	9760	9760	TM	9760	9760
VM	6338	6338	VM	6338	6338	VM	6338	6338	VM	6338	6338
2Spotlight			7Spotlight			7Direct			2Direct		
FR	11.6-12.8	12.4-13.7	FR	6.5-6.7	6.9-7	FR	7.4-7.8	7.8-8.2	FR	12.4-12.8	13.2-14.2
TRI	347928	347928	TRI	347928	347928	TRI	347928	347928	TRI	347928	347928
TM	9760	9760	TM	9760	9760	TM	9760	9760	TM	9760	9760
VM	6338	6338	VM	6338	6338	VM	6338	6338	VM	6338	6338
3Spotlight			8Spotlight			8Direct			3Direct		
FR	9.8-10.7	11-12.1	FR	6.1-6.3	6.4-6.8	FR	6.9-7.2	7.4-7.5	FR	11-12.4	12-12.8
TRI	347928	347928	TRI	347928	347928	TRI	347928	347928	TRI	347928	347928
TM	9760	9760	TM	9760	9760	TM	9760	9760	TM	9760	9760
VM	6338	6338	VM	6338	6338	VM	6338	6338	VM	6338	6338
4Spotlight			1Direct & 1Spot			3Direct & 3Spot			4Direct		
FR	8.9-9.6	9.9-10.4	FR	12-12.8	11.3-12.4	FR	7.7-7.8	8.2-8.7	FR	10.9-12.7	10.4-11.6
TRI	347928	347928	TRI	347928	347928	TRI	347928	347928	TRI	347928	347928
TM	9760	9760	TM	9760	9760	TM	9760	9760	TM	9760	9760
VM	6338	6338	VM	6338	6338	VM	6338	6338	VM	6338	6338
5Spotlight			2Direct & 2Spot			4Direct & 4Spot			5Direct		
FR	8.3-8.9	9.1-9.4	FR	9.6-10.4	10.1-10.7	FR	6.4-6.7	6.7-6.9	FR	8.9-9.1	9.9-10.4
TRI	347928	347928	TRI	347928	347928	TRI	347928	347928	TRI	347928	347928
TM	9760	9760	TM	9760	9760	TM	9760	9760	TM	9760	9760
VM	6338	6338	VM	6338	6338	VM	6338	6338	VM	6338	6338

(FR: Frame Rate, TRI: Triangle, TM: Texture memory, VM: Vertex memory)

It is very clear from Table 4.7 that frame rate is affected by the number of lights in the simulation from one to eight lights, but this variable is still not as critical as triangle and geometry count.

As for particle systems, they are small 3D models with a lifespan to generate smoke, gas, water droplets and many others. In the cloud forest biosphere project shown in Figure 4.26, it is used at the bottom of the waterfall to simulate the effect when water hits the lake. By applying the particle systems onto the scene, the amount of frame rate drops by 3Hz.

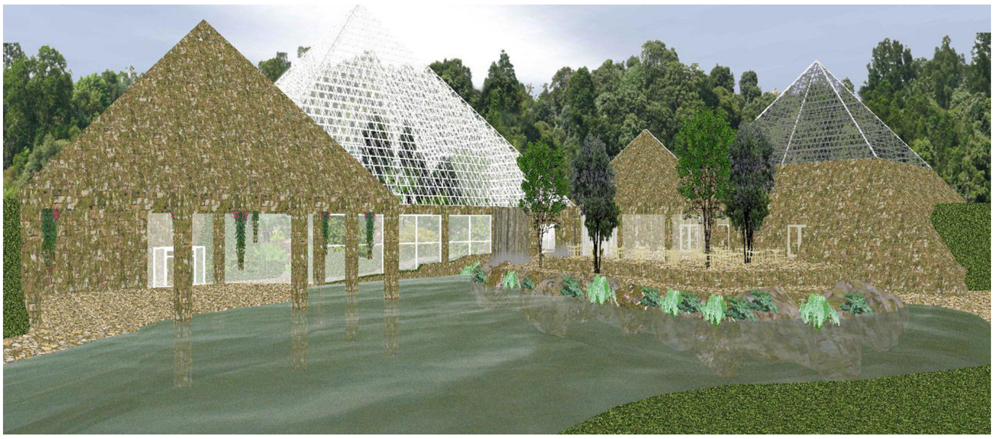


Figure 4.26. Cloud Forest Biosphere with Particle Systems used for the Waterfall

It is also proven that the more particle systems there are in the simulation, the slower the performance will become. A good practice is to always switch off the particle systems if it is not seen by the camera so that it will not calculate over and over again even when it is not seen.

4.2.2 Programmable / Advanced Shaders Count

Traditional shaders have been around for some time but the advanced ones or sometimes called programmable shaders are still in the process of development with new upgrades. They have been used heavily in 3D games with the help of the evolution of the new generation graphics processing unit (GPU). Since the introduction of the first consumer-level 3D hardware accelerated graphics card, the 3Dfx Voodoo in 1995, real-time 3D graphics have become a true reality (Sebastien 2004). By 1999, the ability to perform transform and lighting of vertices on the graphics accelerator became available, hence, advanced shaders were first used then (Möller 2002).

The beauty of advanced shaders is the ability to simulate how the material should look and behave like it is in the real world. Surface qualities such as reflection, refraction, specularity, glossiness, bumpiness and how these surfaces react to light and shadow real-time are given more realism with the use of advanced shaders than with the use of traditional shaders. The traditional shaders such as the flat, Gouraud, Phong, Lambertian and Blinn lack the ability to constantly change in real-time. They merely help bring out the qualities of a material through a flat 2D texture which is basically fixed and dead. These make it impossible for some materials like those with refraction, reflection and water movement to be rendered real-time. Advanced shaders are programmable because changes through time and repetitions can be adjusted. Therefore, it is much more versatile and flexible as well as easy to maintain (Fernando 2004) compared to the traditional shaders. Therefore, they can produce stunningly realistic images (Lastra 1995).

Advanced shaders are, at present, still quite exclusive because they are meant to be designed for specific platforms or specific software only according to the alterations of the programmer. There is a fair amount of work that goes into developing a communication layer that supports shaders (Fernando 2004). Therefore, the process of importing and exporting the advanced shaders from one software to another is quite restricted and limited for an architect who is not a programmer. They are constrained by the number of choices available in the shader library of the software. Even though there are software like AMD Inc. / ATI^{TM} RenderMonkey^{TM} and nVidia® Corp. FX Composer^{TM} in the market to help design and develop shaders, they still require programming skills to transfer them into other programs. They were meant for games in the first place and even games programmers have to accommodate the new architecture in these shaders for next generation games (Penfold 2002).

Advanced shaders are available under the OpenGL^{TM} and DirectX® API (Application Programming Interface). An API functions to communicate with the hardware to tell it which geometry to render and how it is rendered (Sebastien 2004). There are two main types of shaders, namely vertex and pixel shaders. In real-time rendering, vertex shaders manipulate geometry (vertices and their attributes) and pixel shaders manipulate rendered pixels (Luke 2006). Both work hand in hand to generate a shader. They give the developer a lot more freedom to do things that real-time applications have never experienced before (Engel 2002). The Cg (C for Graphics) language is based on both the syntax and the philosophy of C language (Mark 2003). Its rendering is API independent and this allows it to operate under DirectX® or OpenGL™. Figure 4.27 illustrates the relationship of the most famous advanced shader languages in the market currently.

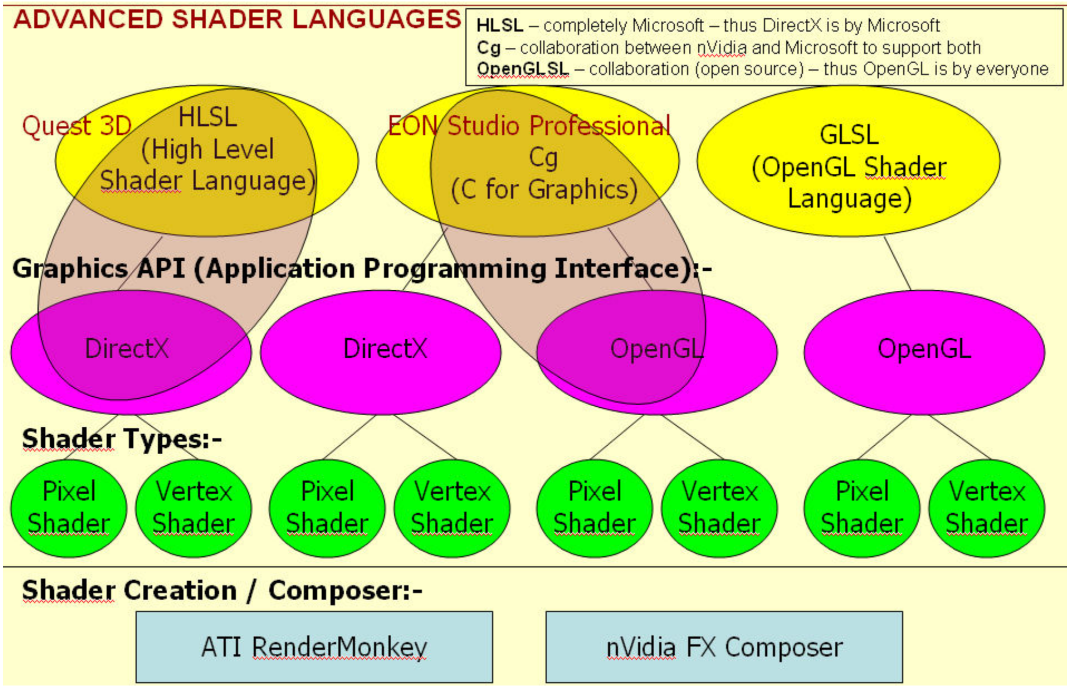


Figure 4.27. Advanced Shader Languages

To demonstrate the difference between a model with shaders and without, metal and water Cg shaders are applied on an unbuilt project -- the Cloud Forest Biosphere designed by the architect, Geoffrey Bawa. The shaders are applied on the lake, the frames and truss system of the pyramid. Figure 4.28 shows the project before and after applying shaders. The lake looks more realistic and the trusses on the pyramid have more definition and depth. Table 4.8 shows both their properties and it clearly shows that applying Cg shaders will slow down the performance of the real-time simulation.

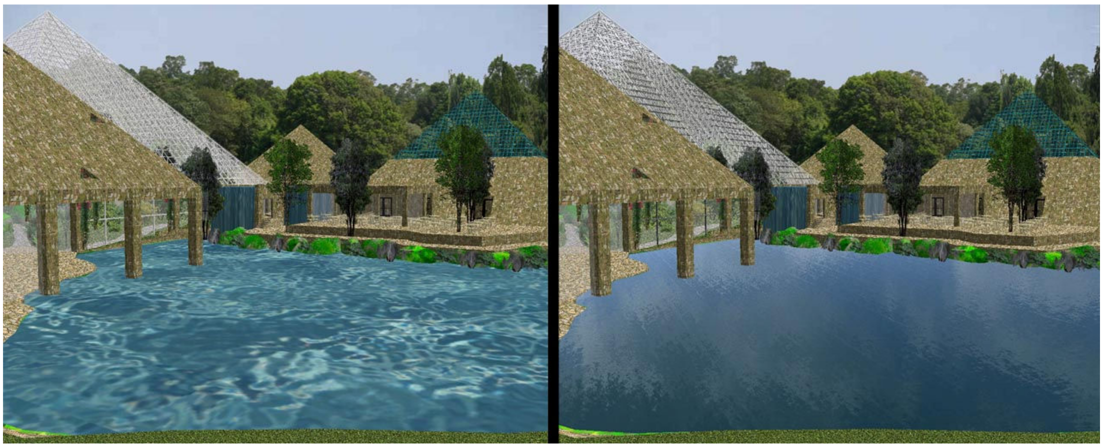


Figure 4.28. Cloud Forest Biosphere without and with Advanced Shaders

TABLE 4.8. Cg Shader Impact on Cloud Forest Biosphere

Model	**CFB without Cg shader**	**CFB with Cg shader**
Frame Rate (fps)	12	7.1
Total Triangles	347,928	347,928
Shaders on Triangles	0	168,118
Amount of Shaders	0	2
Percentage Applied (%)	48.32	48.32
Texture Memory (Kb)	10528	11243
Vertex Memory (Kb)	6338	8794

From there, an experiment was created to find the relationship between advanced shaders and frame rate. The experiments were divided into the usage from one to five shaders per simulation run. For each simulation run, increasing percentage of the triangles in the scene were applied with shaders in increment of 5%, 10%, 20%, 50%, 80% and 100%. The corresponding frame rate, vertex and texture memory used were recorded. For fairness, the same model of Cloud Forest Biosphere was used so that it will be consistent. Model parts were added and subtracted from it depending on the model size range required. The model size is categorized in terms of number of triangles.

The range of model size is 10,000 to 2,000,000 triangles and 6 model sizes were used i.e. Model A (20,014 triangles), Model B (101,832 triangles), Model C (234,978 triangles), Model D (1,009,740 triangles), Model E (1,517,200 triangles) and Model F (2,021,360 triangles). From the collection of 3D models, there is confidence that this range of models represents a good range of architectural projects from a simple building to an urban scale development. The intention is also to consider the range of materials that can possibly be applied with advanced shaders.

The five shaders used in the experiments were:
(i) Chromatic Glass; (ii) Fabric; (iii) Granite; (iv) Wood; (v) Water

The experiment clearly showed that the advanced shaders will slow down the performance of the simulation as expected. If used extensively in huge scenes, they are resource intensive to render, consuming a lot of processing power. However, they can be reasonably used in small to mid range architecture projects. For instance, in the experiment, all models with less than 1,000,000 triangles (for one to five shaders) met a real-time performance of 6 frames per second. They start to impact the frame rate critically when the number of triangles is beyond 1,000,000.

Table 4.9 shows the results from the experiments of the three larger models of Model D (1,009,740 triangles), Model E (1,517,200 triangles) and Model F (2,021,360 triangles).

From the test results, it is clear that frame rate drops significantly at the higher triangle counts as follows:

1. From 50% onwards of Model D applied with shaders (i.e. 504,870 triangles), for one to five shaders
2. From 20% onwards of Model E applied with shaders (i.e. 303,440 triangles), for one to five shaders
3. From 20% onwards of Model F applied with shaders (i.e. 404,272 triangles), for one to five shaders

It is very clear that the frame rate does not change much with the increase in shaders from one to five shaders in the scene. Therefore, it is proven that if there are five shaders and less used in a scene, triangle count rather than shaders affects performance the most.

TABLE 4.9. Simulations with Frame Rate Below 6

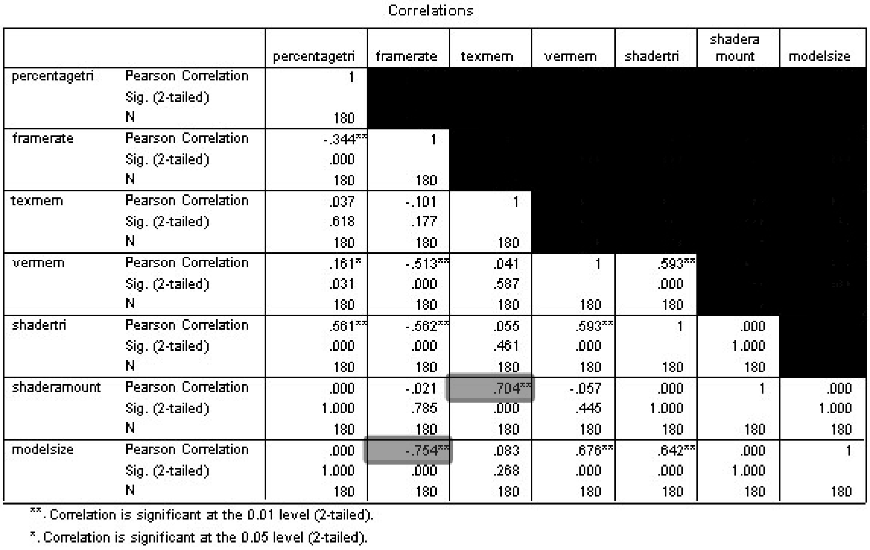
1 Shader					
Model	**Percent**	**Shader**	**Frame Rate**	**Texture Memory**	**Vertex Memory**
D	50%	504870	4.3	1184	9849
D	80%	807792	2.7	1568	13454
D	100%	1009740	2.2	1568	12233
E	20%	303440	4	256	30812
E	50%	758600	2.3	256	40965
E	80%	1213760	1.5	256	48475
E	100%	1517200	1.2	256	55836
F	20%	404272	4	640	25218
F	50%	1010680	1.9	640	30711
F	80%	1617088	1.2	640	45737
F	100%	2021360	0.95	640	53891

2 Shaders					
Model	Percent	Shader	Frame Rate	Texture Memory	Vertex Memory
D	50%	504870	4.4	1312	12901
D	80%	807792	2.6	1152	9349
D	100%	1009740	2.2	1184	15285
E	20%	303440	4.45	2304	29848
E	50%	758600	2.2	2304	40965
E	80%	1213760	1.45	2304	48475
E	100%	1517200	1.2	2304	56090
F	20%	404272	4.05	1152	25218
F	50%	1010680	2	1152	30711
F	80%	1617088	1.3	1152	45737
F	100%	2021360	1	1152	53891

3 Shaders					
Model	Percent	Shader	Frame Rate	Texture Memory	Vertex Memory
D	50%	504870	4.65	2029	13108
D	80%	807792	3.25	2029	15400
D	100%	1009740	2.7	1696	18336
E	20%	303440	4.8	3181	30051
E	50%	758600	2.7	3181	40965
E	80%	1213760	1.7	3181	48475
E	100%	1517200	1.4	3181	56090
F	20%	404272	4.15	2029	31043
F	50%	1010680	2.35	2029	30711
F	80%	1617088	1.6	2029	45737
F	100%	2021360	1.5	2029	53891

Model D (1,009,740 triangles), Model E (1,517,200 triangles), Model F (2,021,360 triangles). The shader column represents the amount of triangles applied with it. Both texture and vertex memory calculated in Kb.

TABLE 4.10. Correlations of All Variables



Correlations

		percentagetri	framerate	texmem	vermem	shadertri	shaderamount	modelsize
percentagetri	Pearson Correlation	1	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]
	Sig. (2-tailed)		[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]
	N	180	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]
framerate	Pearson Correlation	-.344**	1	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]
	Sig. (2-tailed)	.000		[redacted]	[redacted]	[redacted]	[redacted]	[redacted]
	N	180	180	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]
texmem	Pearson Correlation	.037	-.101	1	[redacted]	[redacted]	[redacted]	[redacted]
	Sig. (2-tailed)	.618	.177		[redacted]	[redacted]	[redacted]	[redacted]
	N	180	180	180	[redacted]	[redacted]	[redacted]	[redacted]
vermem	Pearson Correlation	.161*	-.513**	.041	1	.593**	[redacted]	[redacted]
	Sig. (2-tailed)	.031	.000	.587		.000	[redacted]	[redacted]
	N	180	180	180	180	180	[redacted]	[redacted]
shadertri	Pearson Correlation	.561**	-.562**	.055	.593**	1	.000	[redacted]
	Sig. (2-tailed)	.000	.000	.461	.000		1.000	[redacted]
	N	180	180	180	180	180	180	[redacted]
shaderamount	Pearson Correlation	.000	-.021	.704**	-.057	.000	1	.000
	Sig. (2-tailed)	1.000	.785	.000	.445	1.000		1.000
	N	180	180	180	180	180	180	180
modelsize	Pearson Correlation	.000	-.754**	.083	.676**	.642**	.000	1
	Sig. (2-tailed)	1.000	.000	.268	.000	.000	1.000	
	N	180	180	180	180	180	180	180

**. Correlation is significant at the 0.01 level (2-tailed).

*. Correlation is significant at the 0.05 level (2-tailed).

A bivariate correlation is calculated to test the strength of the relationship between two variables without considering the interference from other variables. From Table 4.10, all the seven variables were tested.

The variables are:

1. percentagetri : percentage of triangles from 5-100%
2. framerate : frame rate during simulation
3. texmem : texture memory consumed
4. vermem : vertex memory consumed
5. shadertri : amount of triangles applied with triangles
6. shaderamount : the amount of shaders used from 1-5
7. modelsize : the size of the 6 models from Model A to Model F

From the results, the correlation coefficient between frame rate and model size is seen to be highest at -0.754. The second highest goes to the correlation coefficient between texture memory and shader amount at 0.704. Both are correlated at p-values of less than 0.01 which means they are statistically significant. They have strong negative and positive correlations respectively. This means that the bigger the model size, the lower the frame rate. Another conclusion is that the more shaders there are in the scene, the more texture memory will be consumed.

A graph shown in Figure 4.29 is plotted to represent all five shaders. Six readings (5, 10, 20, 50, 80, 100%) were taken per model for all six models from one to five shaders. This means a total of 180 combinations. This is to get a general fit for all the shaders against the amount of triangles. The formula enables the prediction of frame rate (Y) with the amount of triangles known. The high R Square value of 0.901 in Table 4.11 confirms that the frame rate is highly correlated with the triangles count. On average, real-time navigation can still be maintained at the frame rate limit of 6fps for a model that has 342,195 triangles applied with up to five shaders.

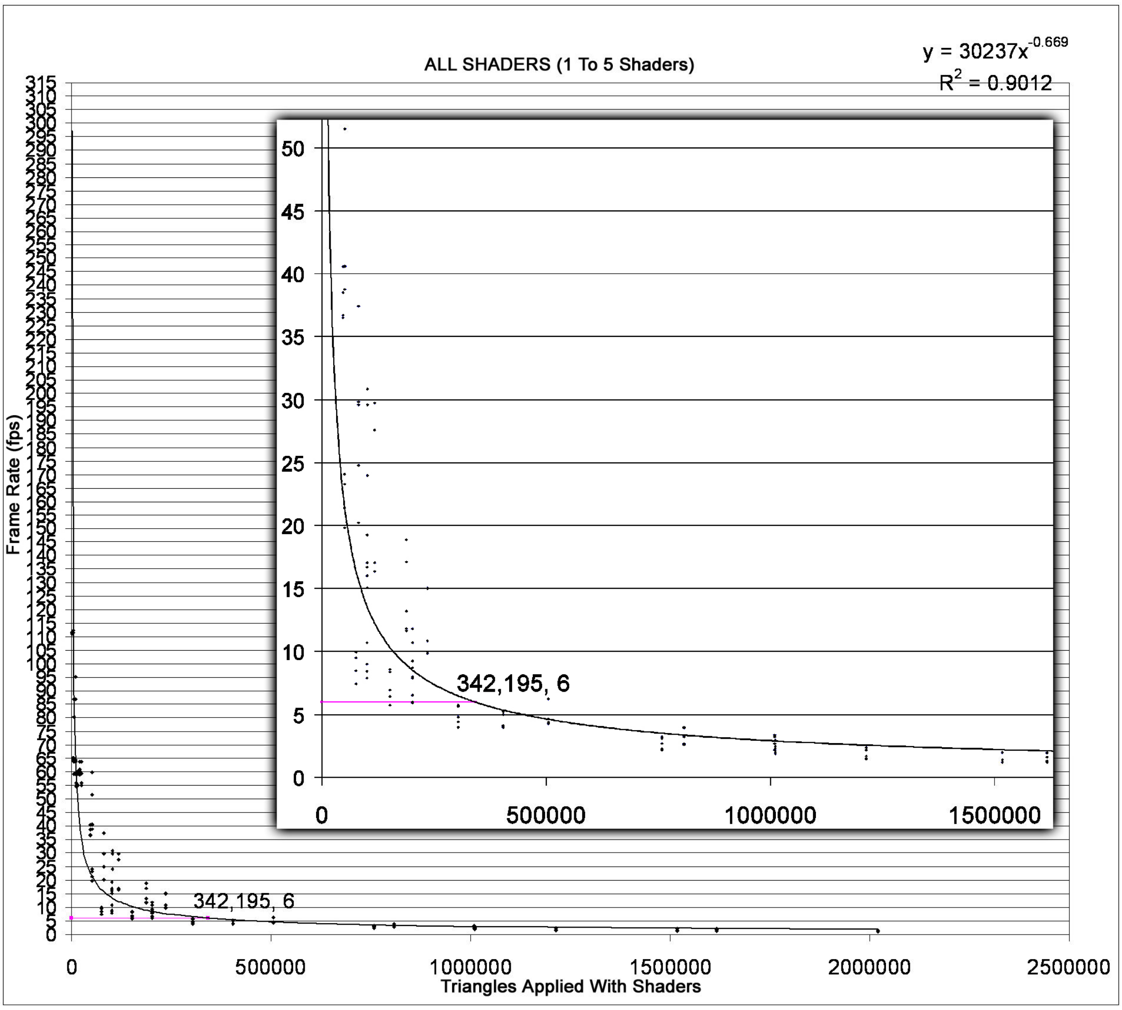


Figure 4.29. Frame Rate against Triangles

TABLE 4.11. All Combinations Graph Fit Result

Model Summary and Parameter Estimates

Dependent Variable: framerate

	Model Summary					Parameter Estimates	
Equation	R Square	F	df1	df2	Sig.	Constant	b1
Power	.901	1624.095	1	178	.000	30237.381	-.669

The independent variable is actualtri.

After that, a graph for each shader amount, as shown in Figure 4.30, is plotted to see the trend for every one of them separately. All the trend types are power graphs with R Square value of 0.8982 and above, which confirm that they fit the formula quite well.

From there, the limit of triangles count to be applied with shaders can be found at frame rate of 6fps for one to five shaders, for instance, a model of 338,350 triangles can have three shaders applied 100% to it and still able to be maintain the threshold frame rate of 6fps in real-time.

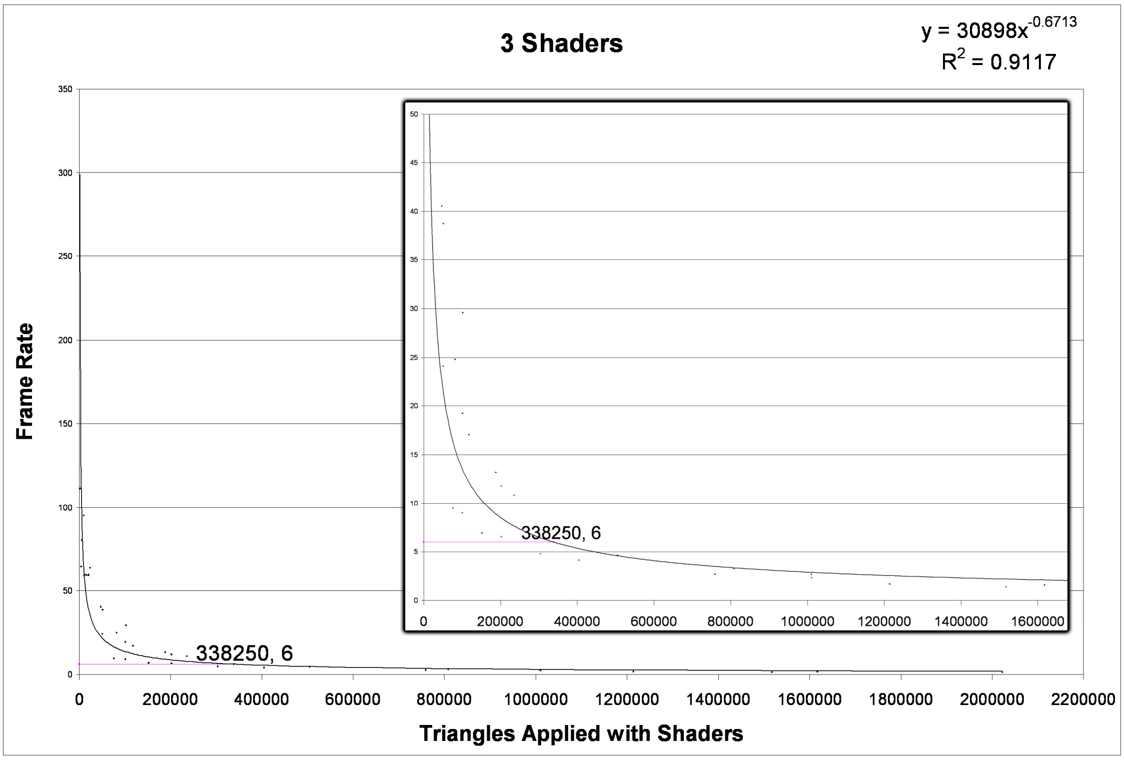


Figure 4.30. Frame Rate against Triangles Count for 3 Shaders

The results clearly show that the amount of triangles is the biggest factor in slowing down the frame rate and not the advanced shaders. This means that advanced shaders are not such a burden to the simulation to the extent that there is a need to be extremely careful in using them.

Three main conclusions are:

1. Shaders can be used extensively (i.e. 100% of the model can be applied with shaders) for a model size range of up to 300,000 to 400,000 triangles before performance suffers (Figure 4.29).
2. The number of shaders (e.g. water, glass, etc) used in the scene will only affect the performance if more than 50% of a model of size up to 1,000,000 triangles is applied with shaders as shown in Table 4.9 above.
3. The number of shaders, up to five types in a scene, will not make much difference to the performance as clearly shown in Table 4.9 above.

The table and graph above inform users of the impact shaders will have on the simulation based on the number of shaders used, the number of triangles (from small to big scale models) and the percentage of the 3D model being applied with shaders. Advanced shaders do affect frame rate but the amount used – up to five as shown in the experiments conducted, show very little impact on frame rate.

Advanced shaders are here to stay and its use will be expected to get more popular in the near future. Their capabilities are expected to improve constantly with the flourishing games market. The expectation is that in the future, there will be more flexibility in integrating and importing/exporting of the shaders between softwares and between different platforms. The architectural industry should take advantage of the realism these advanced shaders can provide to visualise our real-time scenes.

4.2.3 Scripting, Collision Detection, Looped Video and Audio

During the completion of 3D models for VR simulation, a few variables are observed to slow down performance. They include scripting, collision detection, looped video and audio. These variables are used in projects when necessary according to requirements. More tests are needed to fully understand their impacts towards the performance of the VR visualisation.

Scripting in EON^{TM} Studio^{TM} Professional^{TM} is done in Javascript and VisualBasic Script. The scripts are used in the cloud forest biosphere project to detect a position of the camera that activates the fog whenever the user enters the building. It is also used to open and close the doors whenever the person is nearby at a distance. It is also used to activate voice files when approaching the area for some guided tour spoken information.

Collision detection will always slow down the performance by constantly updating when navigation takes place. It includes gravity by always maintaining the viewer at a given height from the ground and from all the surfaces that the viewer contacts in the scene. This is to avoid the user from going through surfaces. In the project, when collision detection is used, it is observed that it does slow down the navigation performance.

In the Cloud Forest Biosphere project, looped video and looped audio are heavily used. The looped video is used to simulate the continuous flow of water from the waterfall. The looped audio is used as 3D sound for insects and animals in the scene. The inclusion of both files in the scene does contribute to performance decline. Compressions to the video and audio quality can be done to make the file size smaller.

4.2.4 Hardware and Software Comparison

It is expected before the experiment that different hardware and software will give different performance results. The hardware specifications are totally different, ranging from different processors to different amount of RAM. For the hardware comparison, four hardware systems were used for the experiment. A part of Chang'An 3D model which is already available in EON^{TM} Studio^{TM}'s EOZ format as shown in Figure 4.31 was used to run on all the systems available.

Figure 4.31. Part of Chang'An Model

It is very clear that the frame rate count is different for all four different hardware specifications as shown Table 4.12. The DELL™ Precision™ 650 was tested in both stereoscopic mode and mono mode. Therefore, this proves that different hardware specifications will create different performance of the same 3D model. Hence, the final experiment will be conducted on all hardware systems to see the overall impact they have on the final results.

TABLE 4.12. Frame Rate Performance of Different Hardware Specifications

Hardware Systems	**Navigation (Hertz)**
DELL™ Precision™ 650 workstation (stereo)	10
DELL™ Precision™ 650 workstation	5
DELL™ Precision™ 670 workstation	45
DELL™ Precision™ 380 desktop	15
DELL™ Inspiron™ 9300 laptop	21.3

For software comparison, EON™ Studio™ Professional™ was compared with Quest3D®. The same 3D model was constructed in 3ds Max® 8 to be exported to both EON™ Studio™ Professional™ as an EOZ format and Quest3D® as a CGR format. The experiment was countered checked with an external software called Beepa FRAPS Version 2.8.2 Build 6488. It will run behind both software when the simulation starts to give it an external calculation of frame rate as shown in Figure 4.32 and 4.33. In the simulation, a yellow figure will appear at the bottom left corner of the screen. The self calculation by the software is located at the top left corner of the screen.

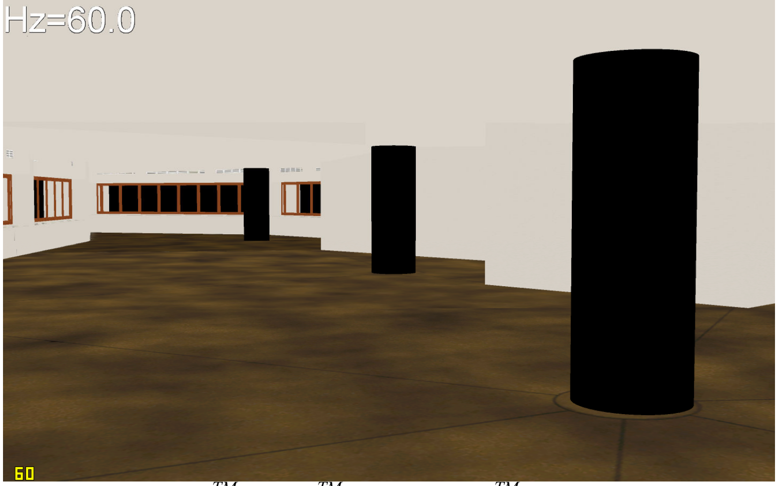


Figure 4.32. EON^{TM} Studio^{TM} Professional^{TM} running at 59 - 60 Hz

```
Framerate: 60
Viewport: 1024 x 768
Bump map: Yes
Pixel shading: Yes
```

Figure 4.33. Quest3D® running at 59 - 60Hz

It is proven therefore that the same 3D model produces the same frame rate regardless of the different software used (either EON^{TM} Studio^{TM} Professional^{TM} or Quest3D®). The same trend is observed even using different hardware configurations. It is assumed that no matter which software is chosen, at least with the basic four variables used, they will generate the same results. The same cannot be said if more variables were included, especially the secondary ones. To further support the case above, an identical experiment was carried out with the 3d models used for Chapter 4.1.5 (Texture Count). The frame rate count is quite consistent for all the eleven 3d models used. The largest model with the most textures (Tri 2000K with 100 textures) was not used because it crashed for the test in EON™ Studio™ Professional™. The difference in frame rate is at most 0.2fps difference as shown in Table 4.13.

TABLE 4.13. Frame Rate Performance of Different Software

Models \ Details	Frame Rate (EON™ Studio™ Professional™)	Frame Rate (Quest3D®)
Tri 100k with 10 textures	21.2	21.2
Tri 100k with 50 textures	21.4	21.4
Tri 100k with 100 textures	20.3	20.1
Tri 500k with 10 textures	2.5	2.6
Tri 500k with 50 textures	3.5	3.5
Tri 500k with 100 textures	4.4	4.6
Tri 1000k with 10 textures	3.5	3.5
Tri 1000k with 50 textures	2.4	2.3
Tri1000k with 100 textures	0.7	0.7
Tri 2000k with 10 textures	1.0	1.1
Tri 2000k with 50 textures	0.9	1.0

Therefore, from this point onwards, the experiments were done fully in EON™ Studio™ Professional™ but with different hardware settings to understand the difference. EON™ Studio™ Professional™ was used because the research started with this software and a lot of projects have already been created in it. Thus, the building of the samples will be a lot easier and faster while there is totally none available yet from Quest3D®.

In conclusion, the independent variable tests have shown how each of the variables influences performance in the VR architectural visualisation. Along the way, time taken, frame rate, texture memory and vertex memory were recorded and analysed. It is clear that triangle size and software do not affect performance at all. Time taken is the same regardless of how complicated the 3D scene gets. Vertex memory and texture memory do increase when simulations get complicated with more complex 3D models but the amount of memory available today in the graphic cards (up to 1.5GB) and systems (2GB to 8GB RAM) are sufficient to sustain them. Table 4.13 gives a summary of the impact of all these variables against frame rate.

The secondary variables are not explored further from this point because of two reasons. There is no reason to optimise them first since their use in projects is optional, depending on project features, requirements and demands. Apart from that, they can only exist if the primary variables are in the models. The goal for this research is to optimise the foundation first, which are the primary variables or the first generation variables. If the first generation variables are not efficient, there is no point in making the second generation variables efficient since they only exist on top of the first generation variables. Findings from this chapter will serve as good preparation for the final part of the research, which is to look into the quantitative and qualitative aspects of the visualisation by conducting an experiment and a survey.

TABLE 4.13. Frame Rate Impact Against Variables

Primary Variable	Time Taken
Triangle Count	x

-√ = mild impact,√ = impact, X = no impact

Primary Variable	Frame Rate
Triangle Count	√
Triangle Size	x
Geometry Count	√
Texture Count	√
Texture Resolution	-√
Vertex Count	√

-√ = mild impact,√ = impact, X = no impact

Secondary Variable	Frame Rate
Light Count	-√
Particle Systems	√
Programmable / Advanced Shaders Count	√
Scripting	√
Collision Detection	√
Looped Video / Audio	√
Hardware	√
Software	x

-√ = mild impact,√ = impact, X = no impact

CHAPTER 5.0: EXPERIMENT AND SURVEY

In the previous chapter, the relationships between variables against frame rate, time taken, texture and vertex memory were recorded and analysed. The tests have clarified how these variables behave in VR architectural visualisation. The experience and knowledge gained will be crucial to the experiments described in this chapter, which deals with the quantitative and qualitative aspects of VR visualisation.

5.1 Experiment

The final experiment here is the first part of the research methodology which looks at the quantitative aspects of optimisation. It takes into account the four core variables in constructing 3D models as well as showing its surface material VR simulation. Unlike in the previous chapter, this chapter will see the testing of all four variables together against frame rate instead of individually.

The intention of conducting this experiment is to come up with a formula to predict the frame rate of a 3D model simulation that is intended to be visualised in the VR presentation. This will help eliminate the process of trial and error at the early stage of design and facilitate planning workflow to save time and effort. It will also give the designers some options to strategize the 3D model design requirements and demands depending on the hardware system they are using to simulate the presentation. They can then wisely choose ways to show their projects in circumstances that is still possible.

5.1.1 The Experiment Variables

The dependent variable is **frame rate**. It is defined as the number of frames per second (fps) and is measured in Hertz. Vertex memory and texture memory were excluded from this experiment because they are not the main contributors to performance decline as identified in the independent variable tests in the previous chapter.

The four independent variables to be measured are vertex count, triangle count, geometry count and texture count. The following diagrams in Figure 5.1, 5.2, 5.3 and 5.4 illustrate the difference of each variable, with vertex to be the corners of triangles, the triangles to represent the surfaces, the two geometries to represent two objects and two textures to map two objects.

(i) Vertex Count

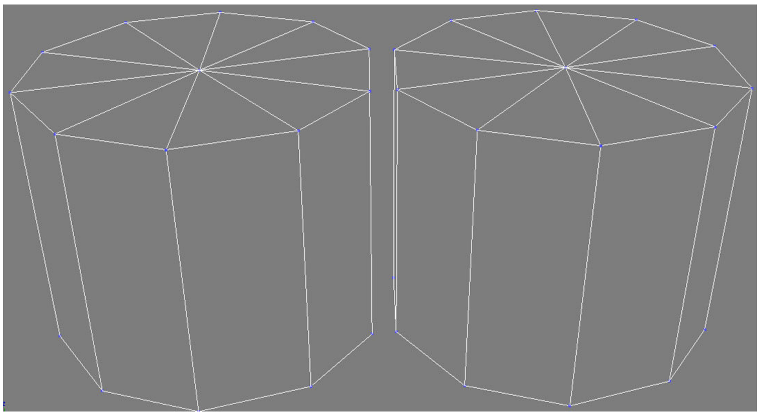


Figure 5.1. Vertex

(ii) Triangle Count

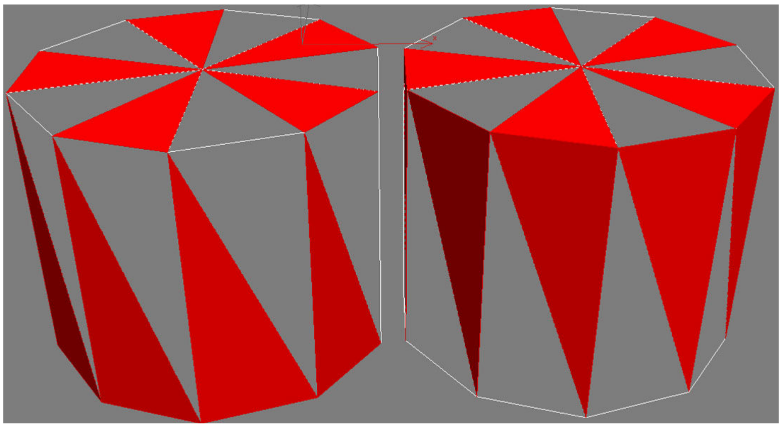


Figure 5.2. Triangle

(iii) Geometry Count

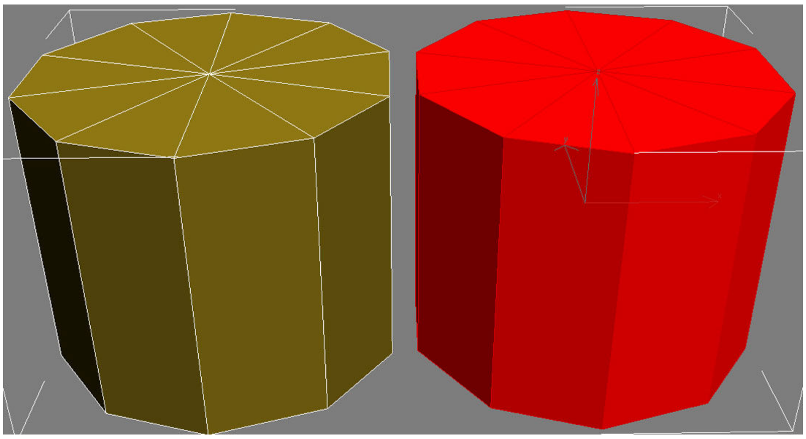


Figure 5.3. Geometry

(iv) Texture Count

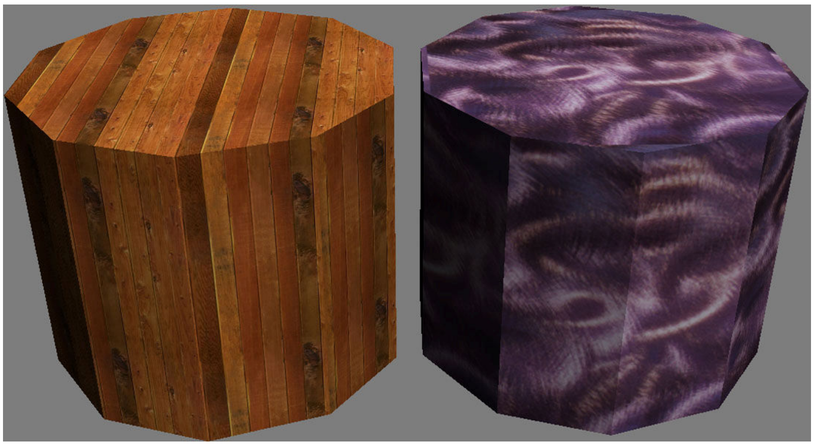


Figure 5.4. Texture

These variables are the most fundamental elements that make up the whole 3D model. They come in various different combinations depending on each student project to be used in the experiment.

5.1.2 The Experiment Samples

The experiment was carried out with samples from Architecture Year 2 to 5 student design projects, independent studies Year 4 student design projects as well research project works. The department collects student design works in a digital archive and this gives the opportunity to utilize those 3D models for the experiment. All these files were sent into 3ds Max® 8 for the final export into EON™ Studio™ Professional™. From all the samples available, the different software used by the students in their design works were:

1. Autodesk® AutoCAD® 2007

2. Google™ Sketchup® 6
3. auto.des.sys, Inc. FormZ® 5
4. Autodesk® Maya® 7
5. Graphisoft® ArchiCAD® 10
6. Bentley Systems, Inc. MicroStation® V8 2004 Edition
7. Autodesk® Revit® Building 9
8. Autodesk® 3ds Max® 8

The opportunity to cover so many types of software used by students of the Department of Architecture, School Design and Environment, National University of Singapore was advantageous. It will give a good coverage of all the common software used in the world today. The textures in all the projects were changed from traditional textures like JPEG, PNG, GIF, TARGA and BMP to DDS (DirectDraw Surface) format in 72dpi with mipmap. All the files were merged (if they are MAX file format from 3ds Max®) and exported into 3ds Max® 8 as DWG2004 and 3DS file formats. From there, they were exported out as EON™ Studio™ Professional™'s EOZ file format to be visualised in the software.

In the final samples collection, there were 105 projects altogether as shown in Table 5.1. There are on average 100 students per year and this means that there is a sample of 105 out of 400 projects (4 years X 100). This means 1 out of 3.81 projects is sampled from the whole Department of Architecture, School of Design and Environment, National University of Singapore. This provided enough sampling to cover the different types, scales and complexities of architectural projects and it is hope that it will give a fair outcome to the results later.

TABLE 5.1. Number of Architecture Student Design Projects Amount

Architecture Student Design Projects	Quantity
Unknown	2
Year 2	40
Year 3	18
Year 4	34
Year 5	11
TOTAL:	**105**

Following the rule of thumb, one independent variable for every 10 observations would be good enough (Elliott and Woodward 2007). Therefore, in this case, 40 observations would be sufficient. However, more samples were taken because of the variety of software used by the students. This is in the hope that wider coverage would be achieved. Since the relationships between frame rate and all the independent variables are not linear, the logarithm of them all was taken for the final analysis. This is to make the relationships linear so that the Simple Linear Multi Regression can be performed.

5.1.3 The Experiment Settings

The experiments were done using five different hardware systems. They were from different categories such as workstations, desktops and laptops. This is in the hope that most hardware specifications available in the world today could be covered. It will definitely help those who use the same hardware systems to prepare and present VR presentations and those who use different hardware systems.

(i) DELL™ Precision™ 650 Workstation (Dual Processor Workstation)

Intel® Xeon® Dual 3.20 GHz (32-bit), 2GB RAM

256MB AGPX8 Leadtek nVidia® Quadro FX3000G (Stereoscopic Mode Ready)

19” LCD Display (Resolution: 1280 X 1024 X 120Hz)

Microsoft® Windows® XP Professional Version 2002 Service Pack 2

EON™ Studio™ Professional™ Version 5.5.0 (build 1241)

(ii) DELL™ Precision™ 670 Workstation (Single Processor Workstation)

Intel® Xeon® 3.2 GHz (32-bit), 1GB RAM

640MB AGPX8 3DLabs Wildcat Realizm 800 (400MHz Integrated DAC)

19” LCD Display (Resolution: 1280 X 1024 X 120Hz)

Microsoft® Windows® XP Professional Version 2002 Service Pack 2

EON™ Studio™ Version 5.5.0 (build 1241)

(iii) DELL™ Precision™ 380 Desktop (Desktop)

Intel® Pentium® 4 HyperThreading 3.40 GHz (32-bit), 1GB RAM

128MB PCI Express X16 ATI™ FireGL™ V3100 (400MHz Internal DAC)

17” LCD Display (Resolution: 1280 X 1024 X 75Hz)

Microsoft® Windows® XP Professional Version 2002 Service Pack 2

EON™ Studio™ Professional™ Version 5.5.0 (build 1241)

(iv) DELL™ Inspiron™ 9300 Laptop (Laptop)

Intel® Pentium® M 750 Centrino 1.86Ghz (32-bit), 2GB RAM

256MB PCI Express x16 nVidia® GeForce^{TM} Go 6800

17" UltraSharp Wide Screen XGA+ TFT Display (Resolution: 1440 X 900 X 60 Hz)

Microsoft® Windows® XP Professional Version 2002 Service Pack 2

EON^{TM} Studio^{TM} Professional^{TM} Version 5.5.0 (build 1241)

(v) DELL^{TM} Inspiron^{TM} 1520 Laptop (Duo Core Laptop)

Intel® Core^{TM} 2 Duo T7300 2GHz (64-bit), 2GB RAM

256MB GDDR2 PCI Express x16 nVidia® GeForce^{TM} 8600M GT

15.4' Widescreen WXGA (Resolution: 1280 X 800 X Hz) with TrueLife

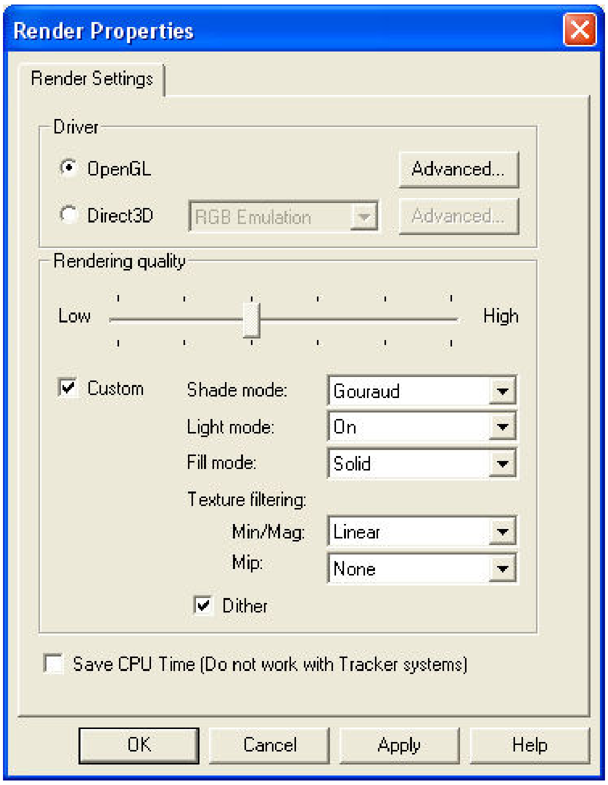
Microsoft® Windows® Vista^{TM} Home Premium

EON^{TM} Studio^{TM} Professional^{TM} Version 5.5.0 (build 1241)

The EON^{TM} Studio^{TM} Professional^{TM} setup for VR simulation was run on OpenGL^{TM} driver with a rendering quality of 3 out of 6. This setting is the minimum quality set with 3D model surfaces still looking smooth and not triangulated. Therefore, this setting was selected. All 3D models were applied with ambient light so that the whole scene would be visible. A walk node was applied on the Camera view to allow the user to navigate freely in any direction possible. Finally, the simulation was run and the frame rate recorded in Hertz unit or frames per second.

Of all the systems above, only system 1 (DELL^{TM} Precision^{TM} 650) was run in stereoscopic mode because the visualisation lab had the facilities to project stereoscopic presentation. Therefore, the triangle count will be doubled and hence, the frame rate will drop by half its amount on average. Only system 2 (DELL^{TM} Precision^{TM} 670) is with the basic EON^{TM} Studio^{TM} which is without the Professional^{TM} version. The basic EON^{TM} Studio^{TM} version has fewer tools for VR presentation, but is sufficient enough for the experiment to be carried out.

Before the simulation was started, the rendering properties of EON^{TM} Studio^{TM} Professional^{TM} were set to run in OpenGL^{TM} driver mode with a rendering quality of 3 as shown in Figure 5.5. The reason for choosing 3 is because it is the minimum level where models do not appear triangulated, as shown in Figure 5.6, for lower and above three differences.



Driver

● OpenGL

○ Direct3D RGB Emulation

Rendering quality

☑ Custom

Shade mode: Gouraud

Light mode: On

Fill mode: Solid

Texture filtering:

Min/Mag: Linear

Mip: None

☑ Dither

☐ Save CPU Time (Do not work with Tracker systems)

Figure 5.5. Rendering Properties

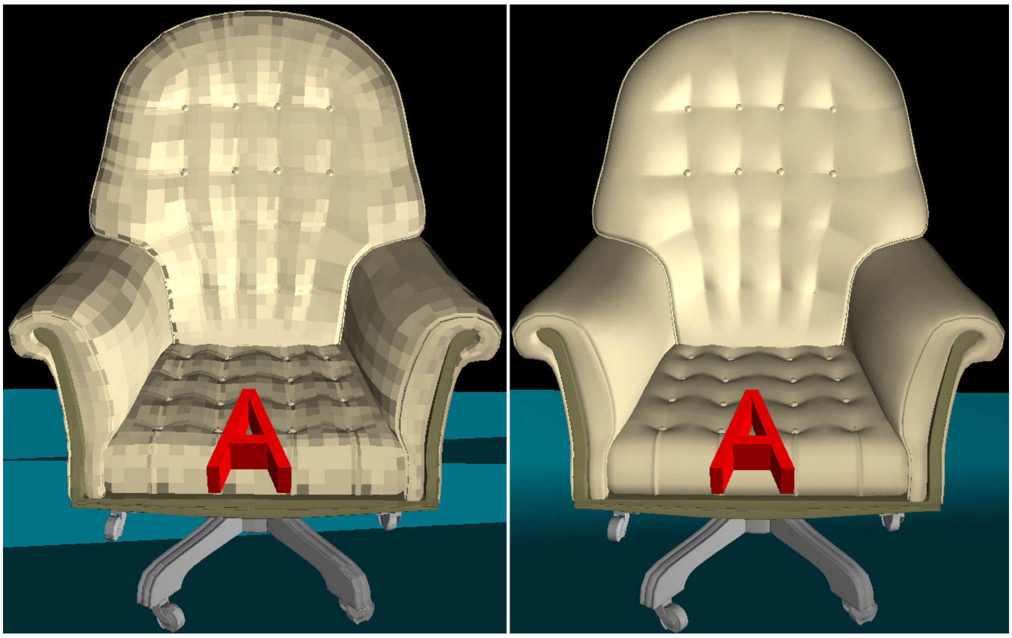


Figure 5.6. Lower and Higher Rendering Quality Chair

In the only hardware system running on Microsoft® Windows® Vista^{TM}, d3drm.dll needed to be installed separately before running any simulation. It has been taken out from the current Windows® Vista^{TM} versions and is available in the older Windows® XP versions.

In conclusion, this experiment looks at the quantitative aspects of optimisation. The research design used was experiments of the 3D models running in the computer. The figures shown on the computer display during running of the simulation were collected as data.

5.2 Survey

This survey was done because of the importance to understand human requirements and needs in the field of architecture. At the end of the day, VR simulations are presented for humans to visualise and understand design of spaces. Therefore, their feedback and opinions are required to prepare the best possible simulations suited for them. It will also help the designers who prepare them to know what is required and demanded by their potential audience. In addition, information about subjects' profiles like age, gender, occupation, faculty/department and 3D games (first person POV (Point-of-View)) (Novak 2005) experience were recorded as a secondary interest to see whether there is / are any trend / trends available. Human opinions and complaints were also recorded during the test to understand their feelings throughout all six simulations.

Therefore, this part of the research looks into the qualitative aspects of optimisation. The research design used was a survey. The survey came in the form of multiple choice questionnaires.

5.2.1 The Survey Questions

The survey was done in a lab with large stereoscopic screens and consisted of six questions related to triangles count, texture resolution and frame rate count. It aims to understand human perception and tolerance of optimisations done on triangle counts, texture resolution and frame rate. This is to give designers of VR simulation feedback and guidelines of what the acceptable ranges are. This is useful for preparing simulations efficiently.

For example, if a texture resolution of 1024 X 1024 is enough for curved surfaces, it would not be necessary for a texture resolution of 4096 X 4096 to be used, as this will waste resources and slow down performance. This is because if 1024 X 1024 resolutions are good enough, there are no added benefits by applying textures of higher resolution. If there are hundreds of higher resolution surfaces in the scene to be simulated, performance will be slowed down. The same case can be applied to a chair of 50,000 triangles. If an optimised amount of 50% is acceptable, 25,000 triangles can be saved for a chair. If the simulation has 100 chairs in a restaurant interior, 25, 000,000 triangles can be saved, which is a large amount. The frame rate acceptance level will enable the prediction calculation of the final VR project. This predicted amount will be acceptable by almost everyone. From this, as many features as possible can be added to reach that limit.

The survey helps in the understanding of human acceptable ranges for all three categories and this will build on top of the experiment results as another factor to consider for VR simulation preparation. The participants were requested to select the 3D model / models which quality/ qualities were not acceptable to them.

The questions are shown in Figure 5.7.

Triangles Complexity Test

From which object onwards the quality is not acceptable to you? Please circle your answer.

1. Chair A. B. C. D. E. F. G.

2. Sink A. B. C. D. E. F. G.

Texture Resolution Test

From which object onwards the quality is not acceptable to you? Please circle your answer.

1. Table A. B. C. D. E. F. G.

2. Curve Wall A. B. C. D. E. F. G.

3. Stone A. B. C. D. E. F. G.

Navigation Test

Which quality/qualities is/are not acceptable to you (too slow)? Please circle your answer.

A. B. C. D. E. F. G.

Figure 5.7. The Survey Questions

5.2.2 Triangles Complexity Test

There were two questions in this category. The two questions involved one object each – a chair and a sink. They were optimised using 3ds Max® 8's MultiRes modifier. It was chosen out of the three optimisation methods available because it can provide the most pleasing results with the lowest triangle count possible. In addition, it does the job fast and easy enough. It was chosen during our optimisation process of the Cloud Forest Biosphere project. The other methods tested were the "Optimise" modifier in 3ds Max® 8 and a 3ds Max® 8 plug-in called Polygon Cruncher 7.22. The choices from A to G were a result of optimising the chair from 100% to 5%. The choices were 100% (A), 55% (B), 45% (C), 35% (D), 25% (E), 15% (F) and 5% (G).

The original chair had 69,854 triangles (100%). It was optimised to 38,179 triangles (55%), 31,218 triangles (45%), 24,214 triangles (35%), 17,237 triangles (25%), 10,263 triangles (15%) and 3,304 triangles (5%) as shown in Figure 5.8. The sink originally consisted of 69,482 triangles (100%). It was optimised to 38,056 triangles (55%), 31,021 triangles (45%), 23,978 triangles (35%), 16,970 triangles (25%), 10,060 triangles (15%) and 3,163 triangles (5%) as shown in Figure 5.9.

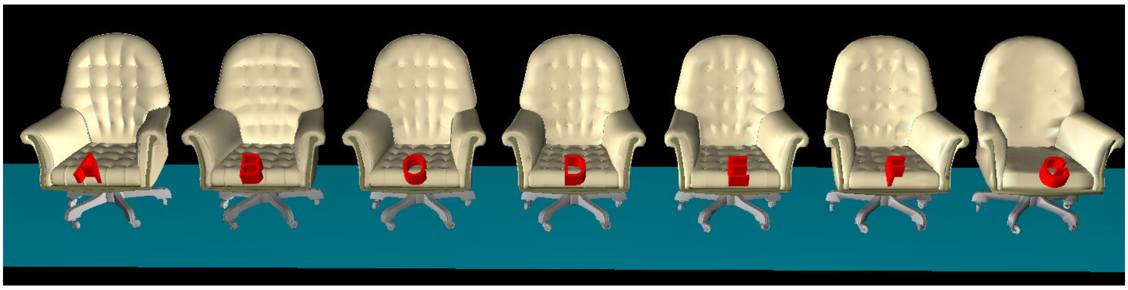


Figure 5.8. Chairs Triangles Complexity Test Decreasing Quality from Left to Right

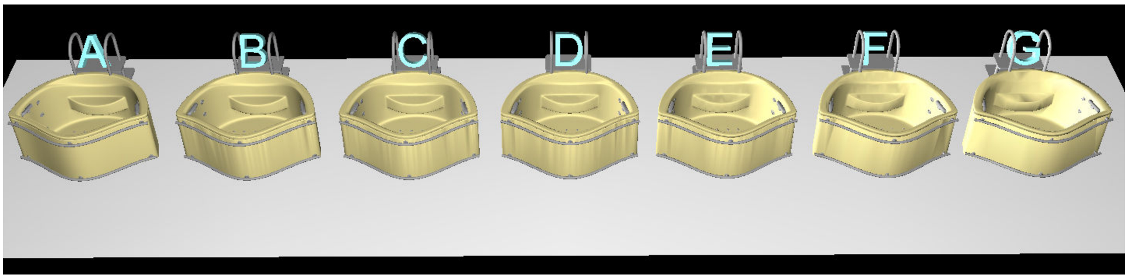


Figure 5.9. Sinks Triangles Complexity Test Decreasing Quality from Left to Right

5.2.3 Texture Resolution Test

The texture resolution test was done with the objective of understanding human acceptance of the limits of texture resolution quality. The texture used were all in DDS DirectDraw Surface) format in 72dpi with mipmap. The seven different level of qualities chosen were 4096 X 4096, 2048 X 2048, 1024 X 1024, 512 X 512, 256 X 256, 128 X 128 and 64 X 64. The highest resolution of 4096 X 4096 is the current highest supported resolution for all 3D graphic cards in 3D applications. Therefore, it it was logically used as the highest resolution in the test. In addition, it is considered the highest texture resolution used in the world today in texture mapping for 3D models.

Different texture resolutions were applied to three kinds of surfaces common in architecture. One was on the flat surface of a table as shown in Figure 5.10, the other was on curved walls as shown in Figure 5.11 and the last was on faceted small surfaces of a rock as shown in Figure 5.12. The level of human acceptance of texture resolution quality applied on different type of surfaces was tested to understand each of their limits.

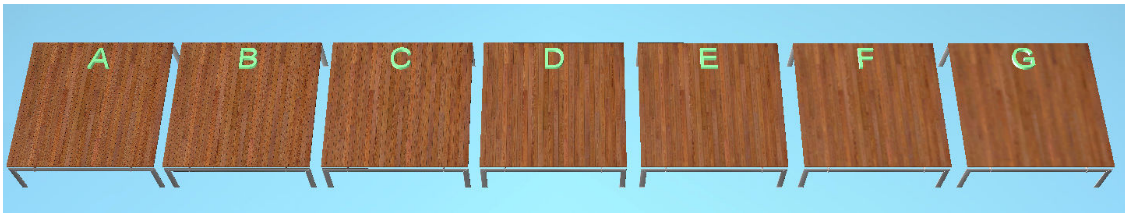


Figure 5.10. Tables Texture Resolution Test Decreasing Quality from Left to Right

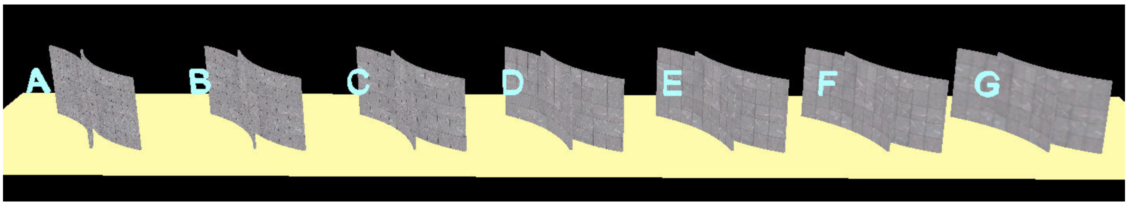


Figure 5.11. Curved Walls Texture Resolution Test Decreasing Quality from Left to Right

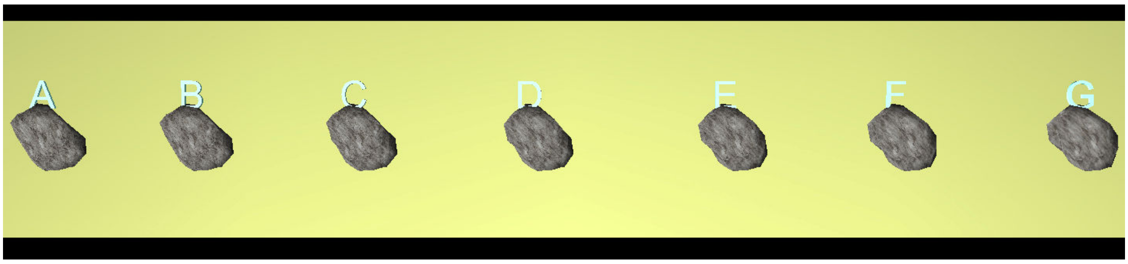


Figure 5.12. Stones Texture Resolution Test Decreasing Quality from Left to Right

5.2.4 Navigation Test

The navigation test was done by getting the participants to view seven repeated pre-recorded animation paths of a scene where they walk towards a few houses which are all fitted in the full screen as shown in Figure 5.13. The animation path ends with the view of the nearest house to the user at full scale fitting the full screen. After a navigation path has finished playing, the animation is played again for the next subsequent choice until the animation ends and the process is repeated from A to G. The participants are given a choice to repeat the playing of any path / paths for them to make their decisions. In the survey, almost all participants asked for a repeat before making their final choices.

Figure 5.13. Navigation Test

5.2.5 The Survey Settings

All 3D models used in the experiment were projected at least to full size on a large screen of 4.5m X 2.5m when displayed to subjects of the survey. This is to ensure that it will be as close as possible to real world size and will be realistic for evaluation. The participants were requested to sit close to the screen, at a distance of approximately 2.5 metres. The simulations were all in stereoscopic mode and hence the participants viewed them with stereoscopic glasses.

The choices were displayed one at a time from the best quality (A) to the worst quality (G). The person who displayed the 3D models announced each alphabet from A to G after verification from the participants that they have had sufficient time to view the 3D model. The average time spent looking at the 3D models was about 8 seconds. After displaying all seven choices, the participants were asked whether they would like to view any 3D models once again. Most users requested for a repeat viewing at the ranges where they will select those 3D models they do not like.

5.2.6 The Survey Subjects

Suitable subjects for the survey are people from all kinds of background and discipline. The argument here is that the potential audience who will view VR visualisation projects could be developers, quantity surveyors, engineers, scientists, accountants, businessmen and people from many other professions other than just architects. Normally, people who are architectural clients are not architects. Therefore, the survey's target participants were from the university community. As a result, 59 subjects from different faculties in the university were recruited to participate in the survey.

Those who took part in the survey included year 1 to year 4 undergraduate students, postgraduate students and a small number of alumni who were already working professionals outside the campus. Subjects' ages ranged from 20 to 31 and consisted of 42 males and 17 females. Out of the 59 subjects, 19 has no 3D games (first person) experience, 18 considered themselves novice players, 12 considered themselves intermediate players and 10 considered themselves advanced players. They came from various departments including architecture, industrial design, real estate, building, electrical engineering, chemical engineering, civil engineering, sociology, general arts and social sciences, mathematics, computing and business. The active participation of people working in diversified fields was encouraging because they most closely resemble the actual audience who would be viewing VR presentations. The views of the actual audience could then be better represented. Virtual reality presentations have been used in so many different industries and it will be crucial to understand the requirements and needs of people from all those different industries.

In conclusion, the final experiment and survey conducted in this chapter utilized and applied the knowledge, experience and findings from the previous three chapters. The final stage of the research, that is to find ways of achieving real-time VR architectural visualisation by exploring both the quantitative and qualitative aspects of the visualisation, has been reached.

CHAPTER 6.0: RESULTS

In this chapter, all the results of the experiments and survey will be discussed.

6.1 Experiment Equations

The variables from Chapter 5 are regressed using a statistical method called Simple Multiple Regression for the purpose of getting an equation to predict frame rate count as well as understanding the strength of each variable in affecting frame rate. It is done with the software SPSS 15.0. Users can use any other statistical software available in the market like Minitab. This method can be used as long as enough samples are collected in the simulation run to get the data. For future upgrades of hardware, software and operating systems, users will need to collect the data from their own 3D models as samples and use Simple Multiple Regression to get the regression equation.

The dependent variable (Y) is the Frame Rate while the independent variables (X) are Triangle Count, Vertex Count, Geometry Count and Texture Count. From the independent variable tests, it is found that the relationships between frame rate against triangle count, vertex count, geometry count and texture count are all not linear. Therefore, the base 10 logarithms of all the variables are taken to achieve linearity before they are regressed using Multiple Linear Regression. The regression will produce five tables, which are descriptive statistics, correlations, model summary, ANOVA and coefficients.

The Multiple Linear Regression procedure performs linear regression on the selected dataset or samples and fits a linear model to an equation:

$$Y = b_0 + b_1 X_1 + b_2 X_2 + \dots + b_k X_k + e$$

where Y is the dependent variable (response) and $X_1, X_2, \dots, X_k$ are the independent variables (predictors) and e is random error. $b_0, b_1, b_2, \dots b_k$ are known as the regression coefficients, which have to be estimated from the data. Linear regression is performed either to predict the response variable based on the predictor variables, or to study the relationship between the response variable and predictor variables.

All CAD and 3D visualisation software should have the functions to count the amount of vertices, triangles, geometries and textures in the 3D model constructed by the user. The user can record the details of all the variables except frame rate, which can be recorded when the model is run in the VR software. All the 3D models are run on different hardware platforms and the figures are calculated.

Table 6.1 shows a summary of the average frame rate for all the samples taken on the six different hardware platforms used. The single processor workstation has the highest average frame rate among all the platforms because of its extremely high-end graphic card, which has more than double the amount of memory than the closest competitor. It also proves that the VR visualisation software, EON^{TM} Studio^{TM} Professional^{TM} does not make full use of the dual processor available in the workstation of the other setup as verified by EONReality^{TM}'s software engineer.

TABLE 6.1. Average Frame Rate for All Samples Taken

Hardware Systems					
Dual Processor Workstation (Stereo)	Dual Processor Workstation (Mono)	Single Processor Workstation	Desktop	Laptop	Core 2 Duo Laptop
42.78 Hz	**62.24 Hz**	**65.82 Hz**	**43.53 Hz**	**30.79 Hz**	**46.99 Hz**

All the samples are regressed to obtain the simple linear multiple regression equation for each hardware configuration.

(i) Dual Processor Workstation (Stereo)

From Table 6.2, it is shown that there are 105 samples in the experiment for all variables. The mean logarithm of frame rate is 1.3186 with a standard deviation of 0.477.

TABLE 6.2. Dual Processor Workstation (Stereo) Descriptive Statistics

	Mean	Std. Deviation	N
logFRs	1.4317	.44377	105
logTRI	4.9994	.67427	105
logVER	4.8190	.68064	105
logGEO	1.7677	.78982	105
logTEX	.9745	.17860	105

From Table 6.3, it is clear that frame rate is highly correlated to triangle count, vertex count and geometry count because the significance level or p-value is less than 0.05. All the variables are negatively correlated to frame rate as indicated by the Pearson Correlation coefficient value.

TABLE 6.3. Dual Processor Workstation (Stereo) Correlations

		logFRs	logTRI	logVER	logGEO	logTEX
Pearson Correlation	logFRs	1.000	-.865	-.857	-.323	-.038
	logTRI	-.865	1.000	.982	.107	.043
	logVER	-.857	.982	1.000	.122	.046
	logGEO	-.323	.107	.122	1.000	.429
	logTEX	-.038	.043	.046	.429	1.000
Sig. (1-tailed)	logFRs	.	.000	.000	.000	.350
	logTRI	**.000**	.	.000	.139	.332
	logVER	**.000**	.000	.	.108	.320
	logGEO	**.000**	.139	.108	.	.000
	logTEX	.350	.332	.320	.000	.
N	logFRs	105	105	105	105	105
	logTRI	105	105	105	105	105
	logVER	105	105	105	105	105
	logGEO	105	105	105	105	105
	logTEX	105	105	105	105	105

A summary of the results is shown in Table 6.4. The values of R range from 0 to 1. Larger values of R indicate stronger relationships. R squared is the proportion of variation in the dependent variable explained by the regression model. The values of R squared range from 0 to 1 as well. Small values indicate that the model does not fit the data well. The sample R squared tends to optimistically estimate how well the models fit the population. The R squared value which is 0.814 means it is closer to 1 and is considered a good fit.

TABLE 6.4. Dual Processor Workstation (Stereo) Model Summary

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	.902(a)	.814	.806	.19536

a Predictors: (Constant), logTEX, logTRI, logGEO, logVER

Table 6.5 shows the analysis of variance. It has a high regression sum of squares in comparison to the residual sum of squares. Therefore, it means that the model accounts for most of the variation in the dependent variable. The significance value of the F statistic is small (smaller than 0.05) which means the independent variables do a good job of explaining the variation in the dependent variable.

TABLE 6.5. Dual Processor Workstation (Stereo)ANOVA(b)

Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	16.664	4	4.166	109.154	**.000(a)**
	Residual	3.817	100	.038		
	Total	20.481	104			

a Predictors: (Constant), logTEX, logTRI, logGEO, logVER
b Dependent Variable: logFRs

Table 6.6 is crucial in constructing an equation for this hardware system. It is from there that the whole formula is developed for prediction of performance. The significant variables are triangle count and geometry count.

TABLE 6.6. Dual Processor Workstation (Stereo) Coefficients(a)

Mode l		Unstandardized Coefficients		Standardized Coefficients	t	Sig.
		B	Std. Error	Beta	B	Std. Error
1	(Constant)	4.176	.176		23.730	.000
	logTRI	-.485	.152	-.737	-3.196	**.002**
	logVER	-.068	.151	-.104	-.452	.652
	logGEO	-.159	.027	-.283	-5.872	**.000**
	logTEX	.298	.119	.120	2.507	.014

a Dependent Variable: logFRs

The regression equation derived from the analysis is:

(7.1) Log (Stereo Frame Rate) =4.176 – 0.485 Log (Triangle Count) – 0.068 Log (Vertex Count) – 0.159 Log (Geometry Count) + 0.298 Log (Texture Count)

(ii) Dual Processor Workstation (Mono)

From Table 6.7, it is shown that there are 105 samples in the experiment for all variables. The mean logarithm of frame rate is 1.6556 with a standard deviation of 0.37881.

TABLE 6.7. Dual Processor Workstation (Mono) Descriptive Statistics

	Mean	Std. Deviation	N
logFRm	1.6556	.37881	105
logTRI	4.9994	.67427	105
logVER	4.8190	.68064	105
logGEO	1.7677	.78982	105
logTEX	.9745	.17860	105

From Table 6.8, it is clear that frame rate is highly correlated to triangle count, vertex count and geometry count because the significance level or p-value is less than 0.05. All the variables are negatively correlated to frame rate as indicated by the Pearson Correlation coefficient value.

TABLE 6.8. Dual Processor Workstation (Mono) Correlations

		logFRm	logTRI	logVER	logGEO	logTEX
Pearson Correlation	logFRm	1.000	-.829	-.824	-.345	-.030
	logTRI	-.829	1.000	.982	.107	.043
	logVER	-.824	.982	1.000	.122	.046
	logGEO	-.345	.107	.122	1.000	.429
	logTEX	-.030	.043	.046	.429	1.000
Sig. (1-tailed)	logFRm	.	.000	.000	.000	.380
	logTRI	**.000**	.	.000	.139	.332
	logVER	**.000**	.000	.	.108	.320
	logGEO	**.000**	.139	.108	.	.000
	logTEX	.380	.332	.320	.000	.
N	logFRm	105	105	105	105	105
	logTRI	105	105	105	105	105
	logVER	105	105	105	105	105
	logGEO	105	105	105	105	105
	logTEX	105	105	105	105	105

A summary of the results is shown in Table 6.9. The values of R range from 0 to 1. Larger values of R indicate stronger relationships. R squared is the proportion of variation in the dependent variable explained by the regression model. The values of R squared range from 0 to 1 as well. Small values indicate that the model does not fit the data well. The sample R squared tends to optimistically estimate how well the models fit the population. The R squared value which is 0.771 means it is closer to 1 and is considered a good fit.

TABLE 6.9. Dual Processor Workstation (Mono) Model Summary

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	.878(a)	.771	.762	.18499

a Predictors: (Constant), logTEX, logTRI, logGEO, logVER

Table 6.10 shows the analysis of variance. It has a high regression sum of squares in comparison to the residual sum of squares. Therefore, it means that the model accounts for most of the variation in the dependent variable. The significance value of the F statistic is small (smaller than 0.05) which means the independent variables do a good job of explaining the variation in the dependent variable.

TABLE 6.10. Dual Processor Workstation (Mono) ANOVA(b)

Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	11.502	4	2.875	84.023	**.000(a)**
	Residual	3.422	100	.034		
	Total	14.924	104			

a Predictors: (Constant), logTEX, logTRI, logGEO, logVER
b Dependent Variable: logFRm

Table 6.11 is crucial in constructing an equation for this hardware system. It is from there that the whole formula is developed for prediction of performance. The significant variable is geometry count.

TABLE 6.11. Dual Processor Workstation (Mono) Coefficients(a)

Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.
		B	Std. Error	Beta	B	Std. Error
1	(Constant)	3.870	.167		23.228	.000
	logTRI	-.362	.144	-.645	-2.521	.013
	logVER	-.088	.143	-.159	-.619	.537
	logGEO	-.152	.026	-.318	-5.942	**.000**
	logTEX	.299	.112	.141	2.661	.009

a Dependent Variable: logFRm

The regression equation derived from the analysis is:

(7.2) Log (Mono Frame Rate) = 3.870 – 0.362 Log (Triangle Count) – 0.088 Log (Vertex Count) – 0.152 Log (Geometry Count) + 0.299 Log (Texture Count)

(iii) Single Processor Workstation

From Table 6.12, it is shown that there are 105 samples in the experiment for all variables. The mean logarithm of frame rate is 1.6211 with a standard deviation of 0.43857.

TABLE 6.12. Single Processor Workstation Descriptive Statistics

	Mean	Std. Deviation	N
logFRpp	1.6211	.43857	105
logTRI	4.9994	.67427	105
logVER	4.8190	.68064	105
logGEO	1.7677	.78982	105
logTEX	.9745	.17860	105

From Table 6.13, it is clear that frame rate is highly correlated to triangle count, vertex count and geometry count because the significance level or p-value is less than 0.05. All the variables are negatively correlated to frame rate as indicated by the Pearson Correlation coefficient value.

TABLE 6.13. Single Processor Workstation Correlations

		logFRpp	logTRI	logVER	logGEO	logTEX
Pearson Correlation	logFRpp	1.000	-.789	-.790	-.555	-.131
	logTRI	-.789	1.000	.982	.107	.043
	logVER	-.790	.982	1.000	.122	.046
	logGEO	-.555	.107	.122	1.000	.429
	logTEX	-.131	.043	.046	.429	1.000
Sig. (1-tailed)	logFRpp	.	.000	.000	.000	.092
	logTRI	**.000**	.	.000	.139	.332
	logVER	**.000**	.000	.	.108	.320
	logGEO	**.000**	.139	.108	.	.000
	logTEX	.092	.332	.320	.000	.
N	logFRpp	105	105	105	105	105
	logTRI	105	105	105	105	105
	logVER	105	105	105	105	105
	logGEO	105	105	105	105	105
	logTEX	105	105	105	105	105

A summary of the results is shown in Table 6.14. The values of R range from 0 to 1. Larger values of R indicate stronger relationships. R squared is the proportion of variation in the dependent variable explained by the regression model. The values of R squared range from 0 to 1 as well. Small values indicate that the model does not fit the data well. The sample R squared tends to optimistically estimate how well the models fit the population. The R squared value which is 0.861 means it is closer to 1 and is considered a good fit.

TABLE 6.14. Single Processor Workstation Model Summary

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	.928(a)	.861	.856	.16652

a Predictors: (Constant), logTEX, logTRI, logGEO, logVER

Table 6.15 shows the analysis of variance. It has a high regression sum of squares in comparison to the residual sum of squares. Therefore, it means that the model accounts for most of the variation in the dependent variable. The significance value of the F statistic is small (smaller than 0.05) which means the independent variables do a good job of explaining the variation in the dependent variable.

TABLE 6.15. Single Processor Workstation ANOVA(b)

Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	17.231	4	4.308	155.349	**.000(a)**
	Residual	2.773	100	.028		
	Total	20.004	104			

a Predictors: (Constant), logTEX, logTRI, logGEO, logVER
b Dependent Variable: logFRpp

Table 6.16 is crucial in constructing an equation for this hardware system. It is from there that the whole formula is developed for prediction of performance. The significant variables are triangle count, geometry count and texture count.

TABLE 6.16. Single Processor Workstation Coefficients(a)

Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.
		B	Std. Error	Beta	B	Std. Error
1	(Constant)	4.219	.150		28.131	.000
	logTRI	-.373	.129	-.573	-2.882	**.005**
	logVER	-.108	.128	-.168	-.845	.400
	logGEO	-.293	.023	-.528	-12.694	**.000**
	logTEX	.315	.101	.128	3.107	**.002**

a Dependent Variable: logFRpp

The regression equation derived from the analysis is:

(7.3) Log (VLabPreProc Frame Rate) = 4.219 – 0.373 Log (Triangle Count) – 0.108 Log (Vertex Count) – 0.293 Log (Geometry Count) + 0.315 Log (Texture Count)

(iv) Desktop

From Table 6.17, it is shown that there are 105 samples in the experiment for all variables. The mean logarithm of frame rate is 1.5072 with a standard deviation of 0.35493.

TABLE 6.17. Desktop Descriptive Statistics

	Mean	Std. Deviation	N
logFRrc	1.5072	.35493	105
logTRI	4.9994	.67427	105
logVER	4.8190	.68064	105
logGEO	1.7677	.78982	105
logTEX	.9745	.17860	105

From Table 6.18, it is clear that frame rate is highly correlated to triangle count and vertex count because the significance level or p-value is less than 0.05. . All the variables except texture count are negatively correlated to frame rate as indicated by the Pearson Correlation coefficient value.

TABLE 6.18. Desktop Correlations

		logFRrc	logTRI	logVER	logGEO	logTEX
Pearson Correlation	logFRrc	1.000	-.873	-.862	-.224	.044
	logTRI	-.873	1.000	.982	.107	.043
	logVER	-.862	.982	1.000	.122	.046
	logGEO	-.224	.107	.122	1.000	.429
	logTEX	.044	.043	.046	.429	1.000
Sig. (1-tailed)	logFRrc	.	.000	.000	.011	.328
	logTRI	**.000**	.	.000	.139	.332
	logVER	**.000**	.000	.	.108	.320
	logGEO	.011	.139	.108	.	.000
	logTEX	.328	.332	.320	.000	.
N	logFRrc	105	105	105	105	105
	logTRI	105	105	105	105	105
	logVER	105	105	105	105	105
	logGEO	105	105	105	105	105
	logTEX	105	105	105	105	105

A summary of the results is shown in Table 6.19. The values of R range from 0 to 1. Larger values of R indicate stronger relationships. R squared is the proportion of variation in the dependent variable explained by the regression model. The values of R squared range from 0 to 1 as well. Small values indicate that the model does not fit the data well. The sample R squared tends to optimistically estimate how well the models fit the population. The R squared value which is 0.803 means it is closer to 1 and is considered a good fit.

TABLE 6.19. Desktop Model Summary

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	.896(a)	.803	.795	.16080

a Predictors: (Constant), logTEX, logTRI, logGEO, logVER

Table 6.20 shows the analysis of variance. It has a high regression sum of squares in comparison to the residual sum of squares. Therefore, it means that the model accounts for most of the variation in the dependent variable. The significance value of the F statistic is small (smaller than 0.05) which means the independent variables do a good job of explaining the variation in the dependent variable.

TABLE 6.20. Desktop ANOVA(b)

Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	10.516	4	2.629	101.676	**.000(a)**
	Residual	2.586	100	.026		
	Total	13.102	104			

a Predictors: (Constant), logTEX, logTRI, logGEO, logVER
b Dependent Variable: logFRrc

Table 6.21 is crucial in constructing an equation for this hardware system. It is from there that the whole formula is developed for prediction of performance. The significant variables are triangle count, geometry count and texture count.

TABLE 6.21. Desktop Coefficients(a)

Mode l		Unstandardized Coefficients		Standardized Coefficients	t	Sig.
		B	Std. Error	Beta	B	Std. Error
1	(Constant)	3.599	.145		24.850	.000
	logTRI	-.430	.125	-.817	-3.444	**.001**
	logVER	-.022	.124	-.042	-.177	.860
	logGEO	-.091	.022	-.203	-4.101	**.000**
	logTEX	.335	.098	.168	3.423	**.001**

a Dependent Variable: logFRrc

The regression equation derived from the analysis is:

(7.4) Log (Research Comp Frame Rate) = 3.599 – 0.430 Log (Triangle Count) – 0.022 Log (Vertex Count) – 0.091 Log (Geometry Count) + 0.335 Log (Texture Count)

(v) Laptop

From Table 6.22, it is shown that there are 105 samples in the experiment for all variables. The mean logarithm of frame rate is 1.4301 with a standard deviation of 0.24618.

TABLE 6.22. Laptop Descriptive Statistics

	Mean	Std. Deviation	N
logFRlap01	1.4301	.24618	105
logTRI	4.9994	.67427	105
logVER	4.8190	.68064	105
logGEO	1.7677	.78982	105
logTEX	.9745	.17860	105

From Table 6.23, it is clear that frame rate is highly correlated to triangle count, vertex count and geometry count because the significance level or p-value is less than 0.05. All the variables are negatively correlated to frame rate as indicated by the Pearson Correlation coefficient value.

TABLE 6.23. Laptop Correlations

		logFRlap01	logTRI	logVER	logGEO	logTEX
Pearson Correlation	logFRlap01	1.000	-.650	-.657	-.583	-.069
	logTRI	-.650	1.000	.982	.107	.043
	logVER	-.657	.982	1.000	.122	.046
	logGEO	-.583	.107	.122	1.000	.429
	logTEX	-.069	.043	.046	.429	1.000
Sig. (1-tailed)	logFRlap01	.	.000	.000	.000	.242
	logTRI	**.000**	.	.000	.139	.332
	logVER	**.000**	.000	.	.108	.320
	logGEO	**.000**	.139	.108	.	.000
	logTEX	.242	.332	.320	.000	.
N	logFRlap01	105	105	105	105	105
	logTRI	105	105	105	105	105
	logVER	105	105	105	105	105
	logGEO	105	105	105	105	105
	logTEX	105	105	105	105	105

A summary of the results is shown in Table 6.24. The values of R range from 0 to 1. Larger values of R indicate stronger relationships. R squared is the proportion of variation in the dependent variable explained by the regression model. The values of R squared range from 0 to 1 as well. Small values indicate that the model does not fit the data well. The sample R squared tends to optimistically estimate how well the models fit the population. The R squared value which is 0.731 means it is closer to 1 and is considered a good fit.

TABLE 6.24. Laptop Model Summary

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	.855(a)	.731	.720	.13033

a Predictors: (Constant), logTEX, logTRI, logGEO, logVER

Table 6.25 shows the analysis of variance. It has a high regression sum of squares in comparison to the residual sum of squares. Therefore, it means that the model accounts for most of the variation in the dependent variable. The significance value of the F statistic is small (smaller than 0.05) which means the independent variables do a good job of explaining the variation in the dependent variable.

TABLE 6.25. Laptop ANOVA(b)

Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	4.604	4	1.151	67.771	**.000(a)**
	Residual	1.698	100	.017		
	Total	6.303	104			

a Predictors: (Constant), logTEX, logTRI, logGEO, logVER
b Dependent Variable: logFRlap01

Table 6.26 is crucial in constructing an equation for this hardware system. It is from there that the whole formula is developed for prediction of performance. The significant variables are texture count and geometry count.

TABLE 6.26. Laptop Coefficients(a)

Mode l		Unstandardized Coefficients		Standardized Coefficients	t	Sig.
		B	Std. Error	Beta	B	Std. Error
1	(Constant)	2.542	.117		21.654	.000
	logTRI	-.116	.101	-.318	-1.146	.255
	logVER	-.102	.100	-.281	-1.012	.314
	logGEO	-.190	.018	-.608	-10.493	**.000**
	logTEX	.301	.079	.219	3.802	**.000**

a Dependent Variable: logFRlap01

The regression equation derived from the analysis is:

(7.5) Log (Laptop Frame Rate) = 2.542 – 0.116 Log (Triangle Count) – 0.102 Log (Vertex Count) – 0.190 Log (Geometry Count) + 0.301 Log (Texture Count)

(v) Core 2 Duo Laptop

From Table 6.27, it is shown that there are 105 samples in the experiment for all variables. The mean logarithm of frame rate is 1.6451 with a standard deviation of 0.16741.

TABLE 6.27. Duo Core Laptop Descriptive Statistics

	Mean	Std. Deviation	N
logFRlap02	1.6451	.16741	105
logTRI	4.9994	.67427	105
logVER	4.8190	.68064	105
logGEO	1.7677	.78982	105
logTEX	.9745	.17860	105

From Table 6.28, it is clear that frame rate is highly correlated to triangle count, vertex count and geometry count because the significance level or p-value is less than 0.05. All the variables are negatively correlated to frame rate as indicated by the Pearson Correlation coefficient value.

TABLE 6.28. Duo Core Laptop Correlations

		logFRlap 02	logTRI	logVER	logGEO	logTEX
Pearson Correlation	logFRlap02	1.000	-.386	-.413	-.730	-.097
	logTRI	-.386	1.000	.982	.107	.043
	logVER	-.413	.982	1.000	.122	.046
	logGEO	-.730	.107	.122	1.000	.429
	logTEX	-.097	.043	.046	.429	1.000
Sig. (1-tailed)	logFRlap02	.	.000	.000	.000	.161
	logTRI	**.000**	.	.000	.139	.332
	logVER	**.000**	.000	.	.108	.320
	logGEO	**.000**	.139	.108	.	.000
	logTEX	.161	.332	.320	.000	.
N	logFRlap02	105	105	105	105	105
	logTRI	105	105	105	105	105
	logVER	105	105	105	105	105
	logGEO	105	105	105	105	105
	logTEX	105	105	105	105	105

A summary of the results is shown in Table 6.29. The values of R range from 0 to 1. Larger values of R indicate stronger relationships. R squared is the proportion of variation in the dependent variable explained by the regression model. The values of R squared range from 0 to 1 as well. Small values indicate that the model does not fit the data well. The sample R squared tends to optimistically estimate how well the models fit the population. The R squared value of 0.698 is not as good a fit as other models but is still closer to 1 and is considered a good enough fit.

TABLE 6.29. Duo Core Laptop Model Summary

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	.836(a)	.698	.686	.09378

a Predictors: (Constant), logTEX, logTRI, logGEO, logVER

Table 6.30 shows the analysis of variance. It has a high regression sum of squares in comparison to the residual sum of squares. Therefore, it means that the model accounts for most of the variation in the dependent variable. The significance value of the F statistic is small (smaller than 0.05) which means the independent variables do a good job of explaining the variation in the dependent variable.

TABLE 6.30. Duo Core Laptop ANOVA(b)

Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	2.035	4	.509	57.846	**.000(a)**
	Residual	.880	100	.009		
	Total	2.915	104			

a Predictors: (Constant), logTEX, logTRI, logGEO, logVER
b Dependent Variable: logFRlap02

Table 6.31 is crucial in constructing an equation for this hardware system. It is from there that the whole formula is developed for prediction of performance. The significant variables are texture count and geometry count.

TABLE 6.31. Duo Core Laptop Coefficients(a)

Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.
		B	Std. Error	Beta	B	Std. Error
1	(Constant)	2.071	.084		24.519	.000
	logTRI	.075	.073	.300	1.024	.308
	logVER	-.153	.072	-.623	-2.119	.037
	logGEO	-.169	.013	-.798	-13.009	**.000**
	logTEX	.245	.057	.261	4.293	**.000**

a Dependent Variable: logFRlap02

The regression equation derived from the analysis is:

(7.6) Log (Laptop Frame Rate) = 2.07 – 0.075 Log (Triangle Count) – 0.153 Log (Vertex Count) – 0.169 Log (Geometry Count) + 0.245 Log (Texture Count)

From the results above, the value of each variable in the equation can be substituted to predict the frame rate from the equations. All the results are quite consistent except for the last result on the duo core laptop. The most important matter is that the R squared value indicates that the models fit the data well, which means it fits the population of the samples well. The goodness of fit is weakest for the duo core laptop, which is at 0.698 probably because it is a new hardware system and is not configured to run on a stable basis with the VR software. Therefore, the accuracy of prediction also depends on the R squared value. The closer it is to the value of 1.0, the more accurate the prediction is going to be. At this stage, the prediction for the frame rate of a VR simulation running on the duo core laptop can be expected to be not accurate at all.

6.2 Ranking of Experimental Variables

In order to rank the variables from the greatest to the weakest in terms of their contribution to performance, there is need to first describe them all on the same scale. This is because all variables are running at different ranges.

(i) Triangle count has the lowest amount at 470 and the highest amount at 1,919,531. This gives an average of 959,530.5.

(ii) Vertex count has the lowest amount at 258 and the highest amount at 1,225,431. This gives an average of 612,586.5.

(iii) Geometry count has the lowest amount at 3 and the highest amount at 5365. This gives an average of 2681.

(iv) Texture count has the lowest amount at 3 and the highest amount at 20. This gives an average of 8.5.

(v) Triangle count (stereo) has the lowest amount at 940 and the highest amount at 3,839,062. This gives an average of 1,919,061.

The LOG10 for all the averages are taken to give us the following values:

(i) Triangle count 5.9821

(ii) Vertex count 5.7872

(iii) Geometry count 3.4283

(iv) Texture count 0.9294

(v) Triangle count (stereo) 6.2831

The values are multiplied with the coefficient of each variable for each hardware system to show the final contribution of each variable.

(i) Dual Processor Workstation (Stereo)

The regression equation from the analysis is:

Log (Stereo Frame Rate) =4.176 – 0.485 Log (Triangle Count) – 0.068 Log (Vertex Count) – 0.159 Log (Geometry Count) + 0.298 Log (Texture Count)

Triangle Count (stereo) = (6.2831 X -0.485) = -3.047

Vertex Count = (5.7872 X- 0.068) = -0.3935

Geometry Count = (3.4283 X -0.159) = -0.5451

Texture Count = (0.9294 X 0.298) = 0.2790

Therefore, it is clear that triangle count is the biggest contributor, followed by geometry count, vertex count and texture count.

(ii) Dual Processor Workstation (Mono)

The regression equation from the analysis is:

Log (Mono Frame Rate) = 3.870 – 0.362 Log (Triangle Count) – 0.088 Log (Vertex Count) – 0.152 Log (Geometry Count) + 0.299 Log (Texture Count)

Triangle Count = (5.9821 X -0.362) = -2.1655

Vertex Count = (5.7872 X -0.088) = -0.5093

Geometry Count = (3.4283 X -0.152) = -0.5211

Texture Count = (0.9294 X 0.299) = 0.2779

Therefore, it is clear that triangle count is the biggest contributor, followed by geometry count, vertex count and texture count.

(iii) Single Processor Workstation

The regression equation from the analysis is:

Log (VLabPreProc Frame Rate) = 4.219 – 0.373 Log (Triangle Count) – 0.108 Log (Vertex Count) – 0.293 Log (Geometry Count) + 0.315 Log (Texture Count)

Triangle Count = (5.9821 X -0.373) = -2.2313

Vertex Count = (5.7872 X -0.108) = -0.6250

Geometry Count = (3.4283 X -0.293) = -1.0045

Texture Count = (0.9294 X 0.315) = 0.2928

Therefore, it is clear that triangle count is the biggest contributor, followed by geometry count, vertex count and texture count.

(iv) Desktop

The regression equation from the analysis is:

Log (Research Comp Frame Rate) = 3.599 – 0.430 Log (Triangle Count) – 0.022 Log (Vertex Count) – 0.091 Log (Geometry Count) + 0.335 Log (Texture Count)

Triangle Count = (5.9821 X -0.430) = -2.5723

Vertex Count = (5.7872 X -0.022) = -0.1273

Geometry Count = (3.4283 X -0.091) = -0.3120

Texture Count = (0.9294 X 0.335) = 0.3113

Therefore, it is clear that triangle count is the biggest contributor, followed by geometry count, texture count and vertex count.

(v) Laptop

The regression equation from the analysis is:

Log (Laptop Frame Rate) = 2.542 – 0.116 Log (Triangle Count) – 0.102 Log (Vertex Count) – 0.190 Log (Geometry Count) + 0.301 Log (Texture Count)

Triangle Count = (5.9821 X -0.116) = -0.6940

Vertex Count = (5.7872 X -0.102) = -0.5903

Geometry Count = (3.4283 X -0.190) = -0.6514

Texture Count = (0.9294 X 0.301) = 0.2798

Therefore, it is clear that triangle count is the biggest contributor, followed by geometry count, vertex count and texture count.

(vi) Duo Core Laptop

The regression equation from the analysis is:

Log (Laptop Frame Rate) = 2.07 – 0.075 Log (Triangle Count) – 0.153 Log (Vertex Count) – 0.169 Log (Geometry Count) + 0.245 Log (Texture Count)

Triangle Count = (5.9821 X -0.075) = -0.4487

Vertex Count = (5.7872 X -0.153) = -0.8854

Geometry Count = (3.4283 X -0.169) = -0.5793

Texture Count = (0.9294 X 0.245) = 0.2277

Therefore, it is clear that vertex count is the biggest contributor, followed by geometry count, triangle count and texture count.

In conclusion, triangle count and geometry count are the biggest contributors to VR simulation performance as shown in Table 6.32. The only inconsistency happens in the brand new duo core laptop, in which vertex count is the biggest contributor. This is because of the new Windows® Vista^{TM} operating system which is not yet very stable. The many difficulties of running most programs in the new operating system could be an issue at this point in time, as these VR software were not meant to run in Windows®

Vista^{TM} in the first place. Another crucial point is that the Core 2 Duo is the only hardware platform that uses a 64-bit computer chip. Since all the programs (i.e. 3ds Max® and EON^{TM} Studio^{TM} Professional^{TM}) are meant to run on a 32-bit computer, the incompatibility issue will always be there. The ranking of vertex and texture is interchangeable for the desktop system. This is fine since the contributions of both texture and vertex variables are so mild that it does not really matter which is third and fourth.

TABLE 6.32. Experiment Variables Ranking

Ranking \ Hardware	**Dual Processor Workstation (Stereo) WinXP**	**Dual Processor Workstation (Mono) WinXP**	**Single Processor Workstation WinXP**	**Desktop WinXP**	**Laptop Win XP**	**Core 2 Duo Laptop Win Vista**
Triangle	1	1	1	1	1	3
Geometry	2	2	2	2	2	2
Vertex	3	3	3	4	3	1
Texture	4	4	4	3	4	4

6.3 Verification of Experimental Results

To prove that the equation derived from Chapter 6.1 to predict frame rate works, a verification is shown in this section. The frame rate of a 3D model is predicted using the equation. Then, the 3D model is exported to EON^{TM} Studio^{TM} Professional^{TM} and a VR simulation is run. The frame rate is then measured and compared with the predicted frame rate.

The 3D model chosen for verification is the unbuilt Geoffrey Bawa's Bali Hyatt project which was built for a research project as shown in Figure 6.1. The model was not used as a sample in the simple linear multiple regression calculation earlier. Therefore, it is fair to use it for verification of the equation. For most 3D visualisation software, the information about the amount of triangles, vertices, geometries and textures can be found as they are normally calculated.

Figure 6.1. Unbuilt Bali Hyatt

The project has 234,812 triangles, 141,394 vertices, 1065 geometries and 5 textures. Since this is tested on the desktop, the regression equation for the desktop is chosen to predict performance. The values of triangles, vertices, geometries and textures are substituted into the equation.

The regression equation from Chapter 6.1 is:

$$\text{Log (Research Comp Frame Rate)} = 3.599 - 0.430\ \text{Log (Triangle Count)} - 0.022\ \text{Log (Vertex Count)} - 0.091\ \text{Log (Geometry Count)} + 0.335\ \text{Log (Texture Count)}$$

(i) The amount of vertex count is 141,394. Log it to get the value 5.1504. Multiply that with the coefficient 0.022.

(5.1504)X 0.022 = 0.1133

(ii) The amount of geometry count is 1065. Log it to get the value 3.027. Multiply that with the coefficient 0.091.

(3.027) X 0.091 = 0.2755

(iii) The amount of texture count is 5. Log it to get the value 0.6990. Multiply that with the coefficient 0.335.

(0.6990) X 0.335 = 0.2342

(iv) The amount of triangles count is 234,812. Log it to get the value 5.3707. Multiply that with the coefficient 0.430.

(5.3707) X 0.430 = 2.3094

Substitute all the figures into the equation.

Log (Research Comp Frame Rate) = 3.599 – 0.1133 – 0.2755 + 0.2342 - 2.3094

= 1.135.

The result is 1.135. Antilog it to get an estimate of 13.6 fps. The actual frame rate from the simulation run in EON^{TM} Studio^{TM} Professional^{TM} is an average 12.9 fps. The estimated figure is very close to the actual figure.

6.4 Survey Results

The survey results show some important trends which are as follows:

(i) The expectations from people with architectural background who have 3D game (first person) experience are highest. The reason for this is very clear because these people have been through the evolution of 3D games and they remarked that those low level triangle count, texture and frame rate resemble ancient games of the past. Most 3D gamers are male and they are normally of the lower age range of below 25 years old.

(ii) The expectations from people who have 3D game (first person) experience and are not from architecture is second highest.

(iii) People from architecture without any 3D game (first person) experience come third in their quality expectations.

(iv) The expectations from people who are not in any of the above categories are lowest.

(v) Some people who have no 3D game (first person) experience complain of cyber sickness and have problems with the focus of 3D models projected on the screen. This is very clear in people who is at the upper limit of the age group, which is 30-31 years old. People around the age of 30 and above are from the generation whose exposure to computers and console gaming is low or at least were exposed very late in their lives. The younger generations are more exposed to the latest games as well as computers and consoles. It is therefore not surprising to get such results.

(vi) Almost everyone remarked that the last simulation on different frame rate count slows down as the frame rate decreases. This is the same as one of the hypotheses of the thesis where time taken seems longer when the complexity of 3D models increases. In actual fact, time taken is the same regardless of frame rate. People do not realise that it is the number of frames which are lessened from each choice of the 3D scene they are comparing while the time taken is the same.

The profiles of the subjects who participated in the survey are as follow in Figure 6.2, 6.3 and 6.4.

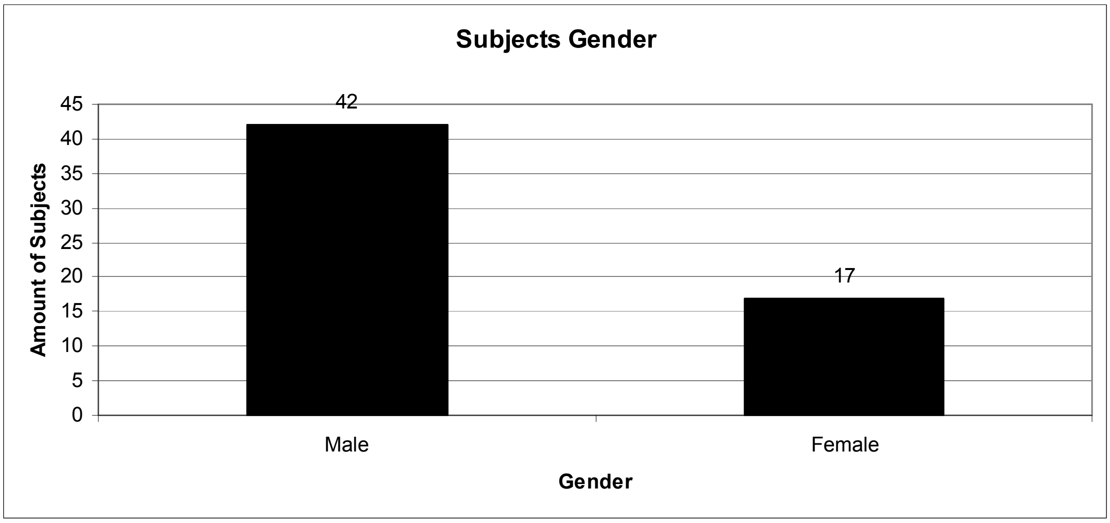


Figure 6.2. Subjects Gender

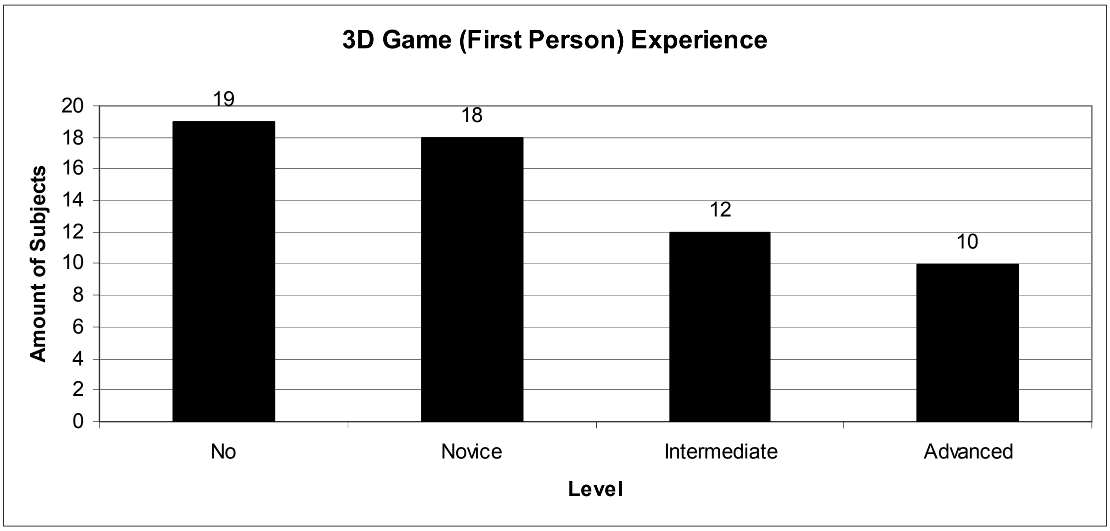


Figure 6.3. 3D Games (First Person) Experience

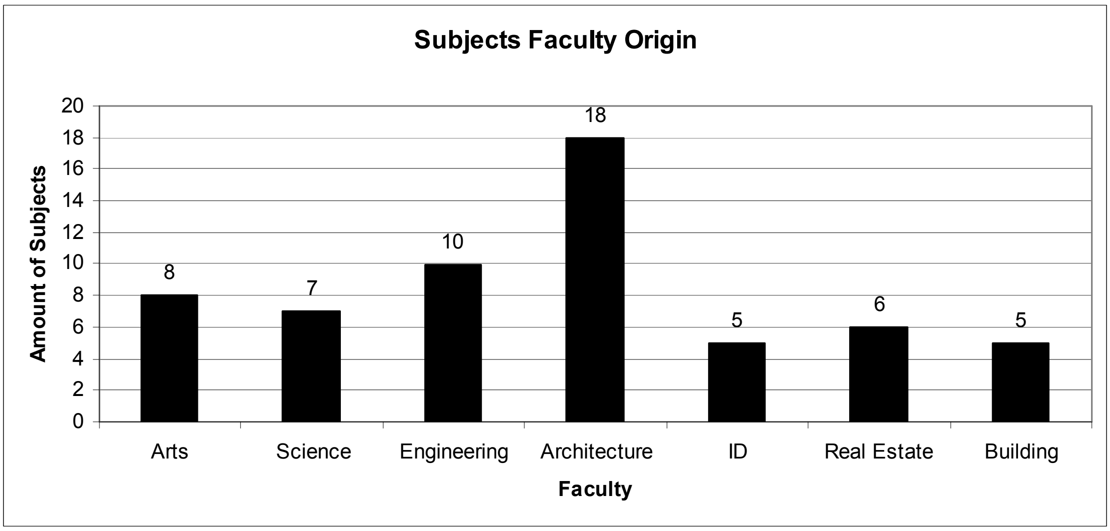


Figure 6.4. Subjects Faculty Origin

The response to each question is shown as follows:

For Question 1, the most popular choice is Chair D, which is selected by 15 people as shown in Figure 6.5. They cannot accept its quality, which has 35% of the total amount of 69,854 triangles. Chair B, which is 55% of the total amount and Chair C, which is 45% of the total amount are selected by six and seven people respectively.

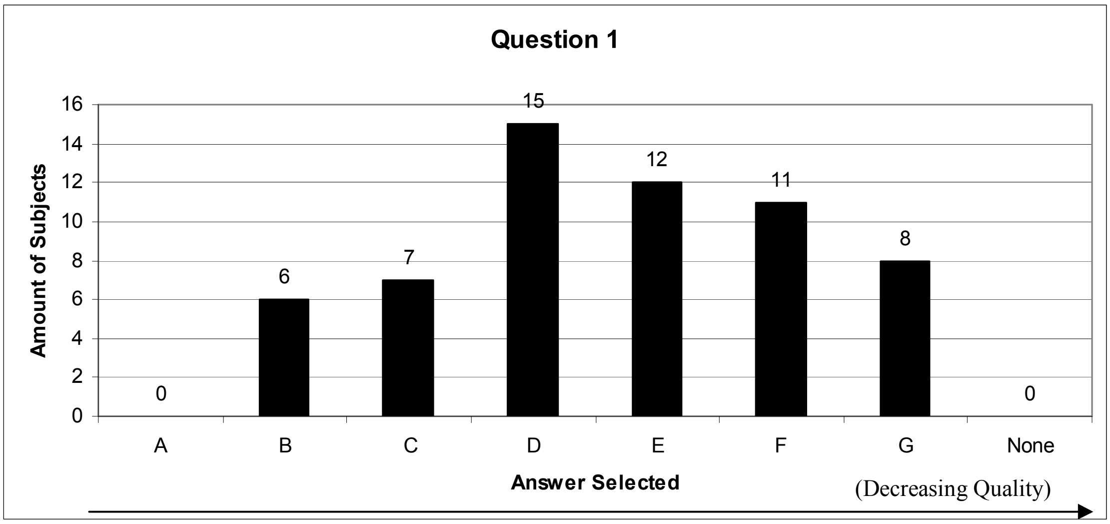


Figure 6.5. Question 1 Responses (Chair Quality)

For Question 2, the most popular choice is Sink E as shown in Figure 6.6. Fourteen people choose it, which is 25% of the total 69,482 triangles. Six people choose Sink B (55%), eight people choose Sink C (45%) and 11 people choose Sink D (35%).

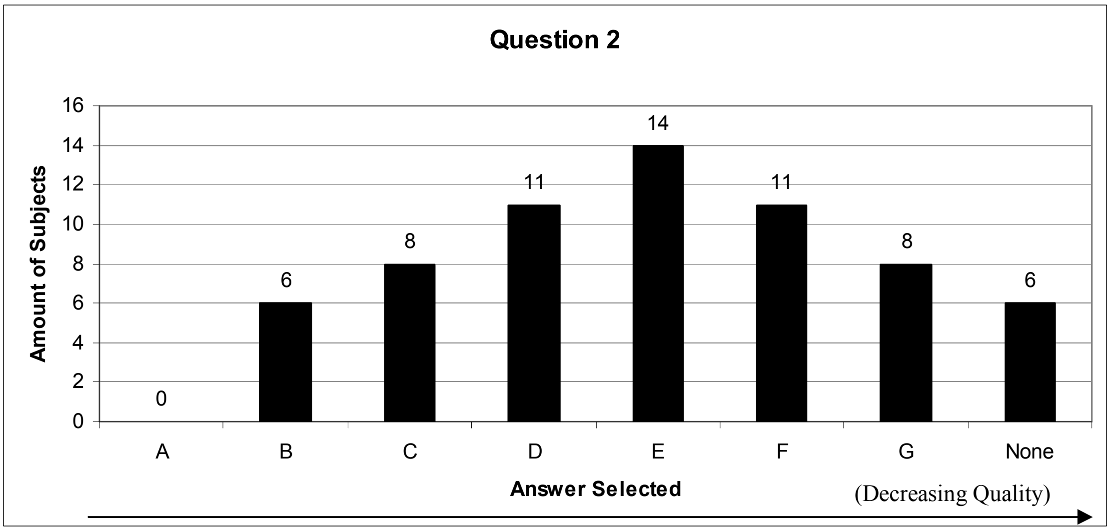


Figure 6.6. Question 2 Responses (Sink Quality)

For Question 3, the most popular choice is Table D with 23 people choosing it as shown in Figure 6.7. Table D is a 512 X 512 pixel texture. There are eight people who have higher requirements in choosing Table C, which has 1024 X 1024 pixel texture.

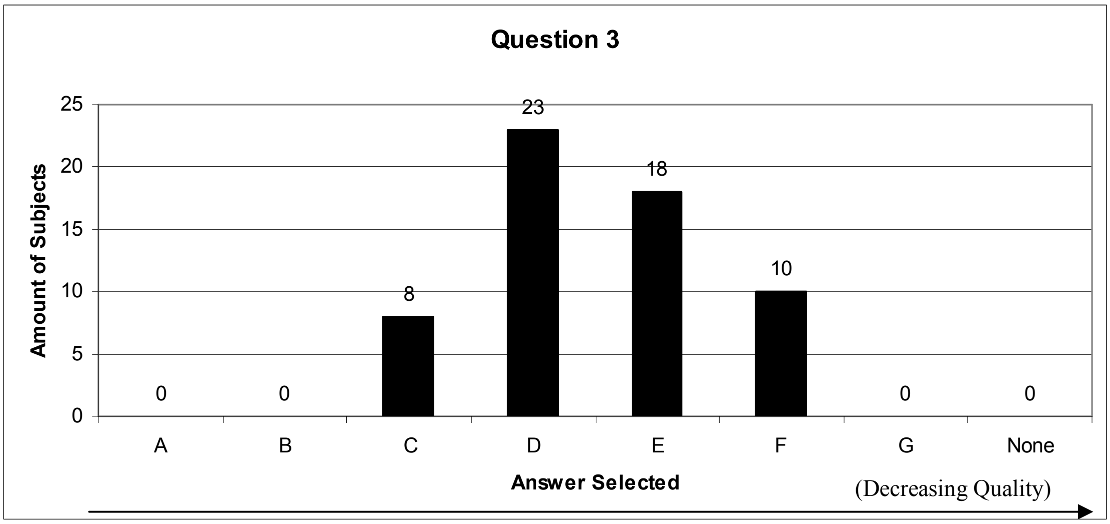


Figure 6.7. Question 3 Responses (Table Texture Quality)

For Question 4, the most popular choice is Curved Walls E, which has a 256 X 256 pixel texture as shown in Figure 6.8. There are 28 people choosing it. Apart from that, there are eight people choosing Curved Walls C (1024 X 1024 pixel texture) and 14 people choosing Curved Walls D (512 X 512 pixel texture).

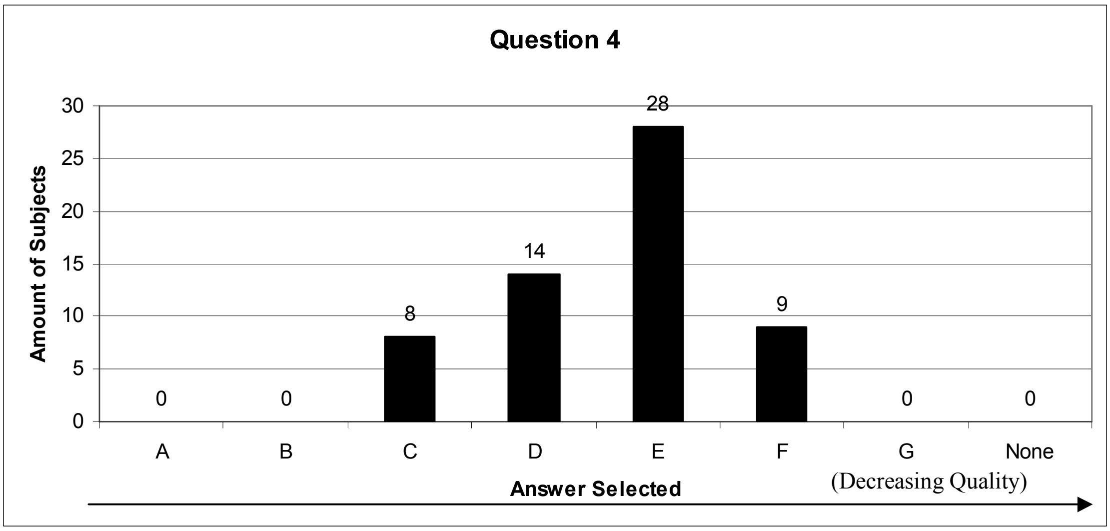


Figure 6.8. Question 4 Responses (Curved Walls Quality)

For Question 5, there are more divided views but the most popular choice is Rock F (128 X 128 pixel texture) with 20 people choosing it as shown in Figure 6.9. There are five people choosing Stone B (2058 X 2058 pixel texture), six people choosing Rock C (1024 X 1024 pixel texture), six people choosing Rock D (512 X 512 pixel texture) and nine people choosing Rock E (256 X 256 pixel texture).

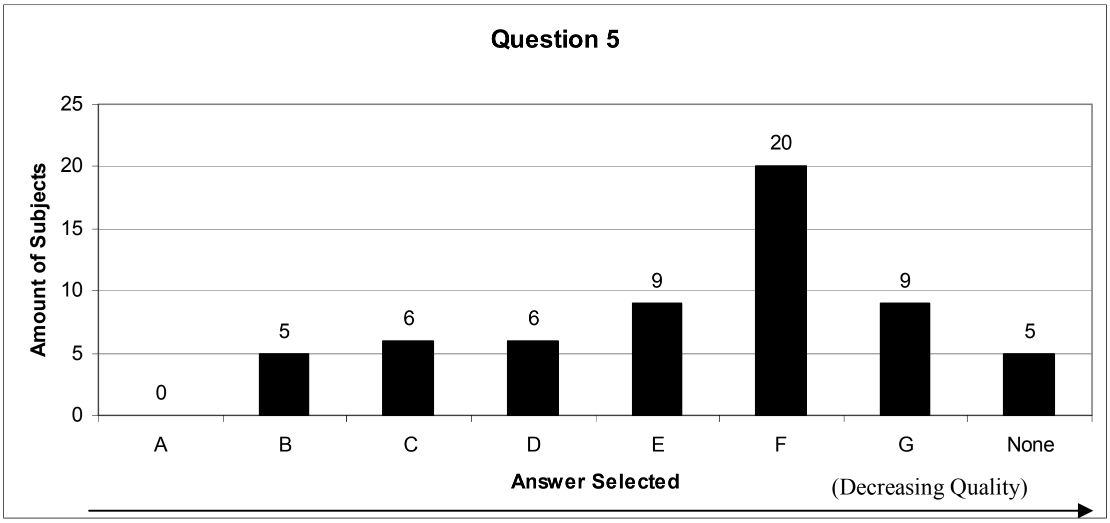


Figure 6.9. Question 5 Responses (Stone Texture Quality)

For Question 6, it is clear that the most popular choice is Simulation C (12Hz) with 17 people, followed by Simulation D (10Hz) with 16 people and Simulation E (8Hz) with 11 people as shown in Figure 6.10.

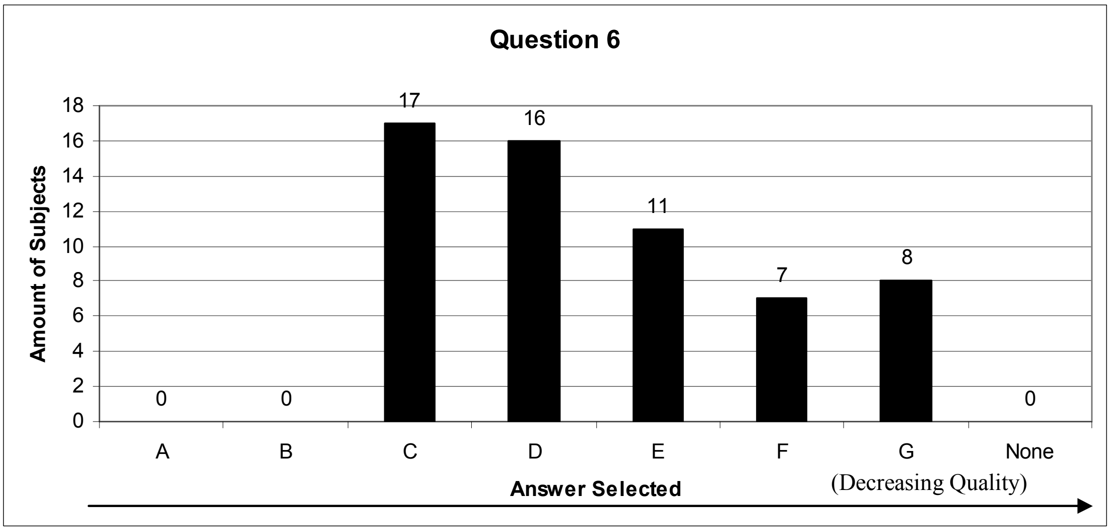


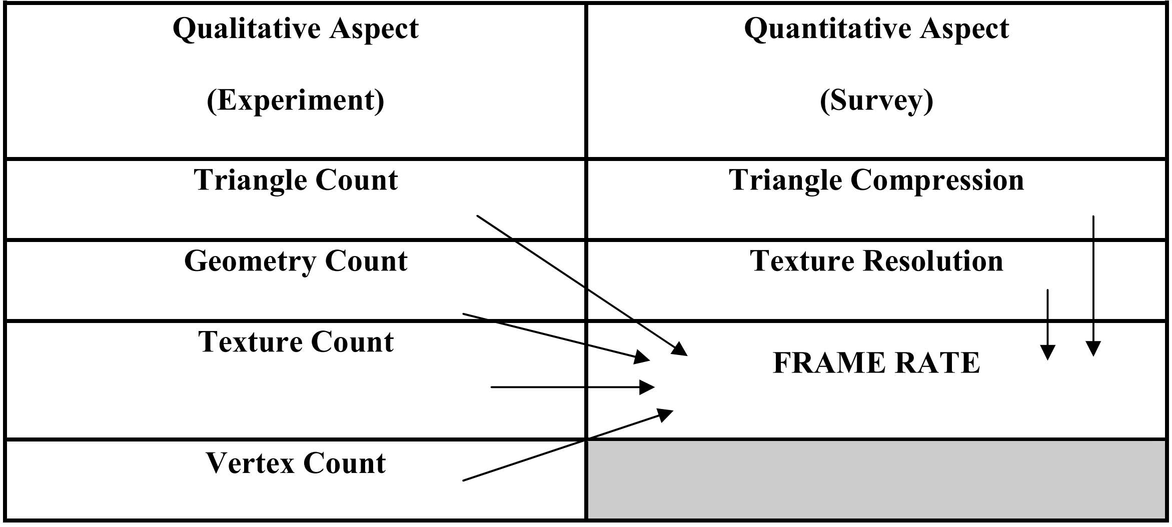
Figure 6.10. Question 6 Responses (Frame Rate Count)

In conclusion, it is clear that the subjects can accept a triangle compression of 65-75% for organic objects. Subjects can accept 1024 X 1024 pixel resolution textures for flat surfaces, 512 X 512 pixel resolutions for curved surfaces and 256 X 256 pixel resolutions for small rough surfaces. Fifteen Hz is the acceptable frame rate, therefore anything at 15Hz and above is preferable for visualisation. The qualitative results are helpful in integrating into the final guideline, areas where you cannot measure using statistics and computers. These preferences by human beings are crucial in preparing simulations with efficient enough quality to satisfy their requirements and quality acceptance of how things should look in real life and at the same time run in the fastest possible way on the hardware platforms.

6.5 Relationship Between Qualitative and Quantitative Aspects

From both the experiment and survey, key relationships between the two can be found as shown in Table 6.33. The frame rate preference measured in the survey relates to everything and is definitely the most important measurement of performance. Anything created in the VR visualisation will influence it. The increase or decrease of triangle count, geometry count, texture count, vertex count, triangle compression or texture resolution will affect frame rate.

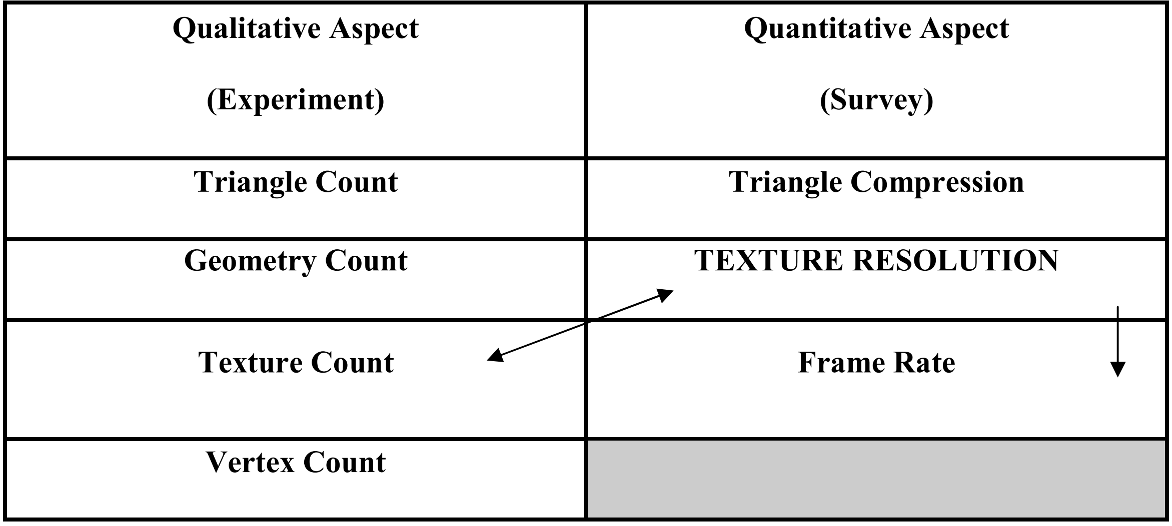
TABLE 6.33. Frame Rate’s Relationships



Qualitative Aspect (Experiment)	Quantitative Aspect (Survey)
Triangle Count	**Triangle Compression**
Geometry Count	**Texture Resolution**
Texture Count	**FRAME RATE**
Vertex Count	

As for texture resolution, it affects frame rate as mentioned in Table 6.34. Apart from that, it is also influenced by texture count. When a scene has very minimum texture counts or when a large surface represents one texture, its texture resolution can be increased. When there are too many texture counts in the scene, the texture resolution should be minimized. Therefore, the relationship goes both ways. Most of the time, this depends on the texture available and the requirements of the project.

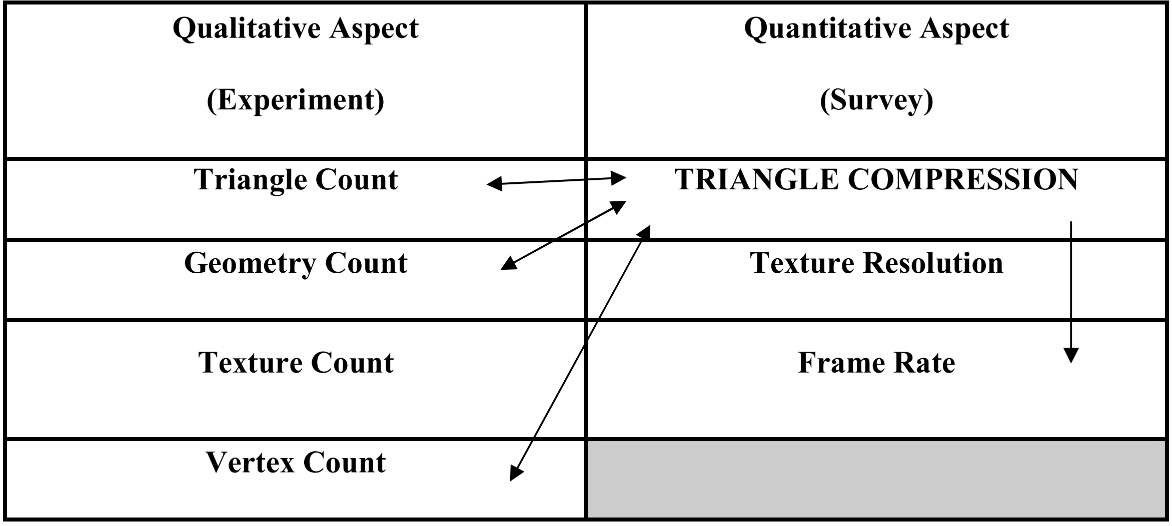
TABLE 6.34. Texture Resolution’s Relationships



Qualitative Aspect (Experiment)	Quantitative Aspect (Survey)
Triangle Count	**Triangle Compression**
Geometry Count	**TEXTURE RESOLUTION**
Texture Count	**Frame Rate**
Vertex Count	

As for triangle compression, it affects frame rate as mentioned in Table 6.33. The more the compression is done, the less the triangle count, geometry count and vertex count is generated. On the other hand, the less the triangle count, geometry count and vertex count is, the less triangle compression is required. The relationship goes both ways.

TABLE 6.35. Triangle Compression's Relationships



Qualitative Aspect **(Experiment)**	**Quantitative Aspect** **(Survey)**
Triangle Count	**TRIANGLE COMPRESSION**
Geometry Count	**Texture Resolution**
Texture Count	**Frame Rate**
Vertex Count	

The understanding of these relationships is important because when a VR visualisation preparation is planned for, at whichever stage, what can and cannot be done which will affect both the qualitative and quantitative aspects will be known. It is a smart way to kill two birds with one stone because a simple decision made can satisfy both criterias.

CHAPTER 7.0: CONCLUSION

Real-time rendering of VR architectural visualisation is affected by many things. The optimisation techniques must be used to prepare the simulations in the most efficient way possible in terms of effort and time spent. The results have shown that triangle count is the biggest contributing factor in affecting performance and this is followed closely by geometry count. Vertex count and texture count have very weak influences on the performance and should not be made top priorities. It is also shown that hardware does affect overall performance.

The survey clearly indicated that users can accept up to 65-75% of optimisation of organic 3D models. They accept 1024 X 1024 pixels resolution textures for flat surfaces, 512 X 512 pixels for curve surfaces and 256 X 256 for small surfaces. The survey also clearly indicates that 15Hz or 15fps is acceptable for the simulation.

7.1 Comparison Against Hypotheses

As for the hypotheses,

(i) Triangle count is the variable which affects VR performance the most. This is speculated from observations of different VR projects created for presentation.

The first hypothesis is true as proven in the final experiment where triangle count affects VR performance the most with geometry count coming in second. As was observed in the VR simulation, the more complex a 3D project, the slower the system becomes. It is proven true in the final experiment where triangle count contributes most to all hardware system performance except the Windows® Vista^{TM} system. It requires more testing as the Windows Vista operating system is not meant to run the VR software in the first place and is not stable yet. Normally it requires service pack upgrades to become more stabilized.

(ii) Time taken to navigate from one position to the other in a VR simulation will increase when 3D models get larger

The second hypothesis is proven wrong during the independent variable test on tree count. The time taken to travel a distance is the same regardless of how complex the 3D scene is. Earlier before calculations were done, there is a sense that simulations get slower and slower when the 3D scenes get more and more complicated. It is proven not so during the independent variable tests involving trees. During the survey however, the users gave the same remarks again regarding the frame rate test. They do not know that the frame rate is decreased from scene to scene. Most of them feel that the time taken is getting longer as frame rate decreases. Therefore a decrease of frame rate gives the perception that simulations seem slower.

(iii) Only hardware will affect VR performance and not software

The third hypothesis is true as proven in the independent variable test on hardware and software comparisons. Software difference contributes little or nothing at all to performance in comparison with hardware configurations. During the independent variable test, different hardware ranges of workstation, desktop and laptop were tested to prove this hypothesis correct. Even during the final experiment, the average frame rate for all the samples taken has proven again that hardware plays a big role in performance.

(iv) Ranking of variables' weight of contribution to VR performance is the same regardless of hardware specifications.

The last hypothesis is true as proven in the final experiment as the ranking for the two largest contributing variables are the same for different hardware systems, which is triangle count followed by geometry count (see Table 6.32 p.180). For most systems, the last two variables are vertex count and texture count. There is inconsistency only in one hardware system with the vertex count and texture count ranking becoming interchangeable. Since their contributions towards performance are extremely small in comparison to triangle count and geometry count, it does not really matter which among them is third and fourth. Unexpectedly, ranking of variables are totally rearranged in the Windows® Vista^{TM} system. The explanation for this deviation is that the operating system was released way after the VR software became available in the market and therefore the VR software is not designed to run on Vista optimally. Apart from that, it is a fact that it requires one to two upgrades before an operating system is stable. Therefore, perhaps it is fairer to compare when the Service Pack 1 or Service Pack 2 of Windows® Vista^{TM} is released. Apart from that, at the point of writing, the next VR software version is in the process of being finalized for market release to support the latest operating systems.

7.2 Recommended Workflow

From both the quantitative and qualitative aspects, a guideline is produced at the end to help designers, architects or anyone who is interested to prepare an efficient 3D model for real-time VR visualisation. It is meant to give them the most efficient preparation possible with the ability to predict frame rate using the equations that are calculated earlier from the experiments.

At the start of the modelling stage, the most important thing is to make sure that every surface created should consist of a maximum of two triangles unless there is a crucial need for subdivisions to more segments due to the project's needs such as more high definition lighting, shadow effects on surfaces and finer detailing of the 3D model. The geometry or object parts of the model should not be divided unnecessarily, even if for different textures on the same object as multi textures can be done. The amount of vertices for a single triangle is 3 at the minimum and more vertices are added only if there is a need to have more points to shape an organic or terrain model.

When making copies of the same 3D object, even in different positions, rotation angles or scales, the clone instances method must be used to save resources. All faces of the 3D models must face outwards so that there will not be transparent surfaces present during the VR simulation later. Alpha channel DDS textures with power-of-2 (256, 512, 1024, 2048, 4096) in 72dpi and mipmap should be used for all surfaces with many holes such as the wire mesh and trees.

For trees, it will be useful if high resolution images are available and is not crucial to have 3D trees. All the pivot points must not overlap each other and surfaces should not be too close to each other. This will avoid flickering of surfaces during the simulation. After that, frame rate can be predicted by using the equation for the closest hardware setup (see p. 175-179 for equations). If the user has a standard Pentium IV desktop, the desktop equation should be chosen as it is closest to their specifications. Since no texture is applied yet at this stage, the designer should at least have a rough idea of how many textures will be used in the final 3D scene. Since this is just a prediction, a rough estimation is good enough. The aim is to achieve a frame rate of 15Hz and anything above that is even better.

Most 3D visualisation software like 3ds Max® has a command to calculate triangle count, geometry count and vertex count. The user can easily get the information to use in the equation. If the prediction shows a failure of achieving 15Hz, the VR 3D scene designer should consider dividing the 3D model into more parts for loading at different points of the VR simulation and compressing the organic 3D models up to 65 – 75%.

Next is the texture mapping stage whereby the user must use DDS format textures to save resources. DDS DXT1 RGB in 72 dpi and power-of-2 texture format with mipmap is sufficient for all standard textures. The survey results give a general guide for texture mapping flat, curved and small rough surfaces. Resolution of 1024 X 1024 is sufficient for flat surfaces, 512 X 512 is sufficient for curved surfaces and 256 X 256 is sufficient for small rough surfaces. For alpha channel textures, it is recommended that the DDS DXT3 ARGB (Explicit Alpha) in 72dpi, power-of-2 with mipmap texture format be used. If the size of the surface is extremely huge, the texture mapping should be done to scale it appropriately in place.

After that, a default ambient light should always be used to light up the entire 3D scene. Adding up to eight lights is possible if required by special project needs. Adding dummies in the scene for accurate positions of many repeated 3D models like trees can be done. These dummies are basically position coordinates where the tree clones can be easily placed later in the VR software.When there are animation paths present, it can be added before exporting to the VR software. The prediction can be repeated again here with one more variable added, that is texture count. If the frame rate is below 15Hz, dividing the 3D model into more parts and compressing any organic 3D models should be considered.

After exporting into the VR software, it should be ensured that the near and far clipping plane is enough to cover the entire 3D model in the camera view. In situations where the entire 3D scene will not be viewed, the range can then be shortened to cover what the navigation range can see. The rendering quality option should be set at the minimal level, at the point where no difference will occur even if it is set higher than that level. This is to avoid extra rendering computation which will not improve quality at all.

Finally, additional features for presentation such as advanced / programmable shaders, particle systems, collision detection, scripting, looped video and audio recording can be added to the 3D scene depending on projects requirements. This can be done if the frame rate is a lot higher than 15Hz. The VR 3D scene designer can add features as long as the performance will stay at 15Hz or above.

Finally, the simulation is started and if it still runs below 15Hz, the 3D model should be divided into smaller parts if it is acceptable to load parts of the model at different points during the navigation. If not, another method to consider is the level of detail (LOD) as well as optimisation of organic model parts. For organic models, optimisations can go as high as 65 - 75%. They can be exported again into the VR software after optimisation to replace the original. The last choice is to use optimisation functions provided by the VR software, which includes the choices of polygon, texture and geometry. Polygon or triangle as well as geometry count should be given priority over texture as they are proven to be the two biggest factors that affect frame rate.

Descriptions of the guideline are summarized in the guideline flowchart in Figure 7.1:

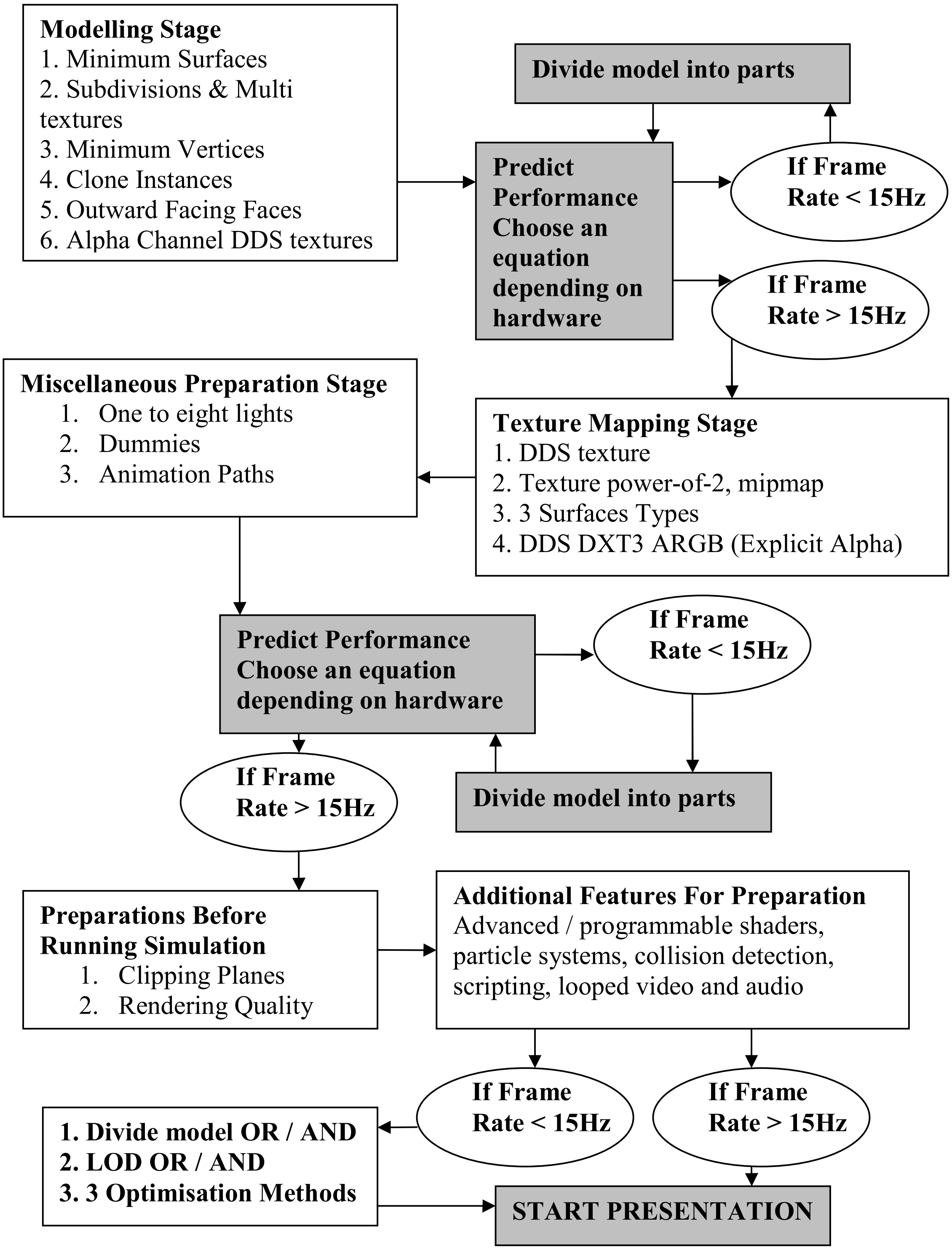


Figure 7.1. Guideline Flowchart

7.3 Final Conclusion

In conclusion, the research goes through a few steps to achieve efficient real-time architectural VR visualisation. The research began with investigating optimisation techniques through literature reviews and self observations. The techniques are crucial to determine what the best methods to adopt in the VR software are. These methods should apply to all VR software while it could be different for game engines. From the tests carried out, minimum subdivision, copy with instance method, billboard or flat surfaces, alpha channel and using DDS power-of-2 texture format are essential steps to make the basic model efficient. After that, the adjustments of view culling techniques and the near and far clipping planes need to be done in the VR software to show only what are necessary in the scene.

LOD usage is optional and normally it is only used when the scenes are extremely complex but everything must be loaded at the same time. Polygon reductions and simplifications are preferred for organic models that are too complex and highly detailed. Geometric forms substitution is used if it is fine to replace a whole façade or surface using textures to represent elements such as windows and doors. The final optimisation is at the last stage where the VR software's own compression method for geometry, triangle and textures is used, if available.

Next, the selected methods are used in the independent variable tests. All the variables are tested independently against frame rate, time taken and memory consumptions. It is found that frame rate is the best measure for performance in VR. Therefore, it is used for all the remaining tests. Four basic variables, which are triangle, vertex, geometry and texture are selected for the scope of this research to find their relationships as a whole with frame rate. It was found out that triangle count is the biggest contributor followed closely by geometry count. The other variables are added features that can be implemented in the VR software too. They are light count, particle systems, programmable / advanced shaders, scripting, collision detection, looped video and audio. All these can be added on top of the basic model to improve their realism. It was during this point that hardware and software comparison is done to check how much impact they have. It was shown that only hardware will affect the results.

In the final part of the research, the quantitative and qualitative aspects of VR are investigated. An experiment is conducted to ascertain the quantitative aspect by taking into account all four fundamental variables together, using simple multiple regression method. They are triangle, geometry, vertex and texture count. From the results, triangle count is the biggest contributor and followed closely by geometry count. The multiple regression equation derived from the experiment can be used to predict the frame rate during the process of constructing any 3D model. The verification process on one of the hardware systems has proven that the prediction is quite accurate.

A survey is conducted and it was found that people with 3D first person game experience and who are also architecture students have the highest demands for visual quality. For triangle complexity acceptance, most people can accept 65 - 75% compression for organic 3D models. Apart from that, texture resolution of 1024 X 1024 is acceptable for flat surfaces, 512 X 512 is acceptable for curved surfaces and 256 X 256 is acceptable for small rough surfaces. A frame rate of 15Hz and above is accepted by everyone.

A guideline is proposed in the final conclusion stage by taking into account all the processes and experiments done in the research, from the optimisation techniques to the independent variable tests, to the final experiment and survey conducted. The VR 3D scene designer can use the guide to make decisions from the modelling, texture mapping, lighting, and animation stages all the way to VR simulation. This guideline aims at accomplishing the research goal, which is achieving efficient real-time VR architectural visualisation.

CHAPTER 8.0: FUTURE RESEARCH

One of the initial intentions of this research was to do DOE (Design of Experiment) in the final portion of the research after simple multiple regression of the four variables. Unfortunately, due to time and resource constraints it could not be done. DOE is done to optimise processes in order to find out the most optimum performance with all the variables combination (Antony 2005).

DOE is a major tool used in the improvement phase of many Six Sigma® projects. Six Sigma® basically indicates a level of performance equating to 3.4 nonconformities per million opportunities, a level of performance considered as world-class (Henderson 2006). A good tool used to calculate DOE is the Minitab statistical software which has such functions.

Further research can be done to include more variables, i.e. scripting, advanced / programmable shaders, amount of lights, collision detection, particle systems, volume shadows and HDR images in the multiple regression. It will be a lot more work to add in so many variables in the experiment but it will give more options to VR 3D scene designers during the early stage of the 3D model construction. The use and comparison with game engines will be also very interesting.

Finally, the exploration of using VR as a design tool for the architectural design process and not merely for the final visualisation is another area for further research. VR can help architects create designs closer to their intentions since stereoscopic VR enables architects to view in three dimensions with depth and to navigate through the space that is being sculpted during the design process.

BIBLIOGRAPHY

3ality Digital LLC. (2008). The online home of 3ality Digital. Retrieved 12 April 2008 from the World Wide Web: http://www.3alitydigital.com/home.php.

Af Klercker, J. (2000). Modelling for Virtual Reality in Architecture, *Pro. of the 18th eCAADe Conference*, pp.209-213.

ACM (2006), Nov. *Real-Time Rendering Resources*. Mass: ACM. Retrieved 17 August, 2006 from the World Wide Web: http://www.realtimerendering.com.

Agi (2000). *Unraveling The Mystery of VSYNC*, USA: Digital Silence. Retrieved 2 November 2006 from the World Wide Web: http://www.d-silence.com/feature.php?id=255

Agi (2002). *What Does "Optimal" Refresh Rate really mean? (Win2k/WinXP)*, USA: Digital Silence. Retrieved 2 November 2006 from the World Wide Web: http://www.d-silence.com/feature.php?id=206.

Aliaga, D.G., and Lastra, A. (1999). Automatic image placement to provide a guaranteed frame rate. *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pp. 307-316. ACM Press / Addison-Wesley Publishing Co.

Antony, J. (2005). *Design of Experiments for Engineers and Scientists*, Elsevier Butterworth-Heinemann, MA, USA.

Beaudoin, J., and Keyser, J. (2004). Simulation levels of detail for plant motion. *Proceedings of the 2004 ACM SIGGRAPH / Eurographics symposium on Computer animation*, pp. 297-304. Grenoble, France. ACM Press.

Bonsor, K. (undated), Nov. *How Internet Odors Will Work*. Howstuffworks. Retrieved 3 February, 2007 from the World Wide Web: http://computer.howstuffworks.com/internet-odor1.htm.

Burdea, G., and Coiffet, P. (2003). *Virtual reality technology*, J.Wiley-Interscience, Hoboken, NJ.

Castle, H. (Ed.). (2005). *4Dspace: Interactive Architecture*. Architectural Design, *75*(1). London: Wiley-Academy.

Chhugani, J., Purnomo, B., Krishnan, S., Cohen, J., Venkatasbramanian, S., Johnson, D.S., and Kumar, S. (2005). VLOD: high-fidelity walkthrough of large virtual environments. *IEEE Transactions on Visualization and Computer Graphics*, *11*(1), 35-47.

Chiu-Shui, C. (2007). Does Color Have Weaker Impact On Human Cognition Than Material? Dong, A., Moere, A.V., and Gero, J.S. (Eds.) Proceedings of the International Conference of CAAD Futures 2007, pp. 373-386. The Netherlands, Springer.

Chiu-Shui, C., and Chien-Hui, W. (2005). How Real is the Sense of Presence in A Virtual Environment?: Applying Protocol Analysis for Data Collection. Anand, B. (Ed.), Proceedings of the Tenth International Conference of CAADRIA 2005, pp. 188-197. New Delhi, TVB School of Habitat Studies.

Coorg, S., and Teller, S. (1997). Real-time occlusion culling for models with large occluders. *Proceedings of the 1997 symposium on Interactive 3D graphics*, pp. 83-90. Providence, Rhode Island, United States. ACM Press.

Czerwinski, M., Tan, D.S., and Robertson G.G. (2002). Women take a wider view. Proceedings of the SIGCHI conference on Human factors in computing systems: Changing our world, changing ourselves, pp. 195-202. Minneapolis, Minnesota, USA. ACM Press.

Darken, R.P., and Sibert, J.L. (1996). Wayfinding strategies and behaviours in large virtual worlds. Conference on Human Factors in Computing Systems. *Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 142-149. Vancouver, British Columbia, Canada. ACM Press.

Digital Urban (2006), *Game Engines*, UK: UCL. Retrieved 11 November, 2006, http://digitalurban.blogspot.com/search/label/Game%20Engines

Dinh, H.Q., Walker, N., Song, C., Kobayashi, A., and Hodges, L. F. (1999). Evaluating the Importance of Multi-Sensory Input on Memory and the Sense of Presence in Virtual Environments, *IEEE Virtual Reality Proceedings*, pp. 222-228.

Duh, H.B.L., Parker, D.E., and Furness, T.A. (2001). An “independent visual background” reduced balance disturbance envoked by visual scene motion: implication for alleviating simulator sickness. *Proceedings of the SIGCHI Conference on Human factors in computing systems*, pp. 85-89. Seattle, Washington, USA: ACM Press.

Elliott, A.C., and Woodward, W.A. (2007). *Statistical Analysis Quick Reference Guidebook With SPSS Examples*, Sage Publications, Inc., United States of America.

Faustner, L. (2007). *Stereoscopic Techniques in Computer Graphics*. Student project, Technische Universitat Wien, Austria. Retrieved 10 September, 2007 from the World Wide Web: http://www.cg.tuwien.ac.at/research/publications/2006/Fauster-06-st

Fernando, R. (Ed.). (2004). *GPU gems : programming techniques, tips, and tricks for real-time graphics*, Addison-Wesley, Boston, MA.

Flavelle C. (2008). Pictures That Move in 3-D. Newsweek. Retrieved 12 April 2008 from the World Wide Web:
http://www.newsweek.com/id/112749.

Fukuda T., Kazuhiro, S., Yeo, W., and Kaga, A. (2006). Development and Evaluation of a Close-range View Representation Method of Natural Elements in a Real-time Simulation for Environmental Design - Shadow, Grass, and Water Surface, *Pro. of the 24th eCAADe Conference*, pp.58-65.

Garland, M., and Paul S. Heckbert, P.S. (1998). Simplifying surfaces with color and texture using quadric error metrics. *IEEE Visualization 98 Conference Proceedings*, pp. 263-269.

Garland, M., and Heckbert, P.S. (1997). Surface simplication using quadric error metrics. *SIGGRAPH 97 Proc.*, pp. 209-216.

Gray, R. (2007), Sept. *Internet Handshake extends the Human Touch*. London, Telegraph Group Ltd. Retrieved 19 September 2007 from the World Wide Web:
http://www.telegraph.co.uk/earth/main.jhtml?xml=/earth/2007/09/16/scitouch116.xml.

Greenemeier L. (2008). Holographic Film for 3-D, Sans Those Silly Specs. Scientific American. Retrieved 12 April 2008 from the World Wide Web:
http://www.sciam.com/article.cfm?id=hologram-photoreflective-polymer-film.

Hua W., Zhang, H., Lu, Y., Bao, H., and Peng, Q. (2004). Huge texture mapping for real-time visualization of large-scale terrain. *Proceedings of the ACM Symposium on Virtual reality software and technology*, pp. 154-157. Hong Kong. ACM Press.

Helman, J.L. (1999). Architecture and performance of entertainment systems, appendix a. ACM SIGGRAPH 95 Course Notes #6, pp. 1.92-1.32.

Henderson, G.R. (2006). *Six Sigma Quality Improvement with MINITAB*, John Wiley & Sons Ltd, Wes Sussex, England.

Kalay, Y.E. (2004). *Architecture's New Media*, The MIT Press, Massachusetts, USA.

Kennedy, R.S., Lane, N.E., Berbaum, K.S., and Lilienthal, M.G. (1993). A simulator sickness questionnaire (SSQ): A new method for quantifying simulator sickness. *International Journal of Aviation Psychology*, *3*(3), pp. 203-220.

Knight, M.M. and Arns, L.L. (2006). The relationship among age and other factors on incidence of cybersickness in immersive environment users. ACM International Conference Proceeding Series, *153. Proceedings of the 3rd symposium on Applied perception in graphics and visualization*, p. 162. Boston, Massachusetts, USA. ACM Press.

Koike, T., Utsugi, K., and Oikawa, M. (2005). VR content platform for multi-projection displays with realtime image adjustment. *Proceedings of the 2005 International conference on Augmented tele-existence, pp. 125-132. Christchurch, New Zealand. ACM Press.*

LaViola, J.J. (2000). A discussion of cybersickness in virtual environments. *ACM SIGCHI Bulletin, 32*(1), 47-56. ACM Press.

Lin, J.J.W., Abi-Rached, H., and Lahav, M. (2004). Virtual guiding avatar: an effective procedure to reduce simulator sickness in virtual environments. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 719-726. Vienna, Austria. ACM Press.

Lo, C.H., and Chalmers, A. (2003). Stereo vision for computer graphics: the effect that stereo vision has on human judgments of visual realism. *Proceedings of the 19th spring conference on Computer graphics*, pp. 109-117. Budmerice, Slovakia. ACM Press.

Mania, K., and Chalmers, A. (2001). The effects of Levels of Immersion on Presence and memory in Virtual Environments: A Reality Centred Approach. *Cyberpsychology & Behaviour*, *4*(2), 247-264.

Mania, K., and Robinson, A. (2004). The effect of quality of rendering on user lighting impressions and presence in virtual environments. *Proceedings of the 2004 ACM SIGGRAPH International conference on Virtual Reality continuum and its applications in industry*, pp. 200-205. Singapore. ACM Press.

Mark, W., and Fussell, D. (2005). Real-time rendering systems in 2010. *International Conference on Computer Graphics and Interactive Techniques, Article No.19*, pp.1-8. Los Angeles, California. ACM Press.

Metaverse Roadmap (2008). Pathways to the 3D Web. Retrieved 12 April 2008 from the World Wide Web:
http://www.metaverseroadmap.org.

Möller, T. (2002). *Real-time rendering*, AK Peters, Natick, Mass.

Morie, J.F., Iyer, K., Valanejad, K., Sadek, R., Miraglia, D., Milam, D., Williams, J., Luigi, D.P., and Leshin, J. (2003). Sensory design for virtual environments. *International Conference on Computer Graphics and Interactive Techniques*, p. 1. San Diego, California. ACM Press.

Ni, T., Bowman, D.A., and Chen, J. (2006). Increased display size and resolution improve task performance in Information-Rich Virtual Environments. ACM International Conference Proceeding Series, *137. Proceedings of the 2006 conference on Graphics interface*, pp. 139-146. Quebec, Canada. Canadian Information Processing Society Toronto, Ontario, Canada.

Novak, J. (2005). *Game Development Essentials*, Thomson Delmar Learning, New York.

Olano, M., Kuehne, B., and Simmons, M. (2003). Automatic shader level of detail. Proceedings of the ACM SIGGRAPH / EUROGRAPHICS conference on Graphics, pp. 7-14. San Diego, California. Eurographics Association.

Omernick, M. (2004). *Creating the Art of the Game*, New Riders Publishing, USA.

Platt, C. (1999), Nov. *You've Got Smell!* WIRED. Retrieved 2 February, 2007 from the World Wide Web:

http://www.wired.com/wired/archive/7.11/digiscent.html.

Ruiz-Rodarte, R., Cuspinera, F.G., and Larios, J. (2005): Real time visualizing tools: Description of an experience where game engines and internet browsers are used as means of exhibition at an archaeological site. *SIGraDi: Proceedings of the 9th Iberoamerican Congress of Digital Graphics,* pp. 689-694. Lima, Peru.

Ryu, J., Hashimoto, N., Sato, M., Soeda, M., and Ohno, R. (2006). A game engine based architectural simulator on multi-projector displays. Proceedings of the 11th International Conference on Computer Aided Architectural Design Research in Asia, pp. 613-615. Kumamoto, Japan.

Seay, A.F., Krum, D.M., Hodges, L., and Ribarsky, W. (2002). Simulator sickness and presence in a high field-of-view virtual environment. *Conference on Human Factors* *in Computing Systems*, pp. 784-785. Minneapolis, Minnesota, USA. ACM Press.

Sebastien, SL: 2004, *Shaders for game programmers and artists*, Thomson/Course Technology, Boston, Mass.

Serious Games Institute (2008). The SGI - Serious about Games. Retrieved 12 April 2008 from the World Wide Web: http://www.seriousgamesinstitute.co.uk.

Simmons, M., and Shreiner, D. (2003). Per-pixel smooth shader level of detail. International Conference on Computer Graphics and Interactive Techniques, p. 1. San Diego, California. ACM Press.

Simpson, J. (2002). *Game Engine Anatomy 101, Part I*, Ziff Davis Media Inc., NY, USA: ExtremeTech. Retrieved 2 November 2006 from the World Wide Web: http://www.extremetech.com/article2/0,1697,594,00.asp.

Siu, A.M.K., Wan, A.S.K., and Lau, R.W.H. (2004). Modeling and rendering walkthrough environments with panoramic images. *Proceedings of the ACM symposium on Virtual reality software and technology*, pp. 114-121. Hong Kong. ACM Press.

Slater, M., Steed, A., and Chrysanthou, Y. (2006). Computer graphics and virtual environments: from realism to real-time, Addison Wesley, Harlow, England.

So-Yeon, Y., and Tutar, M. (2004). A Comparison Study Between A Computer Game & A Non-Game Web3D Environment, S. Gero, J., Chase, S., Rosenman, M. (Eds.), *Proceedings of the Ninth Conference on CAADRIA 2004*, pp.601-612.

Steed, A. (1997). Efficient navigation around complex virtual environments, *Pro. of the ACM symposium on Virtual reality software and technology*, pp.173-180.

Tory, M., and Möller, T. (2004). Human factors in visualization research. *IEEE Transactions on Visualization and Computer Graphics*, *10*(1), 72-84.

Tsang, M., Fitzmaurice, G., Kurtenbach, G., and Khan, A. (2003). Game-like navigation and responsiveness in non-game applications. *Communications of the ACM, A game experience in every application*, 46(7), pp.56-61.

Tyndiuk, F., Lespinet-Najib, V., Thomas, G., and Schlick, C. (2004). Impact of large displays on virtual reality task performance. *Proceedings of the 3rd international conference on Computer graphics, virtual reality, visualization and interaction in Africa*, pp. 61-65. Stellenbosch, South Africa. ACM Press.

U2 (2008). The First Live-Action 3D Concert Movie, Featuring U2. National Geographic. Retrieved 12 April 2008 from the World Wide Web: http://www.u23dmovie.com.

Valient, M. (2004). 3D Engines in games - Introduction. Retrieved 10 January 2007 from the World Wide Web: http://pisa.ucsd.edu/cse125/2006/Papers/Introduction_to_3d_Game_Engines.pdf

Whyte, J. (2002). *Virtual Reality and the Built Environment*. Oxford: Architectural Press.

Willmott, J., Wright, L.I., Arnold, D.B., and Day A.M. (2001). Rendering of large and complex urban environments for real time heritage reconstructions. *Pro. of the 2001 conference on Virtual reality, archaeology, and cultural heritage*, pp.111-120.

Wimmer, M. and Wonka, P. (2003). Rendering time estimation for real-time rendering, *Proc. of the 14th Eurographics workshop on Rendering*, pp.118-129.

Xia, J.C., El-Sana, J. and Varshney, A. (1997). Adaptive real-time level-of-detail based rendering for polygonal models. *IEEE Transactions on Visualization and Computer Graphics*, *3*(2), pp. 171-183.

Yehuda, E.K. (2004). *Architecture's new media: principles, theories, and methods of computer-aided design*, MIT Press, Cambridge, Mass.

Yoshida, S., Yamada, K., Mochizuki, K., Aizawa, K., and Saito, T. (2002). Scope-based interaction: a technique for interaction in an image-based virtual environment. *Proceedings of the workshop on Virtual environments 2002*, pp. 139-148. Barcelona, Spain. Eurographics Association.

Yuan, P., Green, M. and Lau, R.W.H. (1997). A framework for performance evaluation of real-time rendering algorithms in virtual reality, *Pro. of the ACM symposium on Virtual reality software and technology*, pp.51-58.

APPENDICES

I Simple Multiple Regression Results Table Analysis

Descriptive Statistics

	Mean	Std. Deviation	N
logfrlaptop	1.3820	.31861	137
logtricount	5.0638	.67531	137
logvercount	4.8884	.68684	137
loggeocount	1.8918	.79287	137
logtextcount	.9771	.17715	137

The table above displays descriptive statistics for each variable. The mean is the average value. The standard deviation measures the variability (or spread) of the values. N is the number of cases with non-missing values.

Model Summary

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	.859(a)	.738	.730	.16547

a Predictors: (Constant), logtextcount, logvercount, loggeocount, logtricount

This table displays R, R squared, adjusted R squared, and the standard error. R, the multiple correlation coefficient, is the correlation between the observed and predicted values of the dependent variable. The values of R for models produced by the regression procedure range from 0 to 1. Larger values of R indicate stronger relationships. R squared is the proportion of variation in the dependent variable explained by the regression model. The values of R squared range from 0 to 1. Small values indicate that the model does not fit the data well. The sample R squared tends to optimistically estimate how well the models fits the population. Adjusted R squared attempts to correct R squared to more closely reflect the goodness of fit of the model in the population. Use R Squared to help you determine which model is best. Choose a model with a high value of R squared that does not contain too many variables. Models with too many variables are often over fit and hard to interpret.

Correlations

		logfrlaptop	logtricount	logvercount	loggeocount	logtextcount
Pearson Correlation	logfrlaptop	1.000	-.615	-.614	-.676	-.098
	logtricount	-.615	1.000	.982	.212	.123
	logvercount	-.614	.982	1.000	.214	.122
	loggeocount	-.676	.212	.214	1.000	.413
	logtextcount	-.098	.123	.122	.413	1.000
Sig. (1-tailed)	logfrlaptop	.	.000	.000	.000	.128
	logtricount	.000	.	.000	.006	.077
	logvercount	.000	.000	.	.006	.078
	loggeocount	.000	.006	.006	.	.000
	logtextcount	.128	.077	.078	.000	.
N	logfrlaptop	137	137	137	137	137
	logtricount	137	137	137	137	137
	logvercount	137	137	137	137	137
	loggeocount	137	137	137	137	137
	logtextcount	137	137	137	137	137

The correlations table above displays Pearson correlation coefficients, significance values, and the number of cases with non-missing values. Pearson correlation coefficients assume the data are normally distributed. The Pearson Correlation coefficient is a measure of linear association between two variables. The values of the correlation coefficient range from -1 to 1. The sign of the correlation coefficient indicates the direction of the relationship (positive or negative). The absolute value of the correlation coefficient indicates the strength, with larger absolute values indicating stronger relationships. The correlation coefficients on the main diagonal are always 1.0, because each variable has a perfect positive linear relationship with itself. Correlations above the main diagonal are a mirror image of those below. The significance of each correlation coefficient is also displayed in the correlation table. The significance level (or p-value) is the probability of obtaining results as extreme as the one observed. If the significance level (or p-value) is very small (less than 0.05) then the correlation is significant and the two variables are linearly related. If the significance level is relatively large (for example, 0.50) then the correlation is not significant and the two variables are not linearly related. N is the number of cases with non-missing values.

ANOVA(b)

Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	10.191	4	2.548	93.050	.000(a)
	Residual	3.614	132	.027		
	Total	13.805	136			

a Predictors: (Constant), logtextcount, logvercount, loggeocount, logtricount

b Dependent Variable: logfrlaptop

The table above summarizes the results of an analysis of variance. The sum of squares, degrees of freedom, and mean square are displayed for two sources of variation, regression and residual. The Regression shows the information about the variation accounted for by the model while Residual shows the information about the variation not accounted for by the model. The output for Total is the sum of the information for Regression and Residual. A model with a large regression sum of squares in comparison to the residual sum of squares indicates that the model accounts for most of variation in the dependent variable. Very high residual sum of squares indicate that the model fails to explain a lot of the variation in the dependent variable, and you may want to look for additional factors that help account for a higher proportion of the variation in the dependent variable. The mean square is the sum of squares divided by the degrees of freedom. The F statistic is the regression mean square (MSR) divided by the residual mean square (MSE). The regression degrees of freedom is the numerator df and the residual degrees of freedom is the denominator df for the F statistic. The total number of degrees of freedom is the number of cases minus 1. If the significance value of the F statistic is small (smaller than say 0.05) then the independent variables do a good job explaining the variation in the dependent variable.

Coefficients(a)

	Unstandardized Coefficients		Standardized Coefficients	t	Sig.	Collinearity Statistics	
	B	Std. Error	Beta	Tolerance	VIF	B	Std. Error
(Constant)	2.656	.128		20.702	.000		
logtricount	-.155	.112	-.330	-1.385	.168	.035	28.550
logvercount	-.082	.110	-.177	-.742	.459	.035	28.564
loggeocount	-.268	.020	-.667	-13.411	.000	.802	1.246
logtextcount	.431	.088	.240	4.896	.000	.828	1.207

a Dependent Variable: logfrlaptop

The table above is a coefficients table. The unstandardized coefficients are the coefficients of the estimated regression model. Often the independent variables are measures in different units. The standardized coefficients or betas are an attempt to make the regression coefficients more comparable. If you transformed the data to z scores prior to your regression analysis, you would get the beta coefficients as your unstandardized coefficients. The t statistics can help you determine the relative importance of each variable in the model. As a guide regarding useful predictors, look for t values well below -2 or above +2.

II EXPERIMENT SAMPLES

LOGFRLap01 = Log of Frame Rate of Laptop01 (DELL Inspiron 9300)

LOGFRLap02 = Log of Frame Rate of Laptop02 (DELL Inspiron 1520)

LOGFRRC = Log Frame Rate of Research Comp (DELL Precision 380)

LOGFRPP = Log of Frame Rate of Prepocessing Comp (DELL Precision 670)

LOGFRS = Log of Frame Rate of Stereo of DELL Precision 650

LOGFRM = Log of Frame Rate Mono of DELL Precision 650

LOGTri = Log of Triangle Count

LOGVer = Log of Vertex Count

LOGGeo = Log of Geometry Count

LOGTex = Log of Texture Count

TABLE A1. Samples Part 1

Projects	LOGFRLap01	LOGFRLap02	LOGFRRC	LOGFRPP	LOGFRS	LOGFRM	LOGTri	LOGVer	LOGGeo	LOGTex
Research	0.95424	1.62941	0.88649	0.91908	0.63347	0.95424	5.98913	5.71162	2.09342	0.90309
Research	0.99123	1.47129	0.77085	0.74036	0.62325	0.90309	6.13204	5.91716	2.77887	0.95424
Unknown	1.47857	1.50651	1.54531	1.50651	1.50379	1.77379	4.99358	5.13261	1.97313	0.90309
Unknown	1.32838	1.74036	1.50651	1.69897	1.39967	1.68842	4.69667	4.4715	2.55751	1.04139
Year2	1.2833	1.74036	1.05308	1.50379	0.87506	1.17609	5.85144	5.60742	0.90309	0.8451
Year2	1.43775	1.63246	1.74036	1.61066	1.50651	1.85914	4.90538	4.70567	2.04532	1.17609
Year2	1.43775	1.80482	1.26245	1.62118	1.2878	1.60531	5.35205	5.33316	0.8451	0.77815
Year2	1.46389	1.7364	1.24304	1.46835	1.12385	1.38382	5.60788	5.36467	0.77815	0.77815
Year2	1.29447	1.68124	1.06819	1.26007	0.90309	1.21748	5.83749	5.5966	0.77815	0.77815
Year2	1.40824	1.50651	1.62941	1.68842	1.5611	1.80072	4.45433	4.26698	1.86332	0.90309
Year2	1.20412	1.43775	1.50379	1.33041	1.27646	1.52763	4.3142	4.43944	2.23045	1.14613
Year2	1.32838	1.68124	1.13672	1.37291	1.02531	1.29885	5.70748	5.61251	0.69897	0.69897
Year2	1.68124	1.74036	1.80482	1.80956	1.92686	1.80956	4.48419	4.36359	2.233	1.07918
Year2	1.47129	1.7364	1.40824	1.7364	1.38021	1.59879	5.22705	5.0487	1.17609	1.17609
Year2	1.47129	1.74036	1.43775	1.74036	1.3784	1.60206	5.22678	5.04751	1.17609	1
Year2	1.47129	1.74036	1.27416	1.45025	1.12057	1.38382	5.62829	5.50619	0.95424	0.95424
Year2	1.2833	1.58546	1.02938	1.32634	0.96379	1.24055	5.74842	5.58355	1	1
Year2	1.63246	1.74036	1.62941	1.88593	1.77815	2.07918	4.3486	4.02494	0.60206	0.60206
Year2	1.7364	1.80277	1.80956	2.10619	1.77815	2.07041	4.20844	3.89476	0.60206	0.60206
Year2	1.50379	1.68124	1.80482	1.91487	1.85914	2.05385	4.13021	4.07896	2.14301	1.04139
Year2	1.2833	1.54531	1.06446	1.40824	1.04922	1.36173	5.5866	5.45148	0.60206	0.60206
Year2	1.50379	1.74036	1.47129	1.80956	1.57978	1.77379	4.96166	4.71246	1.07918	1.07918
Year2	1.2833	1.68124	1.10721	1.54283	1	1.29885	5.71277	5.44111	1.14613	1.11394
Year2	1.20412	1.47129	1.22272	1.09342	1.02938	1.29885	5.62789	5.86395	2.58433	0.95424
Year2	1.1959	1.58546	1.01703	1.39794	0.86923	1.20412	5.84548	5.7476	0.69897	0.69897
Year2	1.46835	1.7364	1.54283	1.7364	1.60206	1.77815	4.59294	4.29848	0.69897	0.60206
Year2	1.10721	1.33041	1.10721	0.80618	0.86923	1.15534	6.1094	5.82449	2.74974	1.30103
Year2	1.30535	1.62941	1.50379	1.38739	1.33041	1.58771	5.10229	4.86303	2.79309	1.23045
Year2	1.43775	1.7364	1.2833	1.51055	1.17609	1.47712	5.54359	5.26557	0.77815	0.77815
Year2	1.50379	1.7364	1.80956	1.80482	1.77815	2.07918	4.60885	4.43196	1.98227	1.04139
Year2	1.43775	1.74036	1.22272	1.47129	1.1271	1.38021	5.57958	5.37589	0.90309	0.90309
Year2	1.20412	1.63246	1.00432	1.43775	0.87506	1.17609	5.85969	5.61157	0.8451	0.8451
Year2	1.38021	1.62941	1.26245	1.54033	1.21484	1.47712	5.42075	5.15367	0.8451	0.8451
Year2	1.46835	1.7364	1.32838	1.64738	1.2878	1.60206	5.40195	5.13633	0.90309	0.77815
Year2	1.32838	1.7364	1.06446	1.25527	0.85126	1.23553	5.79581	5.65068	1.17609	0.90309

TABLE A2. Samples Part 2

Projects	LOGFRLap01	LOGFRLap02	LOGFRRC	LOGFRPP	LOGFRS	LOGFRM	LOGTri	LOGVer	LOGGeo	LOGTex
Year2	1.33041	1.74036	1.24055	1.60746	1.17609	1.47712	5.42914	5.40035	0.8451	0.8451
Year2	0.98227	1.22272	1.18752	0.89763	0.8451	1.11394	5.49178	5.16451	3.60959	0.77815
Year2	1.20412	1.54283	1.50379	1.35411	1.20412	1.48996	4.40047	4.33939	2.76193	1.04139
Year2	1.47129	1.68124	1.74036	1.79588	1.74036	1.95231	4.62365	4.64363	1.54407	0.8451
Year2	1.47129	1.50651	1.76938	1.70243	1.76118	1.97174	4.96227	4.65393	2.18184	1.11394
Year2	1.50651	1.58546	1.993	1.77525	1.72099	1.95856	4.38487	4.21804	2.19312	0.90309
Year2	0.90309	1.07918	1.02938	0.60206	0.6902	0.90309	5.97268	5.72525	3.67006	1
Year2	0.81291	1.05308	0.94939	0.57978	0.61278	0.88649	5.78875	5.57449	3.74547	0.90309
Year2	1.43775	1.74036	1.30535	1.58771	1.233	1.48001	5.46734	5.22601	1.07918	1
Year3	1.68124	1.7364	1.80956	1.95856	1.94547	2.07041	4.44185	4.16044	2.38917	1.23045
Year3	1.62941	1.68124	1.46835	1.69897	1.38021	1.60531	5.32625	5.07561	1.27875	1.04139
Year3	1.05308	1.43933	1.05308	0.87506	0.85126	1.13354	5.80695	5.56742	3.12581	1.11394
Year3	1.50379	1.58546	1.80482	1.87506	1.86451	2.09691	4.34745	4.17441	1.74036	1
Year3	1.47129	1.74036	1.33041	1.51851	1.29885	1.60206	5.43678	5.18265	1.88081	1.20412
Year3	1.32838	1.47129	1.47129	1.52244	1.45939	1.69197	4.4778	4.40195	2.27646	1.04139
Year3	1.15534	1.32634	1.1959	1.05308	0.91908	1.22789	5.72679	5.49957	2.66839	0.90309
Year3	1.30535	1.62941	1.26007	1.35218	1.26007	1.53782	5.35965	5.12317	2.86629	1.14613
Year3	1.33041	1.47276	1.68124	1.76118	1.58771	1.83378	3.94685	3.81862	2.11727	0.8451
Year3	1.30535	1.74036	1.26007	1.48144	1.20412	1.47276	5.35929	5.06681	1.11394	1.04139
Year3	1.62941	1.74036	1.7364	1.97864	2.08778	2.07918	3.78017	3.61363	1.91381	0.77815
Year3	1.68124	1.74036	2.01578	2.21005	2.00732	2.07041	4.48451	4.6974	1.32222	0.69897
Year3	1.30535	1.58546	1.63246	1.43297	1.32015	1.58546	4.66595	4.45301	2.94002	1
Year3	1.43775	1.68124	1.60746	1.65128	1.59879	1.78247	5.04397	4.82908	2.23553	1.04139
Year3	1.7364	1.80482	2.10619	2.273	2.07041	2.08778	3.36605	3.11059	1.63347	1.11394
Year3	1.62941	1.68124	1.80482	1.90902	1.76567	2.07918	4.84925	4.75527	2.05308	0.77815
Year3	1.68124	1.68485	1.80482	2.02243	1.77379	2.07918	4.47975	4.30496	1.75587	0.95424
Year3	1.05308	1.40824	0.90849	1.00432	0.81954	1.11727	5.88785	5.64649	3.01578	0.90309
Year4	1.47129	1.62941	1.80482	1.8537	1.80956	2.02243	4.05077	3.8408	2.24797	1.07918
Year4	1.47129	1.58546	1.80482	1.79449	1.72916	1.92686	4.64658	4.38194	1.87506	0.77815
Year4	1.12057	1.47129	1.33041	1.16435	1.07188	1.34044	4.87869	4.6252	3.01072	1.17609
Year4	1.68124	1.74036	1.80482	2.18667	2.04571	2.10619	4.11902	3.98967	1.54407	0.69897
Year4	1.01703	1.43775	0.79934	0.80618	0.62325	0.91381	6.1374	6.08829	3.07482	1.14613
Year4	1.40824	1.68124	1.58546	1.63849	1.57403	1.82413	4.6452	4.36758	2.84198	1.04139
Year4	1.58206	1.68485	1.80482	1.91487	1.8488	2.10619	4.56541	4.40398	2.16732	1.07918
Year4	1.08991	1.38202	1.26951	0.98227	0.94939	1.15229	5.08968	4.83359	2.64836	1.04139

TABLE A3. Samples Part 3

Projects	LOGFRLap01	LOGFRLap02	LOGFRRC	LOGFRPP	LOGFRS	LOGFRM	LOGTri	LOGVer	LOGGeo	LOGTex
Year4	1.20412	1.50651	1.35411	1.27184	1.23805	1.50379	5.38191	5.49464	2.08991	1.14613
Year4	1.20412	1.43933	1.50379	1.3483	1.10721	1.33041	4.3889	4.28713	2.61805	1.23045
Year4	1.68124	1.80482	1.88593	2.02979	1.78247	2.07918	4.77143	4.54289	1.97772	1
Year4	1.68124	1.74036	1.62941	1.83378	1.77815	2.08778	4.6805	4.30961	1.14613	0.95424
Year4	1.63246	1.74036	1.80482	2.02243	2.02979	2.07918	4.41852	4.26472	1.74819	0.90309
Year4	1.33041	1.68124	1.50379	1.39794	1.42813	1.67394	5.22437	5.04937	2.51455	1.07918
Year4 DTM	1.10721	1.40824	0.91381	1.00432	0.76343	1.0607	5.99611	5.9008	2.16137	1
Year4 DTM	1.20412	1.68124	1.09342	1.18184	1.02531	1.30103	5.73303	5.47758	2.03743	0.77815
Year4 DTM	1.74036	1.80482	2.10619	2.38021	2.07918	2.07918	3.48401	3.23426	1.60206	0.8451
Year4	0.91908	1.47276	0.87506	0.98677	0.64345	0.93952	6.05422	5.75457	2.42325	1.25527
Year4	1.74036	1.81023	1.97864	2.12483	2.08778	2.07918	4.52795	4.26463	1.50515	1.17609
Year4	1.74036	1.80956	2.10619	2.34674	2.07041	2.07918	3.9843	3.73743	1.14613	1.04139
Year4	1.43775	1.47129	1.74036	1.76716	1.65801	1.90309	4.55772	4.55281	1.69897	0.95424
Year4 IS	1.74036	1.80482	2.10619	2.39794	2.07918	2.07918	3.88564	3.82491	1.36173	1
Year4 IS	1.7364	1.80956	1.88593	2.41647	2.08778	2.10619	3.5293	3.35044	0.90309	0.8451
Year4 IS	1.74036	1.80956	1.80482	2.10619	2.01452	2.07918	4.58737	4.20401	1.86923	1
Year4 IS	1.7364	1.80482	1.97864	2.273	2.07918	2.07918	4.36434	4.55844	1.23045	1.04139
Year4 IS	1.74036	1.80956	1.88593	2.11494	1.77379	2.07918	4.72999	4.52924	1.36173	1.14613
Year4	1.43775	1.7364	1.30535	1.47567	1.12057	1.38917	5.62174	5.38653	1.23045	1.04139
Year4	0.91381	1.13672	0.99123	0.75587	0.70757	0.95904	5.79468	5.94098	3.40603	1.07918
Year4	1.74036	1.80482	2.11528	2.19838	2.08778	2.07918	4.51175	4.2714	1.38021	1.07918
Year4	1.74036	1.80956	1.98588	2.05385	1.77379	2.07918	4.81839	4.68152	1.5682	1.11394
Year4	1.74036	1.80482	1.88593	2.41647	2.07918	2.07918	4.25259	3.97603	1	0.60206
Year4	1.7364	1.80482	1.7364	1.80956	1.58771	1.77815	5.26524	5.00999	1.11394	0.60206
Year4	1.32838	1.68124	1.2833	1.35411	1.17319	1.47567	5.78971	5.52926	1.32222	0.77815
Year4	1.62941	1.80482	1.58206	1.71517	1.48001	1.78675	5.34107	5.01107	1.62325	1.14613
Year5	1.74036	1.80956	1.80482	1.95231	1.77379	2.07918	5.04731	4.77541	1.32222	1.25527
Year5	1.04139	1.43933	1.32838	1.16732	1.04139	1.3032	4.50163	4.2697	3.02036	1.17609
Year5	1.68124	1.80482	1.68124	1.8109	1.59879	1.80072	5.23242	5.00773	1.17609	1.17609
Year5	1.74036	1.80482	2.10619	2.38021	2.07041	2.07918	4.11193	3.81948	0.69897	0.69897
Year5	1.7364	1.80482	1.97864	2.28668	2.07041	2.07918	4.24117	3.97914	1.5682	1.27875
Year5	1.74036	1.80482	1.88593	2.38021	2.07918	2.07918	4.21762	3.99038	0.8451	0.8451
Year5	1.68124	1.80482	1.62941	1.7767	1.57978	1.78247	5.23139	5.39908	1.5563	1.27875
Year5	1.7364	1.80956	1.81889	1.70243	1.60531	1.95231	5.15818	4.92643	1.25527	1.17609
Year5	1.62941	1.74036	1.55023	1.61595	1.38021	1.74036	5.46486	5.18795	1.54407	1.25527

GLOSSARY

3D	Three dimensional, comes in 3 axis of X, Y and Z
3DS	A very common 3D model file format.
ACM	Association for Computing Machinery
Advanced / Programmable Shaders	A set of software instructions used by the graphic resources to perform rendering effects and allowing the 3D application designer to program the GPU. Two major graphics software Libraries are OpenGL and DirectX and they use three types of Shaders: vertex shaders, geometry shaders and pixel shaders.
AGP	Accelerated Graphics Port – a dedicated graphics port that allows system memory to be used for video-related tasks. AGP delivers a smooth, true-colour video image because of the faster interface between the video circuitry and the computer memory.
Aliasing	A distortion of artifact that results when a signal is sampled and reconstructed as an alias of the original signal.
Alpha Channel	In a image element, it is a mask where a portion of each pixel's data that is reserved for transparency information.
ANOVA	Analysis of Variance
Antialias	A technique of minimizing the distortion artifacts known as aliasing when representing a high-resolution signal at a lower resolution.
API	Application Programming Interface

Architecture	Is the art and science of designing and constructing buildings and other structures in the urban context and the built environment.
Augmented Reality (AR)	A combination of real world physical objects with computer-generated data. Among the common research interests are motion-tracking data, fiducial marker recognition through machine vision and construction of controlled environments containing sensors and actuators.
Backface Culling	The back side of some faces will never face the camera so there is .no reason to draw them and therefore will be culled.
Billboard	A flat surface used to replace a potentially complex 3D object with its 2D representative (a sprite) rendered from some point of view and showing this sprite upright to the camera no matter how camera is rotated or positioned with 2 or more planes crisscrossed.
Blinn Shading	A mathematical formula to compute surface characteristics. Blinn shading model simulates ambient reflections, specular reflections to the angle where you view the surface at as well as diffuse reflections.
CAADRIA	Computer Aided Architectural Design Research in Asia
CAD	Computer Aided Design
CAVE	Cave Automatic Virtual Environment, a virtual reality system which the audience visualise inside it. It is a box with 4, 5 or 6 walls systems.

Cg	C for Graphics
Clipping Planes	In VR visualisations, they are used to help the renderer define the boundaries between them which can be seen by the camera and not to calculate anything beyond them to save processing power and memory.
Coefficient of Determination	In statistics, it is also known as R^2 (Square of a correlation coefficient), is the proportion of variability in a data set that is accounted for by a statistical model. An R^2 of 1.0 indicates that the regression line perfectly fits the data.
Collision Detection	In VR visualisations, it is used as a physics simulation particularly to give the feeling of the force of gravity and to collide into surfaces that are colliding with the camera. Without it, the user will go through walls, floors and any 3D objects in the scene.
CPU	Computer Processing Unit
CRT	Cathode Ray Tube
Cyber Sickness	In the research, also known as simulation sickness, which is a condition where a person exhibits symptoms similar to motion sickness caused by watching the VR simulation. It is normally caused by fast navigation with vibrations in different angles.
DDS	DirectDraw Surface, a DirectX format texture
Desktop	A personal computer (PC) with the most common ones with vertical tower cases with LCD monitor.
DPI	Dot per inch

DWG	Autodesk® AutoCAD® drawing file format
DDR SDRAM	Double-data-rate SDRAM that doubles the data burst cycle, improving system performance.
DDR2 SDRAM	Double-data-rate 2 SDRAM – A type of DDR SDRAM that uses a 4-bit prefetch and other architectural changes to boost memory speed to over 400MHz.
Dual-core	An Intel® technology in which two physical computational units exist inside a single processor package, thereby increasing computing efficiency and multitasking ability.
DTM	Design Technology and Management Track
eCAADe	Education and research in computer aided architectural design in Europe
ed.	Editor or edition
EOZ	EONReality™ EON™ Studio™ file format
et. al.	And others, from Latin et alii.
Experiment	A scientific method which consists of a set of observations performed in the context of solving a particular problem or question, to retain or falsify a hypothesis or research concerning phenomena. It is an empirical approach to acquire deeper knowledge about the physical world.
Flat Shading	A lighting technique that shades each polygon on the angle between the polygon's surface normal and the direction of the light

	source, their respective colours and the intensity of the light source.
Frame Rate	Frame frequency, is the measurement of the frequency (rate) at which an imaging device produces unique consecutive images called frames. It is expressed in frames per second (FPS) and in monitors as Hertz (Hz).
Game Engine	The core software component of a computer video game or other interactive application with real-time graphics.
Geometry	Group of surfaces, triangles or polygons in computer graphics
GHz	Gigahertz – A measurement of frequency that equals one thousand million Hz, or one thousand MHz. The speeds for computer processors, buses, and interfaces are often measure in GHz.
GLSL	OpenGL$^{\text{TM}}$ Shader Language.
Gouraud Shading	A method of shading to simulate the differing effects of light and colour across the surface of an object. It can achieve smooth lighting on low-polygon surfaces without the heavy computational requirements of calculating light for each pixel.
GPU	Graphics Processing Unit
Haptic	Sense of touch or contact
HD	High Definition
HDRI	High Dynamic Range Image
HLSL	High Level Shader Language
HMD	Head Mounted Devices

Hyper-Threading	Hyper-Threading is an Intel technology that can enhance overall computer performance by allowing one physical processor to function as two logical processors, capable of performing certain tasks simultaneously.
Hz	Hertz – a unit of frequency measurement that equals 1 cycle per second.
Hologram	A form of photography which allows an image to be recorded in three dimensions.
Hypothesis	A suggested explanation for a phenomenon or a reasoned proposal suggesting a possible correlation between multiple phenomena. It is normally based on previous observations or on extensions of scientific theories.
ID	Industrial Design
i.e.	That is, from Latin id est.
JPEG	Joint Photographic Experts Group**,** the standard algorithm for the compression of digital images.
KB	KiloByte, (1,024 (2^{10}) bytes – a unit of data that equals 1024 bytes but is often referred to as 1000 bytes.
Lambertian Shading	This shading model simulates diffuse reflection but not specular reflection.
Laptop	Also known as notebook, a small mobile computer which has the normal weight of around 1-8 kilograms and can run on a battery or from a normal external power adapter..

LCD	Liquid crystal display – The technology used by portable computer and flat-panel displays.
LOD	Level of Detail, which involves decreasing the complexity of a 3D object representation as the camera moves away further from it. The replacement model with reduced visual quality is normally unnoticed because of the small effect on object appearance when the camera is too far away from it.
M&E	Mechanical and Electrical
MB	Megabyte – A measurement of data storage that equals 1,048,576 bytes. 1 MB equals 1024 KB. When used to refer to hard drive storage, the term is often rounded to 1,000,000 bytes.
Memory	A temporary data storage area inside your computer. Because the data in memory is not permanent, it is recommended that you frequently save your files while you are working on them, and always save your files before you shut down the computer. Your computer can contain several different forms of memory, such as RAM, ROM, and video memory. Frequently, the word memory is used as a synonym for RAM.
MIP	"multum in parvo", Latin for "many in a small place"
Mipmapping	Process of choosing a texture from pool of (identical) textures with various resolutions according to the distance of textured object. The smallest texture is used on the farthest object and high resolution texture is used on near object.

MIT	Massachusetts Institute of Technology
Mixed Reality (MR)	Also referred to as augmented reality, it refers to the merging of both the real and virtual worlds to produce visualisations where both physical and digital objects co-exist and interact in real-time.
Motion Sickness	Also called kinetosis, it is a condition in which a disagreement exists between visually perceived movement and the vestibular system’s sense of movement. Common symptoms of motion sickness are dizziness, fatigue and nausea.
Multiple Regression	A method to find the relationship between several independent or predictor variables and a dependent or criterion variable.
NURBS	Non Uniform Rational B-Spline
Occlusion Culling	Objects that are entirely behind opaque objects may be culled.
Optimisation	Improving the system to reduce bandwidth, runtime, memory requirements or processing power
Particle Systems	A technique to simulate certain fuzzy phenomena such as sparks, mist, clouds, fire, explosions, smoke, snow, fog, dust, hair, fur and glowing trails.
p.	Page
PC	Personal Computer
PCI	Peripheral Component Interconnect, a computer bus for attaching devices to a computer motherboard, running at a speed of 133 MB/sec.

PCI Express	A modification to the PCI interface that boosts the data transfer rate between the processor and the devices attached to it. PCI Express can transfer data at speeds of 250 MB/sec to 4 GB/sec. If the PCI Express chip set and the device are capable of different speeds, they will operate at the slower speed.
Phong Shading	A shading which combines a model for the reflection of light from surfaces with a compatible method of estimating pixel colours using interpolation of surface normals across rasterized polygons.
pixel	Picture Element and it means screen point in a graphic image.
Plasma	Normally used for large television displays, it consists of many tiny cells located between two panels of glass which hold an inert mixture of noble gases (neon and xenon). They are electrically turned into a plasma which then excites phosphors to emit light.
Polygon	Shapes or surfaces with vertices (corners) and sometimes referred to as triangles
Portal Culling	Also known as cell-based culling, it is based on the assumption that the scene consists of clearly separable rooms. A room is an area mostly surrounded by walls which block the view into other rooms except for some relatively small areas, called doors or portals leading into other rooms. When viewing into the room through the door, any object that is blocked by the walls will be culled.
pp.	Pages

Pro.	Proceedings
Processor	A computer chip that interprets and executes program instructions. Sometimes the processor is referred to as the CPU (central processing unit)
Projector	It takes video signal and projects the corresponding image on a projection screen using a lens system and a bright light.
Qualitative	It is subjective and intangible. It is normally used to describe information and not quantified. In the research, it is related to human judgement, experience and reasoning.
Quantitative	It exists in a range of magnitudes and can be measured in a unit
RAM	Random Access Memory – The primary temporary storage area for program instructions and data. Any information stored in RAM is lost when you shut down your computer.
Real-Time Rendering	It means a process, when on the screen is displayed a picture, user makes a response and this feedback has an effect to what is rendered during the next frame. This cycle is fast enough to fool user, that he doesn’t see individual pictures, but smooth animation. Speed of rendering is measured in fps (Frames Per Second). One fps is not an interactive process. With 6 fps feeling of interactivity rises. Fifteen fps allows user to concentrate to action and reaction. Upper limit is 72 fps because of limitation of an eye.
Refresh rate	The frequency, measured in Hz, at which your screen’s horizontal lines are recharged (sometimes also referred to as its vertical

	frequency). The higher the refresh rate, the less video flicker can be seen by the human eye.
Resolution	The sharpness and clarity of an image produced by a printer or displayed on a monitor. The higher the resolution, the sharper the image.
RGB	Red, Green and Blue, the basic channels for an image file
Sample	The part of a statistical population which is actually observed, A subset of a population.
Scripting	A programming language or code that controls a software application.
SIGCHI	Special Interest Group for Computer-Human Interaction
SIGGRAPH	Special Interest Group for Computer GRAPHics
SLI	Scalable Link Interface, not just a graphic card but a system. It leverages the power of two PIC Express graphics cards and the increased bandwidth in PCIE to gain increased performance.
Stereoscopic	A technique capable of recording three-dimensional visual information or creating the illusion of depth in an image.
Subject	Any respondent that is observed for purposes of research.
Surround Sound	Multichannel audio to channels encircling the audience using combinations of additional left, right and back surround speakers.
Survey	Statistical survey functions to collect quantitative information about items / subjects in a human population

SVGA	super-video graphics array – A video standard for video cards and controllers. Typical SVGA resolutions are 800 X 600 and 1024 X 768.
SXGA	super-extended graphics array – A video standard for video cards and controllers that supports resolutions up to 1280 X 1024.
SXGA+	super-extended graphics array plus – A video standard for video cards and controllers that supports resolutions up to 1400 X 1050.
Texture	A source of data, picture or function used to modify a surface of its appearance in every area.
TFT	Thin Film Transistor is a variant of liquid crystal display (LCD) which uses thin film transistor (TFT) technology to improve image quality.
Triangle	The most shape for 3D models in computer graphics and it consists of 3 corners or vertices and three sides or edges. Sometimes can also be called polygons.
UXGA	Ultra-extended graphics array – A video standard for video cards and controllers that supports resolutions up to 1600 X 1200.
Variable	A symbolic representation to denote a quantity or expression that has the potential to change
Vertex	Denotes point in space with its other properties like normal, colour, transparency and others.

Video memory	The memory that consists of memory chips dedicated to video functions. Video memory is usually faster than system memory. The amount of video memory installed primarily influences the number of colours that a program can display.
View Frustum Culling	A geometric representation of the volume visible to the virtual camera. Objects outside this volume will not be visible so they are discarded. Those at the boundary will be cut into pieces.
Virtual Reality	A technology of interaction with an environment simulated by computer. The experience is primarily visual with options to have surround sound and haptic.
Visualisation	A technique for creating multiple images (animation) to communicate design, idea and architectural spaces.
VR	Virtual Reality
VRML	Virtual Reality Modeling Language
VSYNC	Vertical Synchronization refers to the synchronisation of frame changes with the vertical blanking interval, thus ensuring only whole frames are seen on-screen.
Workstation	Normally a computer with very high-end technical specifications which is normally used as a server
WXGA	Wide-aspect extended graphics array – A video standard for video cards and controllers that supports resolutions up to 1200 X 800.
XGA	Extended graphics array – A video standard for video cards and controllers that supports resolutions up to 1024 X 768.