Universitat de Lleida

Optimization of Display-Wall Aware Applications on Cluster Based Systems

Ismael Arroyo Campos

http://hdl.handle.net/10803/405579

Universitat de Lleida

DOCTORAL THESIS

Optimization of Display-Wall Aware Applications on Cluster Based Systems

Ismael Arroyo Campos

This thesis is presented to apply to the Doctor degree with an international mention
by the University of Lleida
Doctorate in Engineering and Information Technology

Director
Francesc Giné de Sola
Concepció Roig Mateu

Tutor
Concepció Roig Mateu

2017

Resum

Actualment, els sistemes d'informació i comunicació que treballen amb grans volums de dades requereixen l'ús de plataformes que permetin una representació entenible des del punt de vista de l'usuari. En aquesta tesi s'analitzen les plataformes Cluster Display Wall, usades per a la visualització de dades massives, i es treballa concretament amb la plataforma Liquid Galaxy, desenvolupada per Google. Mitjançant la plataforma Liquid Galaxy, es realitza un estudi de rendiment d'aplicacions de visualització representatives, identificant els aspectes de rendiment més rellevants i els possibles colls d'ampolla. De forma específica, s'estudia amb major profunditat un cas representatiu d'aplicació de visualització, el Google Earth. El comportament del sistema executant Google Earth s'analitza mitjançant diferents tipus de test amb usuaris reals. Per a aquest fi, es defineix una nova mètrica de rendiment, basada en la ratio de visualització, i es valora la usabilitat del sistema mitjançant els atributs tradicionals d'efectivitat, eficiència i satisfacció. Adicionalment, el rendiment del sistema es modela analíticament i es prova la precisió del model comparant-ho amb resultats reals.

Resumen

Actualmente, los sistemas de información y comunicación que trabajan con grandes volúmenes de datos requieren el uso de plataformas que permitan una representación entendible desde el punto de vista del usuario. En esta tesis se analizan las plataformas Cluster Display Wall, usadas para la visualización de datos masivos, y se trabaja en concreto con la plataforma Liquid Galaxy, desarrollada por Google. Mediante la plataforma Liquid Galaxy, se realiza un estudio de rendimiento de aplicaciones de visualización representativas, identificando los aspectos de rendimiento más relevantes y los posibles cuellos de botella. De forma específica, se estudia en mayor profundidad un caso representativo de aplicación de visualización, el Google Earth. El comportamiento del sistema ejecutando Google Earth se analiza mediante diferentes tipos de test con usuarios reales. Para ello se define una nueva métrica de rendimiento, basada en el ratio de visualización, y se valora la usabilidad del sistema mediante los atributos tradicionales de efectividad, eficiencia y satisfacción. Adicionalmente, el rendimiento del sistema se modela analíticamente y se prueba la precisión del modelo comparándolo con resultados reales.

Summary

Nowadays, information and communication systems that work with a high volume of data require infrastructures that allow an understandable representation of it from the user's point of view. This thesis analyzes the Cluster Display Wall platforms, used to visualized massive amounts of data, and specifically studies the Liquid Galaxy platform, developed by Google. Using the Liquid Galaxy platform, a performance study of representative visualization applications was performed, identifying the most relevant aspects of performance and possible bottlenecks. Specifically, we study in greater depth a representative case of visualization application, Google Earth. The system behavior while running Google Earth was analyzed through different kinds of tests with real users. For this, a new performance metric was defined, based on the visualization ratio, and the usability of the system was assessed through the traditional attributes of effectiveness, efficiency and satisfaction. Additionally, the system performance was analytically modeled and the accuracy of the model was tested by comparing it with actual results.

Preface

The thesis has been structured in order to facilitate understanding of the amazing field of cluster display wall environments this work is focused on. It presents the usability tests, analysis and behavior modeling carried out on an specific cluster display wall, Liquid Galaxy, when running a range of applications with different hardware infrastructures and types of users. By doing so, it helps to identify the capabilities and performance bottlenecks giving accurate estimated user satisfaction values. Accordingly, Chapter 1 describes the alternative cluster visualization systems together with their associated drawbacks and the main objectives of this thesis. Next, Chapter 2 presents the background of cluster visualization systems. In Chapter 3, the performance parameters of the Liquid Galaxy infrastructure built with commodity hardware are evaluated and performance issues are also identified. Chapter 4 analyzes user behavior when running the Google Earth application in Liquid Galaxy using both homogeneous and heterogeneous systems. This section also includes scalability studies, increasing the number of nodes in the clusters from 3 to 8. In Chapter 5, the usability of the Liquid Galaxy when running the Google Earth in different environments is studied. In Chapter 6 shows the modelization of the system behavior and how to predict the behavior of the same system under other parameter values. Finally, Chapter 7 concludes the paper and discusses future directions.

Acknowledgments

I would like to thank Dr. Francesc Giné and Dr. Concepció Roig for having such patience, enthusiasm and dedication to guide me through all of this work and helping me with my shortness of words. I am also very happy to have worked with Dr. Josep Conde i Colom and Dr. Toni Granollers whose wisdom and knowledge have helped me go through hard roads.

I would like to thank my family, friends and beloved ones for supporting me and giving me courage through these years, even when I flew to the other side of the world.

And finally, thanks to Andreu Ibáñez and the University of Lleida for giving me such opportunity to work with the technology of Google and to be able to apply to the Google Summer of Code with such good support and mentorship. This opportunity led me to know the welcoming and charming people from EndPoint Corp. that gave me the chance to live and work in a city that I will remember my whole live.

Thank you.

This thesis has received a grant for its linguistic revision from the Language Institute of the University of Lleida (2017 call).

Contents

1 Introduction **21**

1.1 Multi-Display Visualization Environments 22

1.1.1 Display Wall 23

1.1.2 Cluster Display Wall 24

1.2 Liquid Galaxy: A Cluster Display Wall 25

1.3 Hypothesis and Work Objectives 26

2 State of the Art **29**

2.1 Current Infrastructures 30

2.1.1 Visualization Devices 30

2.1.2 Hardware Configurations 31

2.1.3 Middlewares 32

2.1.4 Applications 36

2.1.5 Liquid Galaxy vs the Other Visualization Systems 37

2.2 Performance Studies in Cluster Display Walls 38

2.3 Current Usability Studies in Cluster Display Walls 39

2.3.1 Usability Context 40

2.3.2 Satisfaction Works 41

2.3.3 Efficiency and Effectiveness Works 42

2.4 Motivation 43

3 Liquid Galaxy System **45**

3.1 Liquid Galaxy Infrastructure 46

3.2 Performance based on the Client-Server Model 47
3.2.1 Web-based Applications 47
3.2.2 Gaming Applications 52
3.3 Performance based on the Master-Slave Model 56
3.3.1 Google Earth 56
3.4 Additional Tools Developed for the Liquid Galaxy Project 63

4 Google Earth Performance 65
4.1 Visualization Rate (VR) 65
4.2 Benchmarking Tests 67
4.2.1 Homogeneous System 67
4.2.2 Heterogeneous System 69
4.3 VR Related to User Behavior 75

5 Usability Analysis of the Liquid Galaxy Platform 77
5.1 Objectives of the Usability Analysis 78
5.2 Driven test 79
5.2.1 Environment 79
5.2.2 Participants 80
5.2.3 Test 81
5.2.4 Satisfaction Results 81
5.2.5 Effectiveness and Efficiency Results 83
5.2.6 Discussion 84
5.3 Laboratory test 86
5.3.1 Environment 86
5.3.2 Participants 88
5.3.3 Test 89
5.3.4 Satisfaction Results 94
5.3.5 Effectiveness and Efficiency Results 97
5.3.6 Discussion 99
5.4 Field test 100

5.4.1 Environment 100
5.4.2 Participants 101
5.4.3 Test 102
5.4.4 Satisfaction Results 103
5.4.5 Effectiveness and Efficiency Results 105
5.4.6 Discussion 106
5.5 Conclusion 107

6 Modeling **109**
6.1 Performance modeling 109
6.2 Machine modeling 112
6.3 User Modeling 115
6.3.1 Modeling of the Learning Period 115
6.3.2 Modeling of T_{Idle} 117
6.4 VR Modeling 118

7 Conclusion and Future Work **121**
7.1 Future Work 126

A Appendix **129**
A.1 VR modeling 129

List of Figures

1-1 Display Wall 22
1-2 Lleida Liquid Galaxy 25

2-1 Parts of a multi-display visualization environment 30
2-2 Cluster display wall infrastructures 33
2-3 Kinds of application running on a Cluster Display Wall 36

3-1 Peruse-a-rue 47
3-2 Peruse-a-rue schema 48
3-3 Images of the two tested environments 49
3-4 Peruse-a-rue - CPU Load for Barcelona tour 50
3-5 Peruse-a-rue - External networking for Barcelona tour 51
3-6 Quake III Arena running in Liquid Galaxy 52
3-7 Quake schema 53
3-8 Percentage of CPU usage - Quake III 54
3-9 Memory usage - Quake. 55
3-10 Internal Networking - Quake 55
3-11 Liquid Galaxy architecture running Google Earth 57
3-12 Images of the three tested environments 59
3-13 Percentage of CPU usage - Squid ON. 60
3-14 Memory usage - Squid ON. 61
3-15 External and Internal Networking - Squid ON. 61
3-16 External Networking - Squid Off. 62
3-17 Percentage of CPU usage - Squid Off. 63

4-1 VR: CPU Usage. 66

5-1 Measuring usability 78

5-2 3D Space Navigator Movement Options 80

5-3 Tour Flowchart - T1 Central Park - T2 Liberty Island - T3 Brooklyn - T4 Brooklyn Bridges - T5 Union Square - T6 Empire State. Each picture is the composition of the 3 screens of the system 82

5-4 Discount usability test results 83

5-5 Tobii Pro Glasses 2 87

5-6 Tour Flowchart - T1a Statue of Liberty from above - T1b Statue of Liberty crown - T2a Barcelona Football Club stadium - T2b The stadium screen - T3a Sydney Harbour - T3b Luna Park - T4a Lleida bridge. Each picture is the composition of the 3 screens in the system 90

5-7 Emotional choices (source LemTool) 92

5-8 T2b Heat map - Common Pattern in the whole study 94

5-9 Mean and error bars from post-test questions - Skilled group (16 people) 96

5-10 Mean and error bars from post-test questions - Less skilled group (11 people) 96

5-11 Mean and error bars from post-test questions - Skilled group (5 people) 104

5-12 Mean and error bars from post-test questions - Less skilled group (8 people) 104

6-1 Scheme to model VR 110

6-2 MXL learning curves 116

6-3 Values of VR and VR_model for for MXL3 to MXL5 (left) and MXL1 to MXL2 (right) 120

A-1 3D representations of the real and estimated VR 130

List of Tables

2.1 Characteristics of Middleware 35

4.1 VR obtained from Benchmarking Tests 67
4.2 VR related to scalability 68
4.3 Kinds of node 71
4.4 Liquid Galaxy composition 71
4.5 VR related to heterogeneous clusters for $n = 3$ 72
4.6 VR related to the Scalability 74

5.1 Number of people in relation to age ranges and MXL 81
5.2 VR for every MXL category 85
5.3 Learning times according to MXL level 85
5.4 Basic Participant Information 88
5.5 MXL in relation to age range 89
5.6 Description of test tasks 92
5.7 Post-test Questionnaire 93
5.8 Results of Q1 Question 95
5.9 Results of Q2 Question 95
5.10 VR Ranges for T1a, T2a and T3a sub-tasks 98
5.11 Variances and Correlations for Q1, Q2 and VR 99
5.12 MXL information grouped by age and profession 101
5.13 MXL in relation to age ranges 102
5.14 Results of Q1 Question 103
5.15 Results of Q2 Question 103

5.16 VR Ranges for the Different Environments 105
5.17 Variances and Correlations for Q1, Q2 and VR 106

6.1 Test results 113
6.2 Time per Draw function (T_{Draw_ref}) 113
6.3 T_{idle} average according to MXL and interest 117
6.4 T_{idle_ref} according to data density and interest for a reference cluster . 118
6.5 VR model estimation and VR real values 119

Chapter 1

Introduction

Nowadays, new trends in information and communication technologies, storage density and increasingly sophisticated data acquisition technologies have led to an explosion in data. The processing of this huge amount of data has to be handled differently in order to obtain all the information at once, which, in consequence, greatly increases the quality of the study of these data. As the information to be visualized becomes more complex, the visualization environments have to meet its requirements. Thus, in a large number of cases, a traditional single low-resolution display is unable to cope with these requirements. This need has led to the development of a new screen technology to achieve really high pixel density in one screen. The counterpart is that this technology is still too expensive for common use and other alternatives are needed. In order to fulfill this necessity, new visualization methods that consist of using multiple displays or projectors are appearing [67].

In this chapter, the leading current multi-display visualization environments and their main features are presented. This then is followed by a presentation of a specific environment this work is focused on, the Liquid Galaxy system [37]. Finally, the objectives of the present work and the reasons behind these are explained.

Figure 1-1: Display Wall

1.1 Multi-Display Visualization Environments

A multi-display visualization environment is a system in which there is more than one display or projector to increase the visualization area. The most common distribution of displays is a display wall, a set of tiled displays forming a grid or a wall (Figure 1-1). By using multiple displays or projectors, the field of view and workplace becomes much larger and enables patterns or common behavior to be found visually at first glance. These systems were designed to provide a better way of analyzing the ever-increasing large data sets produced by advanced technology projects. Nowadays, a big set of new applications that generate a huge amount of data is appearing. So, these applications are the target of this new kind of visualization system. According to Chung et al. in [18], application domains of Large High-Resolution Displays can be classified as follows:

- Immersive Virtual Environments and Modeling. These applications rely on static models and an interface where the user can watch 3D models and interact with them. Some applications include geospatial exploration [33], architecture walkthroughs [21, 25] and design exploration [16].
- Scientific Visualization. These applications tend to represent massive amounts of scientific data at once. This allows the user to study all the data obtained in high-resolution without losing the general context. Some examples can be found

in such fields as biomedical science [79], genomics [91], geosciences [32][57], space science [45] and architecture [80].

- Command and Control. In this application domain, there is a command center with various computers connected to it that send their information in the same or in different windows to it. This allows collaboration between different users while they are sitting in their own personal workspaces. This type of application supports such different fields as the military [84], aerospace, telecommunications [93] and large facility management.

- Imagery and Multimedia Viewing. These applications enable the user to visualize high-resolution data in a wider context. Commonly, these applications rely on using few polygons [73] but use a large amount of high-resolution textures [95] across the different displays. So, these applications are generally optimized to synchronize the data among displays. This type of application enables analysis of the imagery produced by satellites [59], electron microscopes [81], or the viewing of different images organized into a single display wall [58]. Also, it is possible to play synchronized videos [28] or videogames [72] and visualize geographic multimedia applications like Google Earth [35].

The multi-display is just the way in which the information delivered by the application is visualized, but these displays can be managed differently depending on their back-end infrastructure. Consequently, we can classify the multi-display visualization environments into two groups: Display Walls and Cluster Display Walls.

1.1.1 Display Wall

A display wall is an infrastructure in which various screens are distributed in tiles connected to a single powerful computer integrated with multiple video outputs, enabling, in some cases, the construction of a low-cost platform. However, if there are many displays to be managed by the computer, it might need a costly upgrade. The displays are laid out close together to try to minimize the bezel size to create the illusion of only one big display. Depending on the display wall setup, typically small or medium size display are chosen. Thus, this might be the cheapest solution when the application to be visualized requires low-resolution analysis.

The main problem that arises when using display walls is that the images are stretched, as the resolution of a single screen is resized to fit all the screens together. In order to overcome this problem, one possibility is to use high-resolution screens, like the 4K enabled screens [94], with the drawback that the cost would be increased by one order of magnitude. Furthermore, specialized video processors, as well as an upgraded motherboard and CPU would be needed to manage a very large video wall, raising the price of the system drastically. Likewise, it is worth pointing out that this kind of system presents a serious problem of scalability given that the maximum number of displays is limited by the server system constraints.

1.1.2 Cluster Display Wall

A cluster-based display wall consists of a number of synchronized PCs with each node of the cluster having one or more displays connected to it. A cluster can be used in any type of display wall to improve the quality of the visualization by increasing the pixel density efficiently. This solution converts any visualization environment into a scalable high-resolution capable system. These systems are interconnected through a local network, such as Ethernet, or a sustained throughput network, like the Myrinet [14], where visualization of the images across the screens of the cluster display wall is carried out by synchronization protocols to pass data between them. Some examples of cluster display walls are CAVE [24], GeoWall [57] and Garuda [71], which are described in detail in Chapter 2.

Nowadays, the availability of software packages for clusters makes setting up cluster display walls affordable. Thus, as stated in the Survey[18], cluster-based displays are performance, memory and display scalable, easy to maintain and upgrade and simple to add new nodes to. Upgrading a cluster-based display wall is as easy as upgrading the individual nodes in the cluster or adding new nodes. This opens a wide range of new possibilities where the quality of the image is a more restrictive necessity and a single-node display wall will not suffice.

The majority of applications executed in a cluster-based display follow two approaches [17]: *master-slave* or *client-server*. In the master-slave applications, the dataset is mirrored across all the nodes and multiple instances of a program run in parallel, one instance on each node. An example of a master-slave application is Google Earth [35]. In the client-server approach, the server runs a different instance than the one executed by each of the clients, distributing appropriate data to each client node and synchronizing the client nodes. One of many examples of such applications is CaveSL [44], a modified version of the game Second Life.

According to their characteristics, cluster display walls have recently generated a lot of interest, because such displays have the potential to put high-performance visualization environments within the reach of more users [18, 61]. For this reason, this work will be focused on a specific Cluster Display Wall developed by Google, named Liquid Galaxy [37], as a representative case of this kind of infrastructure.

1.2 Liquid Galaxy: A Cluster Display Wall

Figure 1-2: Lleida Liquid Galaxy

The Liquid Galaxy system is a cluster infrastructure made up of eight displays, by default, each connected to a computer node, which provides an immersive visualization. Although the original project had 8 nodes and displays, the cluster can be extended depending on the infrastructure. As a cluster, it has easy scalability by being easily expandable and minimizes its cost by requiring low-cost infrastructure. Although it was originally conceived to run the Google Earth application [35], different applications can run in the Liquid Galaxy. Some examples are Quake III Arena [72], WebGL applications [92] such as Aquarium [41], or video streaming [28]. Figure 1-2 shows the Google Liquid Galaxy infrastructure installed in the Technological Park in the city of Lleida (Spain) [75] and running the Google Earth.

Likewise, Figure 1-2 shows the most usual device used to interact with the system, which is the 3DConnexion Space Navigator [1]. This controller allows navigation and rotation through all the axes to provide a full experience with Google Earth.

The main features of the Liquid Galaxy system are that it is open source, developed and maintained by a big company like Google, and its ease of installation. As a curiosity, this project was born in 2008 from volunteers in a "20% project" during which Google engineers can use 20% of their work time to join in or develop new projects. Nowadays, this new technology is slowly expanding across the world. It can be already found in NASA [65], where they improved the original design by including the use of ClusterGL [66], which enables graphics to be drawn without the need for the applications themselves. Note that this is one of the main advantages of Liquid Galaxy given that being an open-source project means its possibilities for growth depend on the infinite applications that developers create for this system. As an example of this, there are also private companies like EndPoint [29], which install Liquid Galaxy systems, develop their own applications and provide customer service for these systems.

1.3 Hypothesis and Work Objectives

Taking into account the growth of new applications oriented to cluster display wall together with the low-cost requirements of the Liquid Galaxy platform, this thesis lay out the following hypothesis and null hypothesis:

Investigation hypothesis: A Liquid Galaxy built up with low-cost infrastructure is able to execute any kind of application with a good performance and usability from the point of view of the user. In the case that this hypothesis is validated, it makes sense to do a study of the system parameters that will permit to establish a mathematical model to correlate the measures of performance with user experience.

Null hypothesis: A Liquid Galaxy built up with low-cost infrastructure is not able to execute any kind of application with a good performance and usability from the point of view of the user.

According to our hypothesis, our main objective is to extend the use of cluster display walls to all kind of people and environment. For this reason, we study an specific system, named Liquid Galaxy, to analyze its behavior and the feasibility of building this kind of system with commodity hardware. In order to complete this main objective, we set the following objectives that will be developed throughout this work:

- Analyze performance. We want to analyze the main performance metrics of the system (CPU, memory and networking) in order to know its hardware limits while running diverse applications in different scenarios. By knowing the limits and bottlenecks of the system, we can determine the hardware requirements of the system and, thus, drastically lower the price of its parts or even use commodity and/or already-available hardware. This would extend the use of this kind of system to new groups of potential users, such as educational or commercial environments. In order to do this, the main system parameters will be monitored by means of a set of benchmarks developed specifically, analyzing statistically the possible correlations between those parameters.

- Study heterogeneity and scalability. In order to use already-available hardware or expand an existing cluster display wall, we need to know how the system will perform with new and presumably different hardware.

- Evaluate usability. It must be taken into account that one of the most important parts of any system that it is aimed to analyze is the interaction of the user with the system. The usability of a system covers different aspects of user behavior and, in this work, this is evaluated by using the three traditional attributes of usability: satisfaction, effectiveness and efficiency. In order to do so, the usability is evaluated in different day-to-day scenarios and several kinds of users, while studying aspects such as their technological experience and motivation. The satisfaction of the users will be obtained by means of a set of pre/post questionnaires. Likewise, a new metric to measure the efficiency and effectiveness of the users in this kind of system is proposed and monitored throughout all the tests.

- Identify the relation between user behavior and performance. Once we know both the system performance for a given Liquid Galaxy configuration and the user behavior for a given application, these have to be related. In order to do so, we will analyze the correlations between the satisfaction of the users, given by the pre/post questionnaires, with the performance of the system, given by the new proposed metric.

- Model the performance of the system. In order to be able to tell if any given hardware has the potential to run a specific multimedia application, a model that can predict the performance of the system is proposed. This model will aid in knowing beforehand if any given infrastructure is able to run a multimedia application within acceptable performance. In order to do so, we will explore trends in the different parameters and the correlation between the parameters that affect the overall experience. Likewise, our model will be matched with the real results obtained in the tests with real users.

Chapter 2

State of the Art

Cluster-based display walls provide cost-effective and scalable display infrastructures with high resolution and large display area, making them suitable for a wide range of high-resolution applications. As a consequence, this has aroused the interest of the scientific community, and a wide range of new cluster-based display wall platforms have been proposed together with software frameworks orientated to handling the details of synchronizing and distributing the rendering tasks across these nodes [18, 61]. Some of these systems are focused on enabling collaboration between users through screen sharing in the same room, while others are oriented towards on displaying 3D models or analyzing scientific data by using display walls. Accordingly, the performance and satisfaction of their users have provoked the interest of some researchers, and as a consequence, many works on usability tests for these kinds of systems have been published.

The first section of this chapter, dedicated to the state of art, reviews the main cluster display wall infrastructures together with the middleware installed on them and the kind of applications that use these infrastructures. Moreover, a comparison between the Liquid Galaxy and the other infrastructures is described. The second section analyzes the literature focused on the performance of cluster display walls. Likewise, it is important to know how the user interacts with this kind of environment and, thus, the third section describes the main works in the literature devoted to the study of the usability aspects of cluster display walls. The final section explains our motivation for carrying out a study of the specific Liquid Galaxy cluster display wall.

2.1 Current Infrastructures

A multi-display visualization environment can be split into four components as shown in Figure 2-1. These are the device/s used to project the images, the application provides the visualization data, the middleware used to display these data, and the configuration of the computer/s, including the interconnection between them. Next, a review of the main contributions of the literature for each component is described.

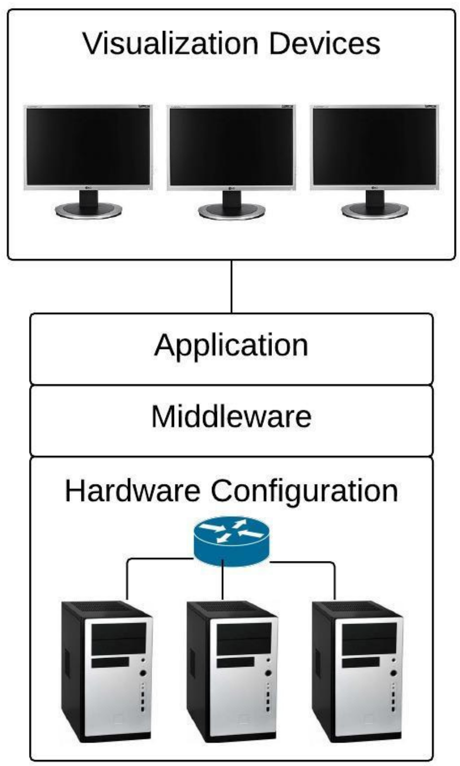


Figure 2-1: Parts of a multi-display visualization environment

2.1.1 Visualization Devices

Depending on the devices used to visualize the multimedia data, there can be a necessity to create virtual immersive environments to have a big work interactive space or allow collaboration between multiple users. According to this, different solutions involving the type or physical positioning of the devices have appeared.

The two main types of visualization devices nowadays are LCD monitors and projectors [67]. The evolution of the LCD panel making the bezels thinner has led this kind of device to become a cheap and easily configurable component to visualize data using multiple displays. This is the most common solution in display walls, in which displays are tiled to form a wall, table, room or even circular formations, as LCD panels are easy to align and are color correct. In the case where the visualization hardware is made up of multiple projectors instead of LCD monitors, the images are blended perfectly, giving a much more immersive experience than with LCD monitors. The drawback is that projectors are more expensive and have a costly maintenance.

An additional feature that includes both devices is that they can be used to create stereoscopic images, where two sets of the same image are displayed, one for the users' left eye and one for the right eye. The user usually needs to wear some kind of glasses to see the 3D image, which gives the user a better spatial perspective [24, 22, 42].

2.1.2 Hardware Configurations

There are two types of hardware configurations to manage the displays or projectors involved in a multi-display visualization environment, ones that rely on a single computer, and others that use a cluster infrastructure [67]. Currently, the single computer infrastructure approach is useful to manage a small number of displays with high-resolution applications or small to medium display walls that do not need high-density visualization. The drawback of using a single computer is that it needs powerful hardware components in order to manage the displays, as the increase of displays also increases the size and the resolution of the images visualized. Thus, the workload becomes higher without the computing capacity being increased. This scalability problem can be solved by using a cluster infrastructure, which provides better scalability without affecting the cost. However, this kind of system needs synchronization and data passing protocols in order to display the visual data correctly [48]. Thus, the network connection must be stable, fast and reliable, given that it becomes the communication interface where all these data is passed through. Some network solutions available are 100-Mb and Gigabit Ethernet, Myrinet [31] and OptIPuter [85] networks. Moreover, the data that is passed is distributed differently depending on the cluster configuration and type of application. That is why abstraction techniques using frameworks or middlewares are installed in these cluster systems to manage the data distribution among displays easily.

2.1.3 Middlewares

As stated previously, when the infrastructure is cluster-based, it usually relies on a middleware or a framework, which facilitates the synchronization among computers to display images, multimedia or any kind of visual data, by abstracting the user on how the application works on a certain hardware configuration. Because of this, applications that are displayed become easy to install and configure, allowing almost any type of application to be run on these systems.

Some of the most representative examples of middleware used in cluster display wall environments are the following:

- SAGE [78], a window manager system, where there are a number of rendering resources shared over the network and displayed in a display wall for interaction.

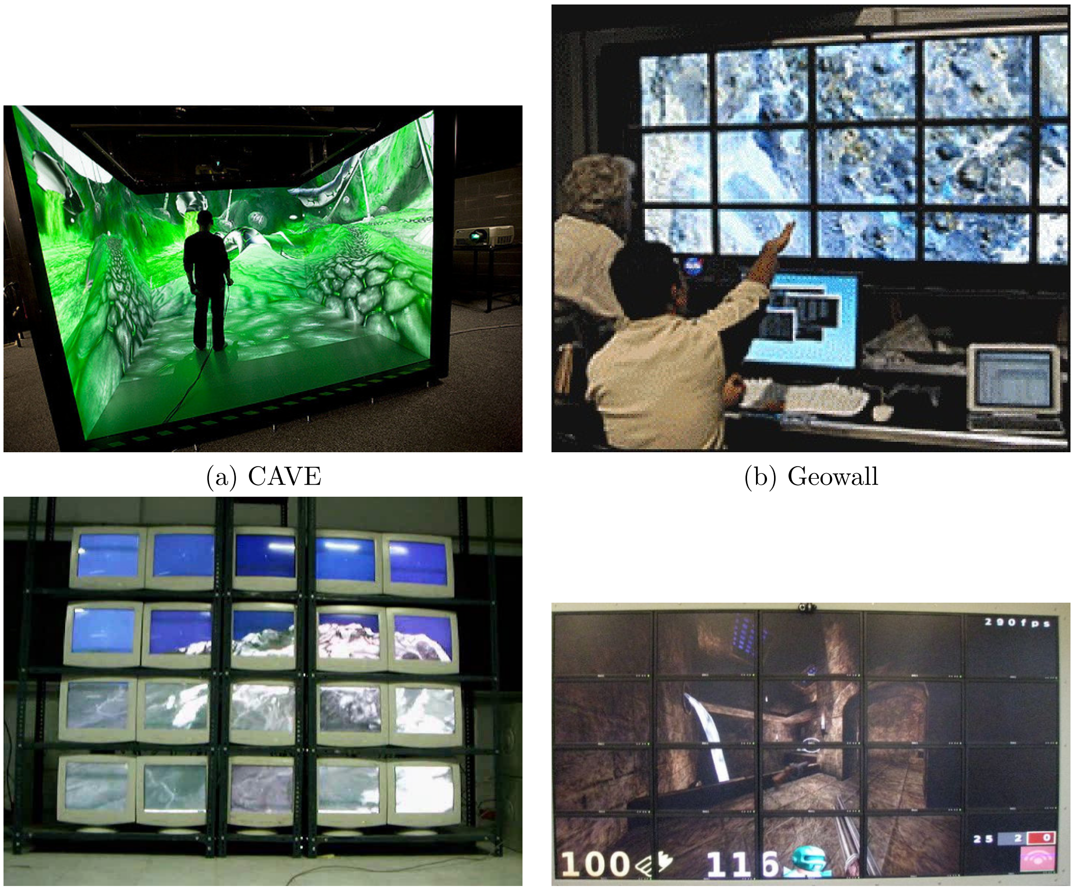
- Dynamo [56], an interactive surface that allows sharing, displaying and exchanging media with other people in the room.

- Equalizer [27], a scalable parallel rendering framework that optimizes the use of multiple accelerated graphics hardware.

- Chromium [50], a stream-processing framework for interactive rendering on clusters.

- CGLX [26], a scalable, high-performance visualization framework for networked display environments.

A middleware can be developed to be adapted to a wide range of infrastructures, such as the previous ones, or for a specific cluster-based infrastructure. In the latter case, what makes them unique is the combination of the infrastructure and the middleware designed for it.



(a) CAVE (b) Geowall

(c) Garuda (d) WireGL

Figure 2-2: Cluster display wall infrastructures

Some examples of current cluster-display walls with their specific middleware are listed next:

- CAVE [24] (Figure 2-2a), an immersive environment which simulates a cave, where all the walls and the floor are projected. This system uses two projectors for each surface and a user detection system to create a 3D experience, where the user can move around the room while he/she can still interact with the application using a wireless controller. CAVE also includes a tracking system that modifies the application to adjust the perspective of the user, which also makes the system aware of where the user is pointing the controller.

- GeoWall [57] (Figure 2-2b), an infrastructure built as a set of tiled 3D displays on a solid structure making a wall. It provides a passive 3D experience by using polarized screens and glasses. Although expensive, this system is cheaper than buying a single big 3D stereoscopic display.

- Garuda [71] (Figure 2-2c) is a commodity cluster display wall that can display Open Scene Graph API (OSG) [74] applications. It is fully scalable and uses a high-end computer as a server and low-cost computers for the client nodes. This system is a good solution for high-resolution visualization for OSG applications, where the code does not need to be modified.

- WireGL [49] (Figure 2-2d) is a scalable cluster that uses off-the-shelf PCs to display unmodified existent applications into a display wall. The system uses a modified network protocol to accelerate OpenGL packet synchronization. Although it is still in development, this solution, which allows the use OpenGL [40] applications without modifying the code, looks promising.

- Liquid Galaxy [37] (Figure 1-2) is a commodity cluster display wall where the screens form a hemisphere to give immersivity. This system provides a good solution for running applications in a low-cost environment.

When building a cluster display wall, in terms of cost, the biggest differences are the type of visualization devices used. In these examples, the systems that use projectors have higher cost and more expensive maintenance than the ones that use displays. Moreover, clusters the use of special displays or additional hardware, like the tracking system in CAVE or the 3D displays in GeoWall.

In the inner operation of middleware, some characteristics can be distinguished that are presented next [18]:

- Task distribution model. Specifies the point in the rendering process where information is distributed to the other nodes in the cluster. The model is named distributed application if the application and data are replicated to every node at the beginning. Then, the portion of the image to be computed by each node is determined through synchronization packets among nodes. The other approach used is the distributed renderer model, where the image is split by one node and sent to the other nodes, which only receive the data to be visualized

- Programming model. Depending on how much a graphics API has to be modified to enable it to be used in the framework, this parameter can take three degrees of invasiveness. A non-invasive model allows APIs to be used transparently. Minor code modifications models need small changes to run these APIs. A structurally invasive model is when the application to be run needs to be specifically written for the cluster display wall.

- Graphics API. This parameter specifies the APIs that are used in the cluster display wall. The best known, documented and widely used API for cluster display walls is OpenGL API [40]. This is why most cluster display walls aim to run applications made by this API.

Table 2.1: Characteristics of Middleware

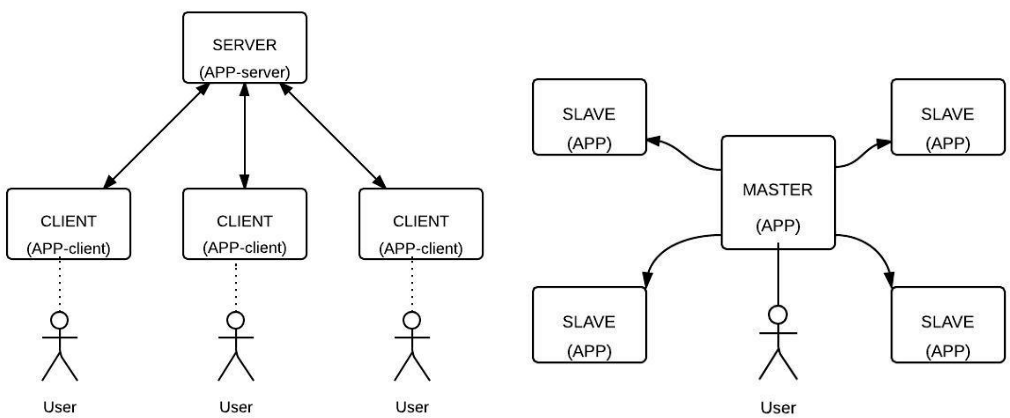
Name	Task Distribution	Programming Model	Graphics APIs
CAVE	Distributed application	Structurally invasive	OpenGL
CGLX	Distributed application	Minor code modification	OpenGL, GLUT, 3D graphics API
Chromium	Distributed renderer	Non-invasive	OpenGL, 3D graphics API
Dynamo	Distributed application	Non-invasive	No dedicated API
Equalizer	Distributed renderer	Structurally invasive	OpenGL, OSG, 3D graphics API
Garuda	Distributed renderer	Non-invasive	OSG, 3D graphics API
Geowall	Distributed application	Minor code modification	OpenGL, Blitz3D, 3D graphics API
Liquid Galaxy	Distributed application	Minor code modification	No dedicated API
SAGE	Distributed renderer	Non-invasive or minor code modification	No dedicated API, supports OpenGL
WireGL	Distributed renderer	Non-invasive	OpenGL, 3D graphics API

Table 2.1 summarizes the mentioned characteristics that are associated with the most representative middlewares and cluster display wall infrastructures. As can be seen, there is no predominance in the task distribution model, as both the distributed application and the distributed renderer models are used. Regarding the programming model, the best practice is to program under a non-invasive model to allow

applications to be run in the cluster display wall without modification. This is why more non-invasive and minor code modification models rather than structurally invasive ones can be seen in this table. As for the APIs used, there is wide acceptance of the OpenGL API and, therefore, it is the most commonly used API. It is worth pointing out that the use of an API is independent of the programming model and the task distribution model. These models are set by the kind of middleware used.

2.1.4 Applications

As explained in the previous sections, there are many types of visualization systems, but depending on the configuration of the system and the communication protocol to be applied, we can differentiate the following two types of cluster display wall applications [83].



(a) Client-Server (b) Master-Slave

Figure 2-3: Kinds of application running on a Cluster Display Wall

- Client-server: In the client-server model (Figure 2-3a), there are two sides of the application running on the system. The server is running the server-side application (APP-server), which is the main application, and which stores and refreshes the clients' status. Each client runs an instance of the client-side application (APP-client), which is connected to the server and modifies their status and/or some of the content of the APP-server.

The client-server architecture can be used to execute different types of applications with variable degrees of interaction. Interactive applications are those that allow a user to access a client node and modify its status. Consequently, a client node will send the changes to the server, which notifies the other clients about these changes. Most videogames, like Second Life [62] or Cube 2: Sauerbraten [89], use this principle to update each users view depending on the action of the other clients.

Non-interactive applications do not usually allow any changes by the user. However, in the case that the user could make some, these would not affect either the server or the other clients status. Examples of non-interactive application are VLC Media Player [90] or the in-flight entertainment systems [64].

- Master-slave: In the Master-slave model (Figure 2-3b), each node runs the same application (APP) and the master has to manage the synchronization among slaves to ensure consistency. The master is the only node that accepts user input. Thus, the user only interacts with that node, and every time the master modifies its status, it sends the changes to the slaves. Applications for this kind of architecture are usually interactive, such as Google Earth [35] or ZygoteBody [98], an application to visualize the parts of the human body.

2.1.5 Liquid Galaxy vs the Other Visualization Systems

As mentioned in the previous chapter, this thesis is focused in a specific cluster display wall, named Liquid Galaxy, that uses a custom middleware which is able to run both master-slave and client-server applications. Although it was conceived to run the Google Earth application, it can be adapted to execute any kind of application. Liquid Galaxy is designed to use multiple instances of an application instead of image distribution, making it suitable when little or no modification of the original application is required, or when the images are already distributed by the streaming application [90].

Liquid Galaxy uses low-end computers over a Ethernet network to manage commodity LCD displays, making this system easily expandable with more displays and nodes, including nodes with different hardware. Thus, heterogeneity is a feature that can be exploited.

Also, instead of using tracking systems or wireless controllers like CAVE, this system uses an affordable 3D controller or a mouse with a keyboard to interact with the applications.

In contrast with the systems explained, Liquid Galaxy constitutes a cheap approach and its degree of complexity is low enough to be built without problems. Thus, it constitutes a non-dedicated system, which can potentially be exploited for use in a wide range of fields, such as education, entertainment, travel advertising or scientific research.

2.2 Performance Studies in Cluster Display Walls

Given our proposal of extending the use of Liquid Galaxy to several environments to provide them with a low-cost cluster display wall infrastructure, a key aspect that is crucial to be studied is its performance. This will show the suitability of this platform for extensive use. We focused on the works that study the performance of the cluster display wall system. This also allows us to verify the key performance parameters to take into account. Thus, the performance analyses carried out by some authors in the literature with different cluster display walls were studied.

Humphreys et al. [49] described their proposal to render OpenGL applications using their WireGL middleware and compared its performance with another middleware named Broadcast [51]. They ran three different benchmarking applications over a variable distribution of displays ranging from 1x1 to 8x4 devices, while monitoring the performance in frames per second. They stated that their system was a bit slower when the distribution was 1x1 but it maintained the performance across all configurations whilst the Broadcast middleware did not. The performance and scalability results of this middleware are interesting, as it can be scaled without losing performance, but there was no satisfaction testing involved.

Neal et al. [66] studied the performance of ClusterGL in comparison with Chromium and BroadcastGL in a network bandwidth constrained environment. For this work, they used the Symphony display wall [88] to run two gaming applications using the aforementioned frameworks while logging the frames per second during the test. Also, they used different optimization techniques and noted the impact of each individual technique and the combination of these. They measured the increase in frames per second of ClusterGL against the other middlewares and the performance upgrade from the optimization techniques. This work shows a parameter (frames per second) used to measure the performance, which would have given a good user feedback if they had driven user tests, which was not the case.

In general, works that study performance, as the described in this section, are not related to usability or user experience, as they do not take the user into account. Nevertheless, these works focused on comparing the results obtained from one system with another.

2.3 Current Usability Studies in Cluster Display Walls

Another aspect that is worth considering in the context of the present thesis is to know the feelings of the users when using the Liquid Galaxy platform, feelings also known as the user experience or usability. Thus, our purpose is to study Liquid Galaxy with the three traditional attributes that define usability [68], namely satisfaction, efficiency and effectiveness, and also the relation between these. For this reason, the following sections first contextualize and describe the usability and user experience software attributes. Then, we present and analyze some significant works in the literature that study usability focused on cluster-based display walls.

2.3.1 Usability Context

The usability study of any interactive system is an important quality attribute that tells how easy it is to learn and to use an interactive system. As the main purpose of the cluster display walls is to run applications for the end-user, we are interested in studying these systems from the Human-Computer Interaction (HCI) point of view. HCI is the discipline concerned with the design, evaluation and implementation of interactive computing systems for human use and the study of major phenomena surrounding these [46]. Usability and user experience are two of the most important software attributes studied by HCI practitioners and researchers. The most recent standard [55] presents usability as the *"degree to which a product or system can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use. Usability can either be specified or measured as a product quality characteristic in terms of its sub-characteristics, or specified or measured directly by measures that are a subset of quality in use"*. As for the user experience, it is defined in [54] as *"a persons perceptions and responses that result from the use and/or anticipated use of a product, system or service"*.

For our study, we focus attention on analyzing the three usability parameters, understanding these as follows:

- Effectiveness refers to the accuracy and completeness with which specified users can achieve specified goals in particular environments.

- Efficiency is the resources used in relation to the accuracy and completeness of goals achieved.

- Satisfaction refers to the comfort and acceptability of the work system for its users and other people affected by its use.

Effectiveness and efficiency are quantitative parameters, as they can be measured in terms of performance and goal achievement. Satisfaction, on the other hand, is a qualitative and subjective parameter, making it much more difficult to measure. The next two sections describe some works that helped us to understand these parameters in relation to the cluster-based display walls.

2.3.2 Satisfaction Works

This section describes the most representative works analyzing the satisfaction of the users of a cluster display wall. All of them are described in relation to our purpose of analyzing the relationship between satisfaction and performance.

Tan et al. [86] studied how their system, called Infocockpit, can make information memorable. Infocockpit is a projector-based display wall with ambient visual and auditory displays that engage human memory for location. In their work, the authors made users complete semantic tasks that consisted of remembering pairs of words and then recalling them. In addition, they asked the participants the system they had learned each pair of words, shown in a random order, and they achieved 67% of correct answers. This also correlates with the number of correctly answered pairs, achieving 47% correct answers for the display wall in contrast with the 30% achieved in the desktop computer. They studied the results, both quantitative and qualitative, showing how many words the users could remember statistically. They concluded that the Infocockpit improves the memory of the user in relation to a desktop computer.

Ball and North [11] studied the effectiveness of a 3x3 large tiled display compared with two smaller displays. They tested both environments with a task with quantitative results based on finding targets of different sizes. They concluded that display walls that use physical navigation significantly outperform smaller displays that use pan and zoom navigation. Also, they introduced observations that the users made during the test, adding a subtle qualitative element into the research.

All of these works focused on studying the satisfaction or preference of use using multi-display environments. Despite this, the researchers did not take the system performance into account so no relation with performance can be derived.

2.3.3 Efficiency and Effectiveness Works

The previous works were related to the satisfaction, associated with the subjectivity of the participants, while this section describes the most representative works focused on the efficiency and effectiveness of the systems, which are related to the system performance. All of them are related to our purpose of defining a metric to measure these characteristics quantitatively. The literature defines effectiveness as the ability to complete a task and efficiency as how many resources are expended in completing it [53].

Liu et al. [63] compared physical navigation in front of a wall-size display against pan and zoom on single desktops. They designed a task that consists of moving disks with labels between containers so that each container holds disks of the same class, written on the label, colored green or red if they are correctly or incorrectly classified. Their work took into account the task completion time, number of actions, movements of the viewpoint and participants, etc. The results showed that large display walls are better for harder tasks or small labels, whereas a single desktop is better otherwise.

Bi and Balakrishnan [13] focused on user behavior in large-scale displays and demonstrated that users performed better using this kind of system than single desktops. The authors made participants perform everyday work in a week-long study on both single or dual-monitor desktops. Every half an hour, the participants had to write an activity log stating which system they had better experience with. Moreover, an interview after each working day was recorded in order to gain further information about the events written in the activity log. Also, the windows and mouse activity were recorded in a video to show detailed information about the tasks. The results obtained from the logged activities were processed statistically and some specific cases, when someone acted different than the rest, were commented. The observations in the activity logs from the participants were only used to compare the preference of use between the two systems, single-computer or display wall, in comparison with the tasks.

2.4 Motivation

As explained in previous sections, despite the growing literature on usability of cluster display wall systems [97, 30, 2], to the best of our knowledge, there are no studies that focus on establishing a relationship between the performance of the system and the satisfaction from the users using it. In general, these studies are focused on the performance results of specific tasks, like completion time or scores, but they do not study how they affect the user behavior and/or experience. On top of this, there are no studies about the performance or any aspect of usability with the Liquid Galaxy infrastructure.

Accordingly, we are interested in defining a new metric which gives an idea about the efficiency and effectiveness of a given cluster display wall, Liquid Galaxy in our case, while running a specific application. Likewise, this metric should be easily correlated to the satisfaction of the user.

Therefore, one goal of this thesis is to try to relate the usability aspects of the User Experience (satisfaction, effectiveness and efficiency [53]) of the Liquid Galaxy cluster using different configurations of heterogeneity and scalability. Thus, it will enable us to predict beforehand if a system will perform well enough from the user point of view without the need to carry out any user tests.

Chapter 3

Liquid Galaxy System

The Liquid Galaxy system [37] is a cluster display wall originally built to run Google Earth [35] in order to create an immersive experience for the user. Liquid Galaxy lets you navigate around the globe with its 6-axis controller, allowing you to instantly zoom in, zoom out, and turn in a completely fluid motion. You can also search and navigate to specific locations automatically using an optional touch-screen interface.

The Liquid Galaxy system presented in this chapter was built specifically to run Google Earth. However, the immersive visualization environment that Liquid Galaxy provides opens up this kind of system to be used in a wide range of applications, using master-slave and client-server models, that can benefit from this feature. Some examples of applications that can run on this system are WebGL applications like Aquarium [41], video streaming [28] and video games like Quake III Arena [72].

This chapter describes the performance tests that were carried out in this platform by running a set of different representative applications in order to monitor the main performance parameters of CPU, RAM and network. Thus, it gives a preliminary idea about our assessment of using a commodity Liquid Galaxy infrastructure to run a wide range of applications in different scenarios and potential users.

3.1 Liquid Galaxy Infrastructure

All the experimentation carried out in this chapter was done on a Liquid Galaxy platform built with off-the-shelf hardware. Thus, we wanted to evaluate the viability of using general-purpose platforms to visualize high-resolution images in a cluster-based way. The experimentation platform was a homogeneous cluster display wall made up of 3 nodes. Each node was composed of an Intel Core i5 3GHz, with 2x4GB RAM 1600 MHz, SSD 128GB, NVidia GT620 and a 32" screen. The interconnection between nodes was through a Gigabit Ethernet network with an Internet connection of 10 Mbps.

As data is accessed constantly and low access time to the disks is required, the nodes must be built with Solid-State Disks (SSDs). This is because multimedia applications normally use a disk cache in order to achieve a faster visualization of the data. In the case of using a disk cache that stores data from Internet, using SSDs is more important if the Internet traffic is heavy, as the disk must process the data petitions from the nodes and also store the downloaded data. Thus, using a hard drive cache significantly reduces the visualization time of the multimedia applications that require a high volume of Internet data.

Note that we used an initial minimum size of 3 nodes in this chapter because the width of three 32" displays, the size used in this experimentation, covers the human 60 horizontal viewing angle from a distance of 1m [52]. In the next chapter, the scalability issues that this system may have when increasing the number of nodes and displays are analyzed.

Using this Liquid Galaxy infrastructure, we carried out a performance analysis of the system by running some representative applications of the two existing models: client-server and master-slave. Accordingly, the key computing resources, CPU, RAM and networking were monitored by means of the following tools: Top, which was used to monitor the CPU and the RAM, and Tshark, a command-line based Wireshark [19] version, which is a packet sniffer and was used to monitor the network traffic.

3.2 Performance based on the Client-Server Model

As explained in Section 2.1.4, the client-server applications are characterized by the fact that the client nodes run the same application and are connected to a server node, which runs a different application, a server-side application. Thus, the server only has to synchronize the different clients who are connected to it. This section describes the main types of client-server applications that were introduced in Section 2.1.4 and how they perform on the Liquid Galaxy system.

3.2.1 Web-based Applications

Nowadays, web applications have become important tools to be used in any type of web-supported devices. Different web-based applications have been adapted for the Liquid Galaxy but all of these rely on the same method. This method commonly uses a server application based on an event synchronization Javascript tool, named Node.js [23], to synchronize the visualization of the multimedia application across screens. The WebGL applications [92] are the most widely used when there is a need to visualize or render data across a web browser.

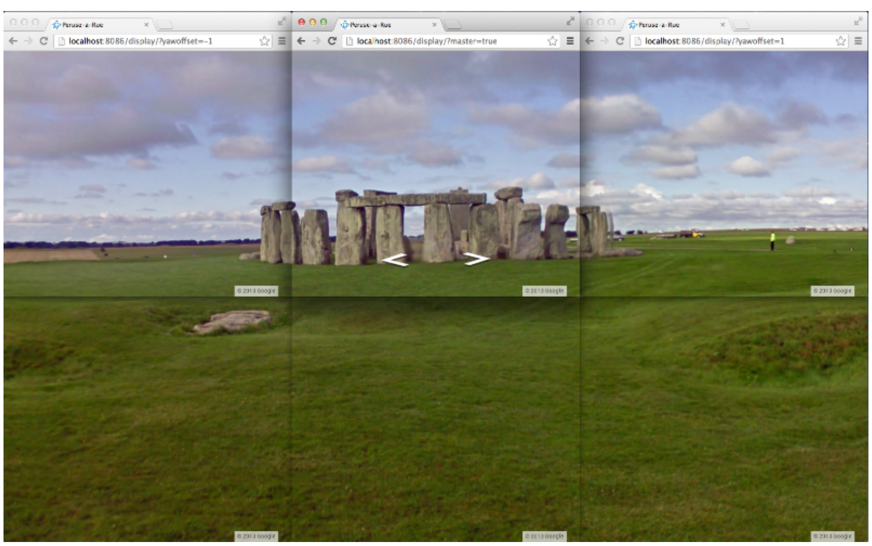


Figure 3-1: Peruse-a-rue

We studied a specific web-based application named Peruse-a-rue [36], a Google Street View modification in which you can visualize the surroundings by using 360 degree photos when navigating along the street. There are two main objects in this application, the camera, which tells where the user is looking at, and the user position, which is where the person is virtually placed on the world map. Figure 3-1 is a representation of Peruse-a-rue using multiple web browser windows, where each browser can be opened in a different computer, making it suitable for visualizing on a cluster display wall. The distribution, visualization angles and offsets of each display can be easily configured by passing parameters to the URL. In every user position of Peruse-a-rue, there is a 360° photo split up into several smaller photos of a certain resolution. The application loads these small chunks of data only when they are needed and, thus, depending on the Internet connection, the visualization of the images might be blurry for a few seconds before loading completely.

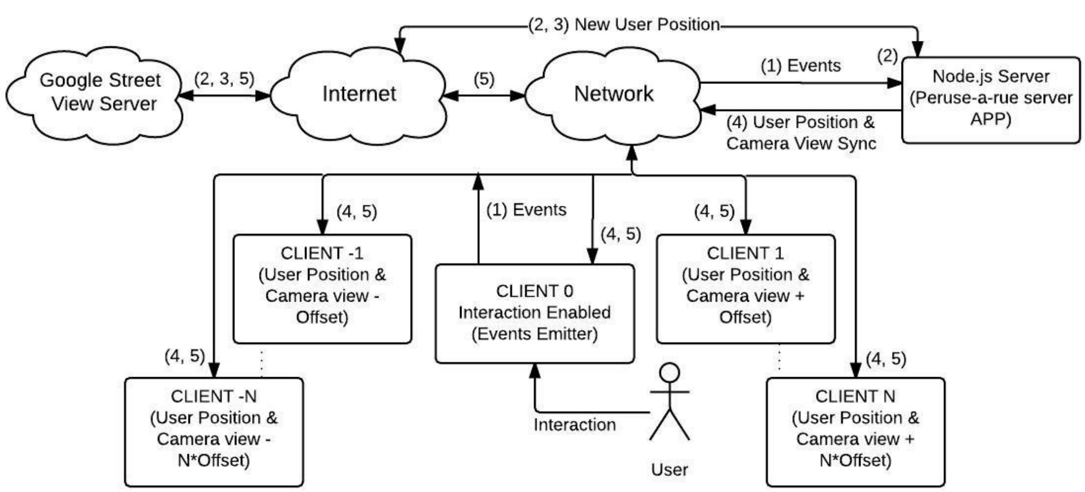


Figure 3-2: Peruse-a-rue schema

The schema of how the Peruse-a-rue runs in Liquid Galaxy is depicted in Figure 3-2. Every client node uses a web browser that is connected to the server through the local network, but the user can interact only with the client node that has the interaction enabled, which is usually the middle node (Client 0). Each movement the user makes entails that this client node launches a synchronization procedure that consists of the following steps:

(1) The Client 0 node captures the position and orientation indicated by the user and sends them to the Node.js Server node. These events consist of the camera view, which is the rotation values of each axis, and the actions to move to the next position in the street.

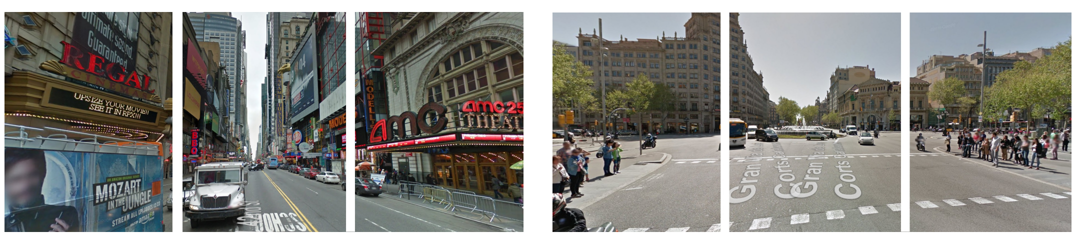
(2) The Node.js Server node receives the events sent by Client 0 and prepares the camera view parameters to be replicated. In the case of a new user position request, the server node requests the Google Street View server in Internet in order for the identifier of the new position.

(3) Google Street View server receives the request and sends back the answer with the new identifier.

(4) The Node.js Server sends the new camera view and the new user position to the clients.

(5) Every node accesses Google Street Server through Internet in order to download the required imagery to be visualized independently from the other nodes.

Experimental Results



(a) New York tour (b) Barcelona tour

Figure 3-3: Images of the two tested environments

In order to study the performance of the Peruse-a-rue, we developed a benchmark with a simple tour for two different cities. The tours were performed in the cities of Barcelona and New York, as can be seen in Figure 3-3, have different image resolution. This difference between both tours is great enough to be studied thoroughly in a future work. The starting point for both tours are one of the main street from each city. In the case of Barcelona it was chosen “405, Avinguda Diagonal”

(41.3968332, 2.159211,17) facing North-East, and for New York it was "371 W 42nd St" (40.7579422, -73.9920296,19) facing South-East. Then, the benchmark moves the visualization point forward, one step at a time, each 5 seconds until it stops after 60 seconds. For this experimentation, the server node was implemented as a background process of the middle client node; so it acted as a server and a client node.

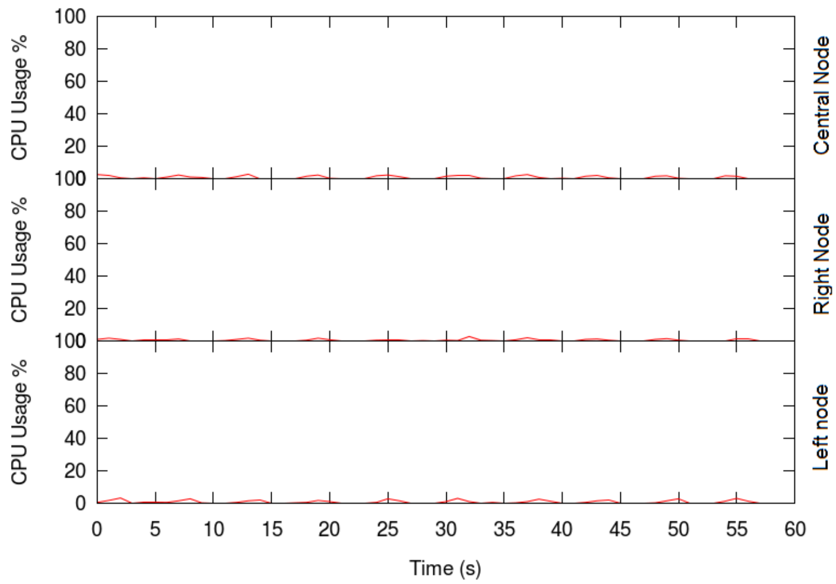


Figure 3-4: Peruse-a-rue - CPU Load for Barcelona tour

Figure 3-4 shows the CPU load when running Peruse-a-rue on Liquid Galaxy for the server and the three clients nodes (central, right and left nodes). Both tours, from Barcelona and New York, depict an almost identical graph, so, it is redundant to show both. For this reason, only the results for Barcelona are depicted. The CPU load monitored during the benchmark in Barcelona is shown in red. As can be seen, there was almost no activity in the CPU. The same low activity was observed for the memory consumption. This was due to the fact that the Peruse-a-rue application runs over a web browser, which was characterized by constant and low CPU and RAM consumption. Accordingly, we determined that this behavior can be extended to other client-server applications that run in Liquid Galaxy.

Figure 3-5 depicts the bandwidth consumption for the tour of Barcelona. In this

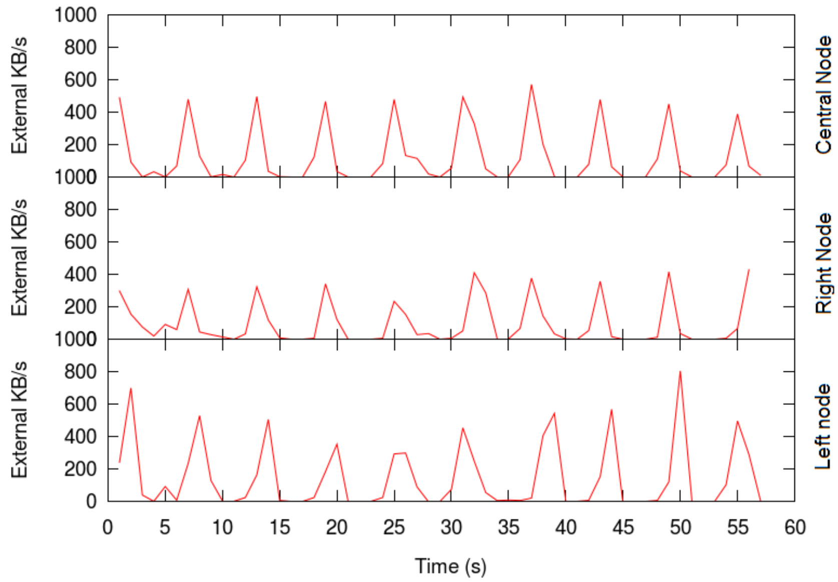


Figure 3-5: Peruse-a-rue - External networking for Barcelona tour

figure, the ten peaks of traffic related to the movements done every 5 seconds in our benchmark can be seen. Note that each peak is due to the loading of several images at once, whereas when there is very low traffic, it means that there are no images to be loaded because there is no movement in the tour. It has to be noted that the images provided by Google are low-resolution to allow a fluid motion of the application, thus, the bandwidth consumption is low in relation to other applications. It must be taken into account that the 10-Mbps broadband connection is not used at its maximum capacity throughout the test, achieving peaks of 400KB/s per node at specific moments of the test.

After studying the performance of this kind of applications, it can be stated that there is no problem from the performance point of view when executing these applications in Liquid Galaxy and, thus, no further study in this field is needed. We have seen that the only potential problem could be the bandwidth, but, as the images being downloaded are in a low resolution, there is enough with a conventional Internet connection.

3.2.2 Gaming Applications

Most of the multiplayer video games on the market use a client-server model, in which the clients are connected to a single server. The server runs a server-side application that manages the information about the events happening in the game. The events can be started from the server itself, like changing weather, or by the clients, like changes in position or performing other actions. These events are processed by the server, which transmits the response to all the connected clients. The clients run a client-side application, which loads all the graphics using the information provided by the server. Therefore, the information transmitted from the server to the clients is mostly events notifications, providing a fast synchronization.

Figure 3-6: Quake III Arena running in Liquid Galaxy

A video game named Quake III Arena was studied in order to analyze the performance issues while running it on Liquid Galaxy. Figure 3-6 depicts this setup. The procedure to provide an immersive experience with Quake III on Liquid Galaxy was to connect all the nodes to the server. Then, the middle node (Client 0) was set to be the actual player while the other nodes were just spectators of that player with a viewpoint offset to the physical position inside the cluster.

Figure 3-7 shows how the synchronization protocol of Quake III Arena works in the Liquid Galaxy. It consists of the following steps:

(1) The user interacts with Client 0 by means of a mouse or keyboard. These events

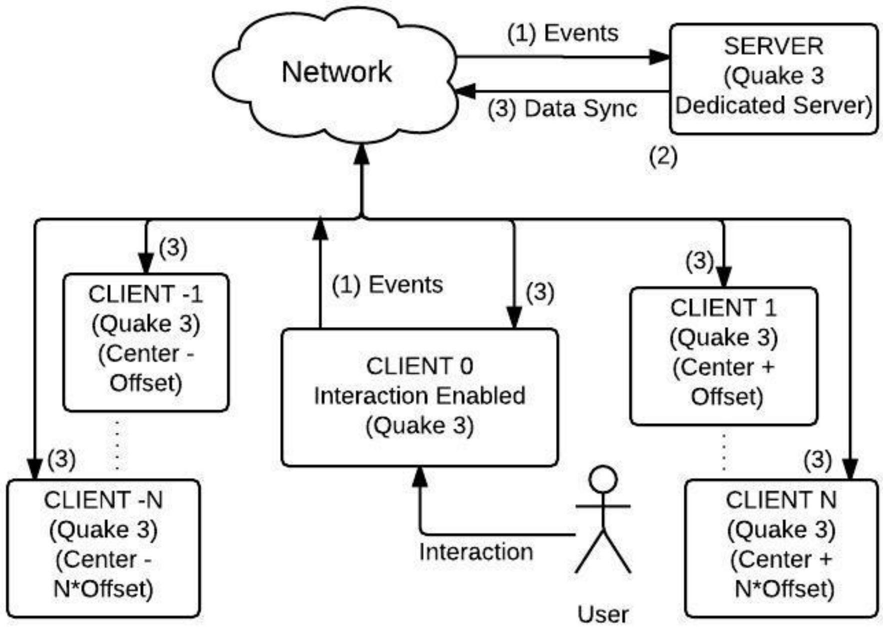


Figure 3-7: Quake schema

are constantly sent to the server node.

(2) The server node receives and processes the events sent by the client that the user is interacting with and changes its status depending on the client and/or the server events. Examples of events can be: the position and target of each player, the players' kills, power-ups, weapons and ammunition spawns, timer, team changes, etc.

(3) The server sends the status changes to all of the clients, and this means the client applications show the modifications.

Experimental Results

In order to study the performance of the Quake III Arena application in Liquid Galaxy, the application was monitored for a short period of time while a user was playing. The initial test had a duration of $60s$ and was performed in the map $Q3DM6$ with 5 players being controlled by the game, also called bots.

The analysis of CPU, memory and network traffic consumption obtained for the Quake III Arena game showed the same results for all the client nodes. Thus, we present the results obtained for the server node and only one of the clients.

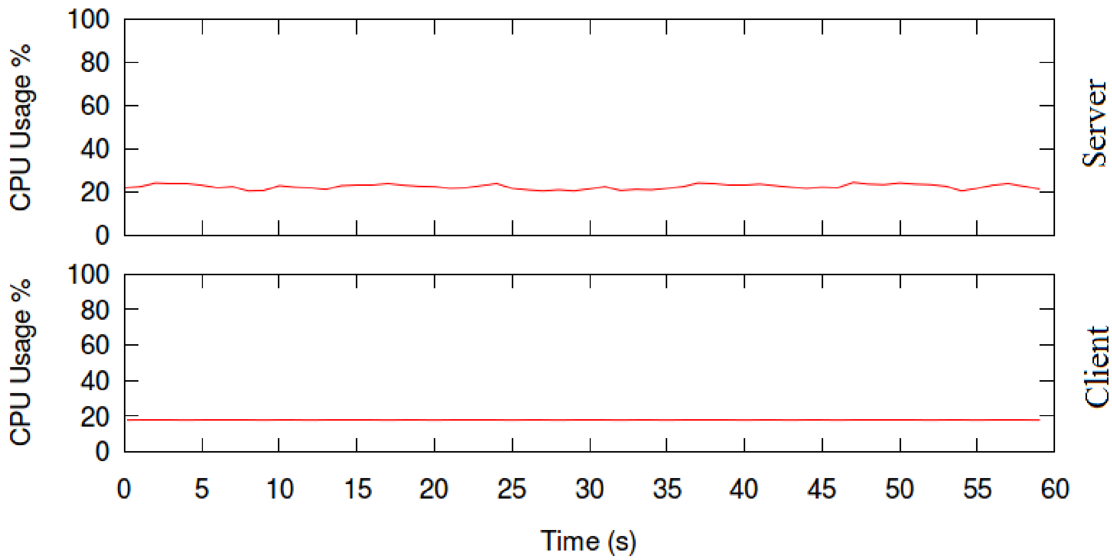


Figure 3-8: Percentage of CPU usage - Quake III

In Figure 3-8, it can be appreciated that there was very low CPU usage in the server and the clients. We can see that the server had slightly higher CPU consumption than the clients, given that it has to broadcast all the events to the clients. This was because there were only 3 clients connected to the server. These results match the the analysis given by Barri et al.[12] which showed that the CPU consumption is related to the number of players in the server. Their works studied how the application only affects the CPU consumption of the server node. This CPU load in the server node increases proportionally to the number of clients due to the events that the server has to broadcast to the other clients. However, given that the CPU only has to process and send the game events, CPU consumption never reaches high rates of usage. That is why the client CPU load is low independently of the number of players. Note that this consumption is always lower than 30%.

As for the memory usage, Figure 3-9 shows that the application loaded all its assets on joining the game and, thus, the memory use remained constant throughout

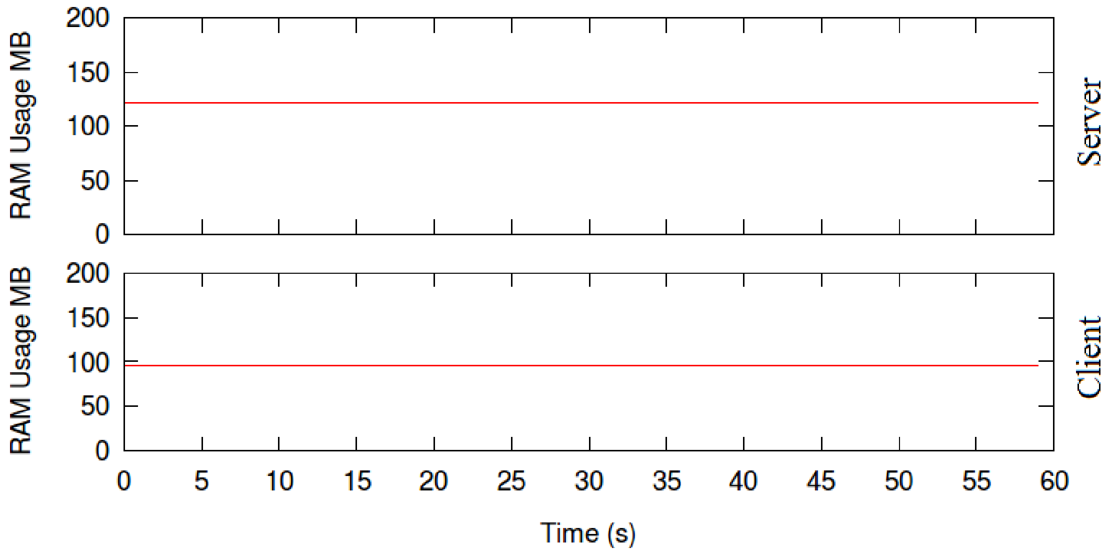


Figure 3-9: Memory usage - Quake.

the test.

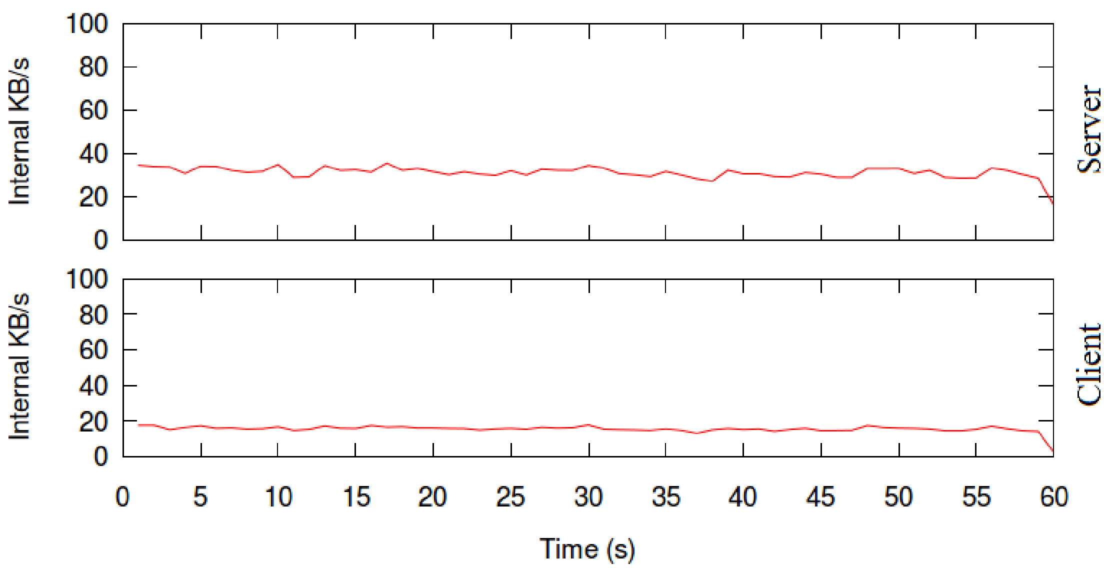


Figure 3-10: Internal Networking - Quake

As for the network traffic, the test was performed with the server inside the same local network, so no external traffic was involved. However, as can be seen in Figure 3-10, the traffic was purely synchronization without high-weight packets being sent. This means that the bandwidth of the connection is not as important as the latency, better known as ping, between server and clients.

Having studied the gaming applications, we concluded that with a system with medium quality hardware there is no sign of any problems with the visualization system and, thus, we decided to cease studying this kind of application from the point of view of performance issues. Likewise, the only problem that could arise is when many users are connected to the server, as the resource consumption in the server would increase proportionately. This scenario is unlikely in a Liquid Galaxy driven environment, as the cluster is designed to be used by a single person and, thus, the number of clients connected to the server is limited by the number of screens used.

3.3 Performance based on the Master-Slave Model

This section describes the performance tests carried out with Google Earth as a representative case of a master-slave application that runs in Liquid Galaxy. Google Earth was originally the main application used in the Liquid Galaxy system. As we explained in Section 2.1.3, most of the applications that use the master-slave model run the same application in all the nodes and use a similar synchronization protocol to Google Earth.

3.3.1 Google Earth

Google Earth was the first application adapted for Liquid Galaxy and, thus, we are very interested in studying its behavior.

The connection schema of the Liquid Galaxy with Google Earth is depicted in Figure 3-11. We can see that each node runs the Google Earth application (GE) and the user only interacts with the master node by means of a 3D mouse. Each movement of the mouse makes the master node launch a synchronization protocol that consists of the following steps, indicated in Figure 3-11:

(1) The master node captures the coordinates (Coords) of the position in Google Earth indicated by the user. These coordinates are codified in an UDP packet,

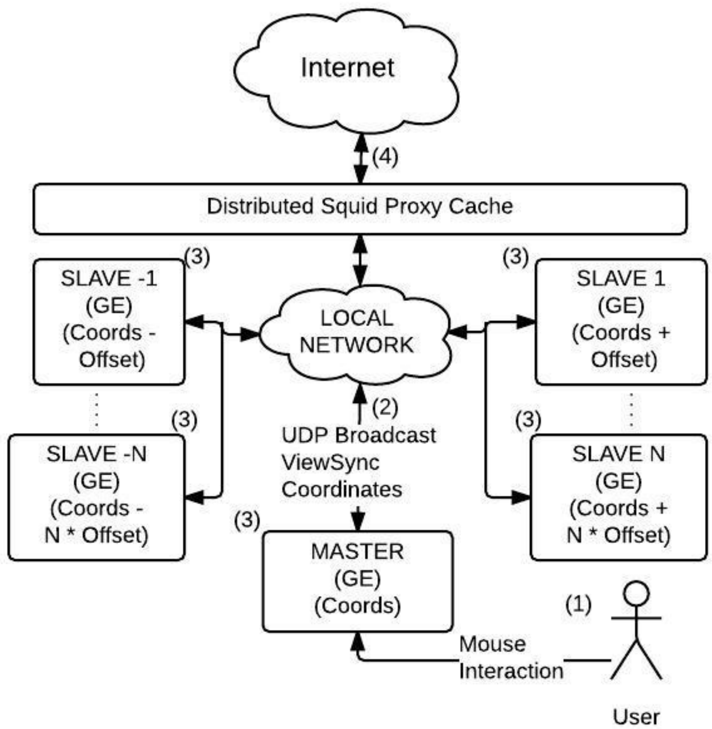


Figure 3-11: Liquid Galaxy architecture running Google Earth

named ViewSync, which contains the following information from the view in the application: counter, latitude, longitude, altitude (or zoom), heading, tilt, roll, time start, time end and planet name.

(2) When the master's view is moved, it sends ViewSync packets to broadcast with the aim of sending them to all the nodes in the network.

(3) Every slave node has a configuration file, which holds an offset from the original master's view. When the slaves receive ViewSync packets from the master, they calculate and adjust their relative view by adding the local offset.

(4) Every node accesses Internet with its own coordinates to download the required data (maps, imagery, 3D layer, etc.) independently from the other nodes.

As we have seen in previous steps, every node executes its own instance of Google Earth, so every node needs an Internet connection to download all the data. For this reason, a web-proxy distributed cache fits very well. Squid [82] is the distributed cache included in the Liquid Galaxy repository for the Google Earth application. All nodes will share the same disk cache stored in each node's SSD. With this solution, the number of data requests from Internet can be reduced. So, if the information is available in the Squid cache, shared by all nodes as peers, it is taken independently by each node.

Experimentation was carried out to measure and categorize the overhead produced by the use of Google Earth in Liquid Galaxy in order to show the resource requirements in normal displacements around the globe.

For the application under study, the requirements for computing resources vary with the environment to be visualized. Thus, in order to have different environments to test, we measured the following three environments, illustrated in Figure 3-12, with significant differences in the number of high-definition 3D buildings to be visualized:

- *City (Barcelona, Spain):* A high-density place made up of 3D buildings.
- *Town (Horsens, Denmark):* A low-to medium-density place that combines places with a few 3D buildings and areas without them.

(a) City tour

(b) Town tour

(c) Desert tour

Figure 3-12: Images of the three tested environments

- *Desert (Sahara, NW Sudan):* A low-resolution area without any 3D buildings to be loaded.

In each environment or tour, the user moves throughout eight specific points of interest with a previously established time interval between consecutive jumps. We distinguish the following two intervals:

- Short time jumps: the time interval was set at 30 seconds.
- Long time jumps: the time interval was set at 60 seconds.

A wide range of time intervals was tested and the minimum interval chosen was 30 seconds, as this is the minimum amount that allows us to differentiate the performance across all the tours in our current system configuration. We must take into account that a lower interval does not allow all the requested data to download fully from the Internet. The interval of 60 seconds allows us to analyze when the system is able to load each image completely.

As every node is connected to the Internet, the Squid proxy cache boosts the performance by providing a way to store and access HTTP objects downloaded from the Internet in a peer-to-peer distributed cache for better bandwidth performance. That is why we carried out the experimentation with the Squid cache turned on, and also off, to evaluate its effect on the performance of the system.

Experimental Results

Both the master and slave nodes were monitored in our experimentation. The results obtained showed that the master and the slave nodes behaved the same way. Thus, only the results for the master node are presented.

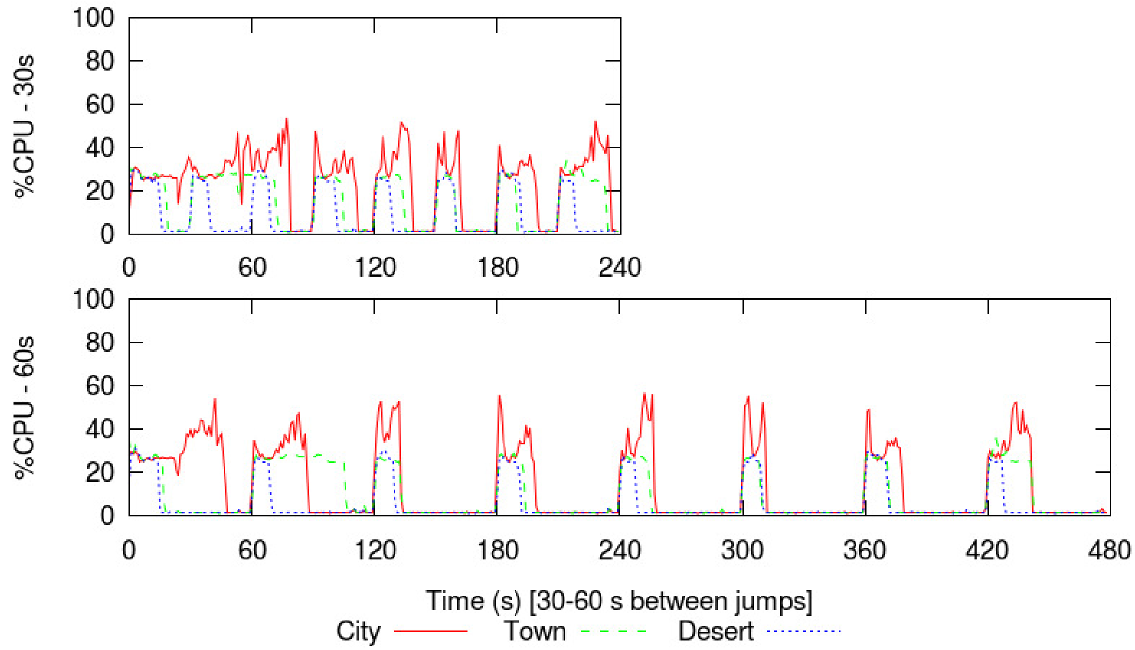


Figure 3-13: Percentage of CPU usage - Squid ON.

Figure 3-13 shows the percentage of CPU usage for the three environments described and both timing jumps. In both cases, a clear difference between the three environments can be seen. In the case of the desert tour, there were many idle intervals because it lacked 3D buildings and had a low-resolution imagery. This made the time interval between jumps long enough to allow the images to load fully. In the other environments, the city and town, similar behavior can be seen in the CPU usage, because both contained 3D buildings. However, due to the higher density of 3D buildings in the city, the system needed more CPU to render all the visual data, giving higher peaks in the CPU use. According to the CPU results, it can be seen that there is no need for the nodes to have a very high-performance CPU, given that the average usage was below 40% in all cases. Nevertheless, the time between jumps is determinant for fully visualizing the images of the environment. We can see that a timing jump of 30s is not long enough to load completely the images in some points (see top graph of Figure 3-13). From here on, the results for the 60-seconds test are presented because they provide better visualization of the differences between the tested environments.

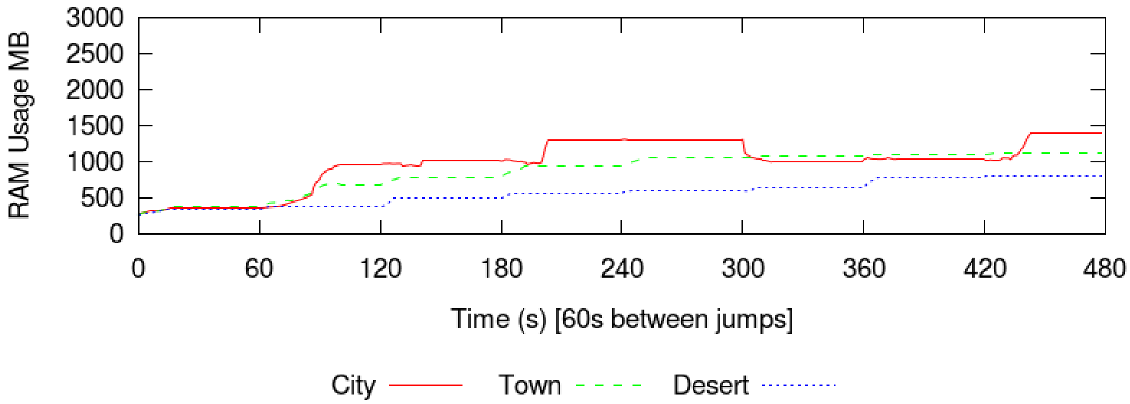


Figure 3-14: Memory usage - Squid ON.

As in the previous analysis, another studied parameter was the memory usage throughout the test with the Squid cache activated. This is shown in Figure 3-14. All tests proved that the RAM always increases moderately regardless of the environment and the timing jump. Memory will always be reserved at the same rate until the limit established by the application is reached (in this case, 1500 MB), at which point the OS will free memory pages. This limit is always less than the total memory available.

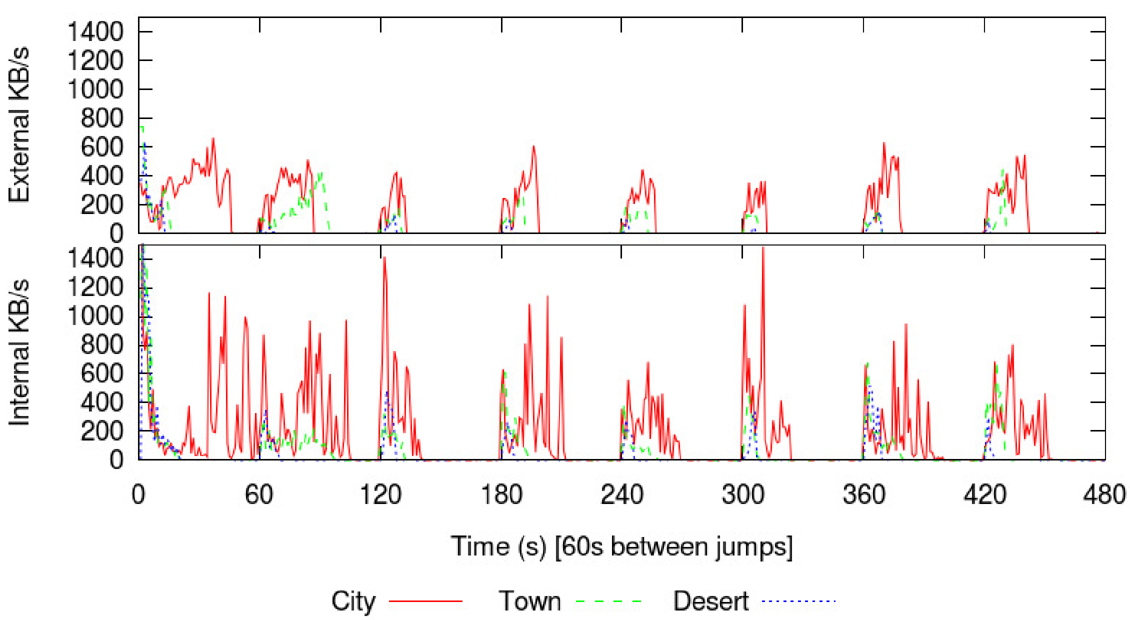


Figure 3-15: External and Internal Networking - Squid ON.

A key performance parameter is the network usage because Google Earth has a heavy request of data from the Internet and many internal synchronization packets per second. Figure 3-15 presents the network graph measured in kilobytes/s. It shows the external traffic (the data downloaded from the Internet) and the Squid internal traffic (that shows how data is passed between nodes). From the results of both graphs in Figure 3-15, it can be observed that, despite the Squid cache being activated, we can see that data was still downloaded externally whenever a request was made to change position. This happened because Squid only had an average of 20% cache hits, leading the nodes to download new data when the next place to visit had not been cached. We can also note that the desert tour had a lower data request because it had no 3D buildings and only low-resolution maps to download. There was also a common starting peak in both graphs and all the tours. This initial peak appeared because the tour started with a general view of the Earth before moving to the first coordinate and, thus, all the data had to be loaded until the first position was reached.

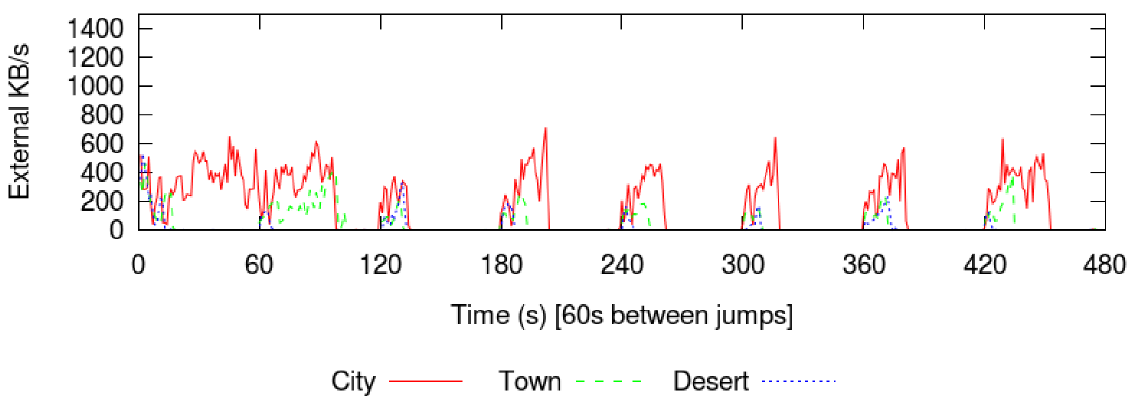


Figure 3-16: External Networking - Squid Off.

To help us understand how much the proxy cache Squid influences the system, we performed the same tests with Squid disabled. Figure 3-16 shows the external networking when Squid was disabled. If we compare it with the external traffic when Squid was enabled (Figure 3-15 top), at first glance there seems to be no difference, although the download time is seen to be somewhat longer. This is very important, because it tells us that when Squid was enabled, it shortened the download time from the Internet, which is where we detected a bottleneck.

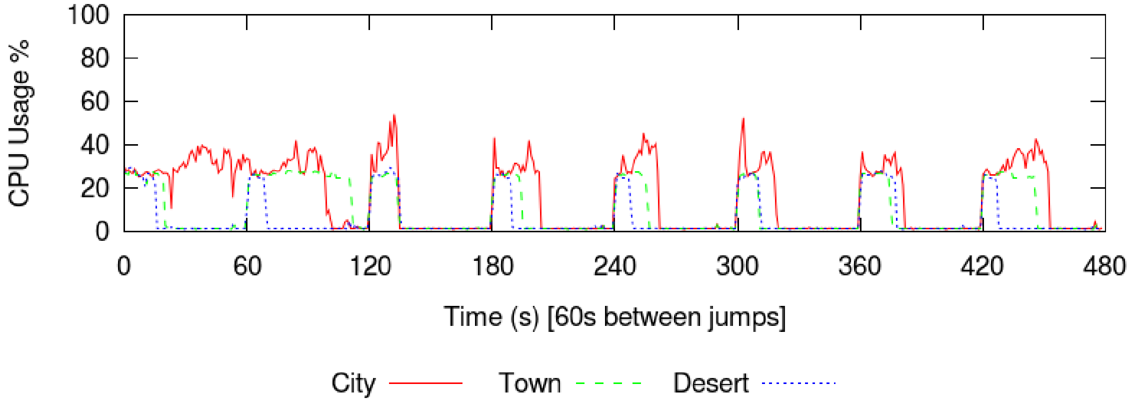


Figure 3-17: Percentage of CPU usage - Squid Off.

Figure 3-17 shows the CPU usage when Squid was disabled. If we compare this with the same case with Squid enabled (Figure 3-13 bottom), we can see that the CPU was active longer than when Squid was enabled, but with a lower load. This was because the required data had to be downloaded from the Internet without the help of Squid, as we saw in the networking graph. This means that the data downloaded slower and the CPU was rendering all the data until the download had been completed, meaning lower but longer CPU usage.

The behavior of memory performance was the same when Squid was enabled as when it was off, so there is no need to present that graph.

The performance results obtained in this section reveal that the CPU, memory and local network were powerful enough to visualize images with different densities of 3D buildings; while the external network bandwidth was the real bottleneck of the system.

3.4 Additional Tools Developed for the Liquid Galaxy Project

As shown previously, the Liquid Galaxy infrastructure was initially conceived for the purpose of running Google Earth. However, it is a versatile platform that, in fact, constitutes a general-purpose cluster display wall. According to this, companies and institutions like EndPoint Corp [29] in New York (USA) and the UWS Wonderama Lab [96] in Penrith (Australia) have developed modifications of some existing applications [20] to adapt them to Liquid Galaxy. Moreover, Google has been giving students around the world more possibilities of contributing to the Liquid Galaxy project with their own projects with the Google Summer of Code (GSoC) [34]. GSoC is an annual international program in which students are given a grant by Google to work on their projects during the summer and develop functional projects.

Different applications for the Liquid Galaxy are being developed every year as part of the Google Summer of Code projects. Specifically, some of these works were part of this thesis and these are summarized next.

- The Liquid Galaxy Benchmarking project [5] carried out in 2013, was aimed at providing a tool to benchmark and monitor the performance of the Google Earth application to help diagnose possible bottlenecks. This tool was used to analyze the performance of Liquid Galaxy while running Google Earth and Quake III arena, both presented in this thesis.

- The Liquid Galaxy Web Benchmarking project [6], done in 2014, was a similar tool whose focus was the Peruse-a-Rue, a Street View version for Liquid Galaxy, explained in Section 3.2.1. This project provided a tool for benchmarking and monitoring the performance of such client-server applications and can be easily adapted to other types of web-based applications.

- In the summer of 2016, a cloud point visualization tool for Liquid Galaxy was developed[7]. This project allowed different types of cloud point files in a web-based application in Liquid Galaxy to be displayed. Such files can be obtained by using LIDAR, project Tango, or 3D modeling applications. It will enable Liquid Galaxy to use other kinds of applications, such as BIM (Building Information Modeling) applications or precision agriculture.

All of these projects have a user-friendly web interface, are open-source and available online.

Chapter 4

Google Earth Performance

With the evaluation of the performance based on the elements of system configuration discussed in the previous chapter, we were able to verify that the Liquid Galaxy cluster display wall, built with commodity hardware, has enough computing capacity to run and visualize any kind of application. Moreover, our previous results reveal that Google Earth is the most sensitive application to the performance issues with different kinds of Liquid Galaxy configurations.

According to this, in this chapter we propose a new metric defined to measure the performance of the system in relation to different configurations. Taking this metric into account, Liquid Galaxy is benchmarked running the Google Earth application in different environments to test the scalability and heterogeneity issues of the system.

4.1 Visualization Rate (VR)

A new metric that gives information about the performance of the Liquid Galaxy was defined taking into account that the CPU usage tells us when the system has loaded all the visual elements. With this information, we defined the Visualization Rate (VR) as the average CPU idle time for a cluster of n nodes with the following equation:

$$VR = \frac{100}{n} \sum_{i=1}^{n} \frac{T\_idle_i}{T\_total}, \tag{4.1}$$

where T_total is the total time of the test and T_idle_i is the time when the CPU load of node n_i is below a minimum threshold.

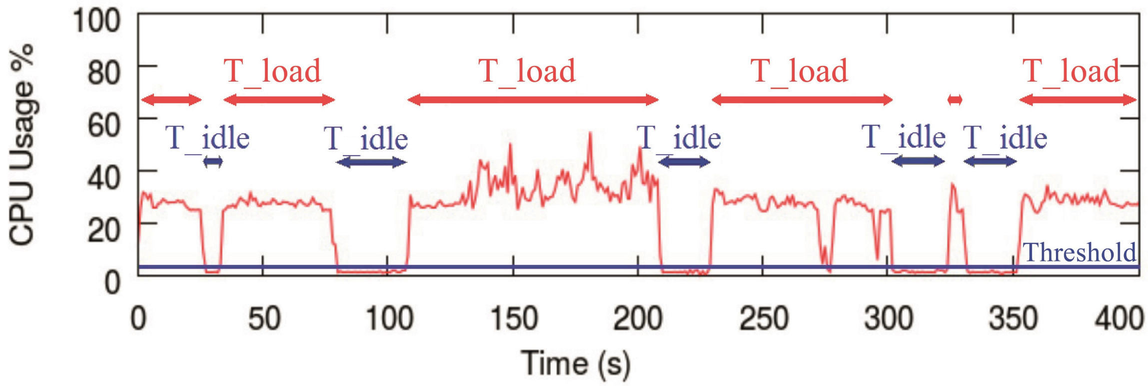


Figure 4-1: VR: CPU Usage.

Note that a CPU load below this threshold means that the CPU is idle and the images have been fully loaded. This procedure is illustrated in Figure 4-1, where the blue line near 5% CPU Usage is the threshold chosen for this particular case. This threshold was calculated using the values of the CPU when the application was in an idle state, which means that the application was using the minimum resources from the system. Thus, any value above that threshold would count as workload. The peaks depicted in this Figure correspond to when the CPU was processing the imagery and the values that drop below the threshold represent when the CPU became idle. The CPU usage information was gathered every second from the information given automatically by the system monitor. Note that when the VR is near 100%, it denotes a high visualization rate and, so, presumably a good user perception as images are fully visualized. On the other hand, having a VR equal to 0% means that the data has not been fully loaded and, thus, it has been ineffective. Any value between 0% and 100% indicates how efficiently the system has performed.

Taking into account the commented before, it can be seen that VR is a metric that not only gives information about performance, but also could be used to relate this performance with user perception.

4.2 Benchmarking Tests

Having defined our new metric, we performed tests to evaluate the VR metric for both homogeneous and heterogeneous cluster configurations using the automated scripts described in Section 3.3.1 to simulate user behavior. This gives an understanding of the performance of the Liquid Galaxy system in relation to several scenarios with different densities of 3D buildings and patterns of users.

4.2.1 Homogeneous System

Table 4.1: VR obtained from Benchmarking Tests

	Short Timing Jump (30s)			Long Timing Jump (60s)		
	City	Town	Desert	City	Town	Desert
Squid ON	15 %	27 %	57 %	48 %	56 %	78 %
Squid OFF	11 %	24 %	45 %	39 %	50 %	72 %

The homogeneous cluster used in this experimentation was made up of a variable number of nodes, from 3 to 8, where each node was configured as described in Section 3.1.

Table 4.1 shows the impact that the type of environment, the timing jump and proxy cache have on the VR metric when we used our benchmarking tours (City of Barcelona, Town of Horsens, Desert of Sahara) in a cluster of 3 nodes.

It can be seen that there was a noticeable difference between each tour, which tells us that the type of environment is very important because the more 3D buildings that have to be rendered, the lower the VR is. Additionally, it can be observed that the short timing jump gave a lower VR value, so the user had less time to visualize the fully loaded images than in other cases.

Another important parameter to be considered is whether Squid is enabled or disabled, because the Squid ON reduces the external downloaded traffic and images can be visualized faster, thus increasing the VR value. This improvement due to the Squid ON is clearly visible in Table 4.1.

Table 4.2: VR related to scalability

	Short Timing Jump (30s)			Long Timing Jump (60s)		
Node	City	Town	Desert	City	Town	Desert
3	15 %	27 %	57 %	48 %	56 %	78 %
5	5 %	22 %	50 %	31 %	52 %	77 %
8	2 %	14 %	46 %	14 %	36 %	68 %

In order to analyze how performance parameters behave for bigger platforms, we analyzed scalability in terms of VR, when the number of nodes increases from 3 to 8 nodes.

Table 4.2 shows the results obtained in the scalability study with Squid ON. As was expected, we can see how Short Timing Jumps achieved a very low VR value for the City and Town Tours compared to the Long Timing Jumps. In general, we can see that the overall VR decreases when the number of nodes increases. This happens because the broadband connection is a fixed resource and the bandwidth must be divided between the nodes. Thus, the incoming input of data to be processed by each node per unit of time is diminished and, as a consequence, the rendering time is lengthened. So, the suitable number of nodes is determined by the type of Internet connection.

It is also worth pointing out that when the number of nodes is scaled, not all the nodes load the Internet data at the same time. Due to the networking protocols of the router, there is a slight delay of 3 seconds between the central and side nodes of the cluster. This delay can influence the VR value but not the real user perception, given that the user focuses on the central node and does not give much importance to the side nodes. This behavior is be explained in section 5.3.4. As explained previously, the VR is averaged from all the nodes in the cluster without taking their position into account. With this fact in mind, we can say that the user can have a better perception of the multimedia application using more displays with the same VR value.

After studying the scalability of this specific cluster display wall, we can say that the bandwidth of the Internet connection is a bottleneck, as it limits the speed of rendering.

4.2.2 Heterogeneous System

In order to extend the usage of a cluster display wall to a wider set of people, we are interested in building a low-cost heterogeneous system. In this context, a real usage cluster would be built with different kinds of nodes, assembling a heterogeneous cluster. According to this, we specified different configurations of Liquid Galaxy setups on which the different tests were carried out. Each configuration was made up of n commodity PCs chosen from the list of nodes given in Table 4.3. For each kind of node used in this experimentation, Table 4.3 shows its identifier N_i, hardware characteristics, benchmarking time T_i in seconds and relative power W_i. It is worth pointing out that the hardware specifications of each node were above the minimum required by Google to run Google Earth.

The benchmarking time T_i was calculated by running two different benchmarks in each node N_i as follows,

$$T_i = 0.80 * T_{FPU_i} + 0.20 * T_{Integer_i} \tag{4.2}$$

where T_{FPU_i} and $T_{Integer_i}$ were the times obtained in the *FPU Raytracing* and *CPU N-Queens* benchmarks, respectively, from the *hardinfo tools* [87]. The weight assigned to each benchmark was obtained by monitoring the Floating Point and Integer instructions when a standard user was navigating with Google Earth running in the Liquid Galaxy cluster.

The relative power W_i shows the power of the node N_i in relation to the fastest node in Table 4.3, which was N_4. This was calculated by means of the following relation:

$$W_i = \frac{T_{min}}{T_i}, \tag{4.3}$$

where T_i was the benchmarking time of node N_i and T_{min}, the lowest benchmarking time from all the nodes shown in Table 4.3 (11.5 seconds in this case).

Table 4.4 shows the different Liquid Galaxy configurations used throughout this experimentation sorted by their Heterogeneous Degree. The following information is shown for each configuration:

- n: number of nodes in the cluster

- Het_Degree: Heterogeneous Degree parameter that shows how different the nodes are in any specific configuration. This is calculated as follows:

$$Het\_Degree = \frac{\sum_{i=1}^{n-1} \frac{W_{max}-W_i}{W_{max}}}{n-1}, \tag{4.4}$$

  where n is the number of nodes in the cluster, W_i is the relative power of node N_i and W_{max} is the relative power of the fastest node in the cluster. Note that a $Het\_Degree = 0$ means that the cluster is homogeneous and a value of 1 means that the cluster is completely heterogeneous. In practice, the Het_Degree of the compositions that we built only ranged between 0 and 0.43 because the upper bound was limited by two different aspects: the slowest nodes, which were limited by the minimum specifications given by Google, and the fastest nodes, which were limited by the fact that we only wanted to use commodity hardware.

- Node composition: the number of nodes of each type, shown in Table 4.3, that make up the platform.

All the configurations had a local network of 100 Mbps and a standard 10 Mbps broadband Internet connection. It is worth pointing out that we always maintained a type N_1 node as the master node in all the configurations. Most of these compositions are built with three nodes because it is the minimum configuration recommended by Google. The most recommended configurations for clusters are with $n = 3$, $n = 5$ and $n = 8$.

Given our purpose of analyzing the performance of the new metric in relation to different values of Het_Degree, we first evaluated the VR over the six specific Liquid

Table 4.3: Kinds of node

Node (N_i)	Characteristics	T_i(s)	W_i
N_1	i5 3330 @ 3GHz (4 cores), nVidia GeForce GT620, 8GB RAM 1600MHz, SSD	20.15	0.57
N_2	i5 3330 @ 2.4GHz (4 cores), nVidia GeForce GT620, 8GB RAM 1600MHz, SSD	25.24	0.46
N_3	i5 3330 @ 2GHz (4 cores), nVidia GeForce GT620, 8GB RAM 1600MHz, SSD	30.75	0.37
N_4	Intel Xeon @ 3.4GHz (4 cores), nVidia GeForce GT420, 6GB RAM 1333MHz, SSD	11.5	1
N_5	i5 3470 @ 3.2GHz (4 cores), MESA DRI Intel IvyBridge, 4GB RAM 1600MHz, HDD	17.58	0.65
N_6	AMD Athlon 64 3500+ (1 core), ATI RS480, 1GB RAM 400MHz, HDD	36.22	0.32
N_7	Core2 Duo @ 3.00GHz (2 cores), Intel 4 Series Graphics, 4GB RAM 1300MHz, HDD	21.13	0.54

Table 4.4: Liquid Galaxy composition

n	Het_Degree	Node composition
3	0.43	2xN_1 + 1xN_4
3	0.22	2xN_1 + 1xN_6
8	0.19	2xN_1 + 1xN_2 + 1xN_3 + 2xN_5 + 2xN_7
3	0.18	2xN_1 + 1xN_3
5	0.17	2xN_1 + 1xN_3 + 1xN_5 + 1xN_7
3	0.12	2xN_1 + 1xN_5
3	0.1	2xN_1 + 1xN_2
3	0	3xN_1

Table 4.5: VR related to heterogeneous clusters for $n = 3$

Short Timing Jump (30s)					
Het_Degree	W_{avg}	$W_{slowest}$	City	Town	Desert
0.43	0.71	0.57	14 %	36 %	59 %
0.22	0.49	0.32	4 %	13 %	49 %
0.18	0.50	0.37	8 %	26 %	55 %
0.12	0.60	0.57	14 %	25 %	58 %
0.1	0.53	0.46	11 %	25 %	54 %
0	0.57	0.57	15 %	27 %	57 %
Long Timing Jump (60s)					
Het_Degree	W_{avg}	$W_{slowest}$	City	Town	Desert
0.43	0.71	0.57	51 %	57 %	78 %
0.22	0.49	0.32	14 %	24 %	63 %
0.18	0.50	0.37	43 %	54 %	77 %
0.12	0.60	0.57	50 %	56 %	78 %
0.1	0.53	0.46	46 %	53 %	77 %
0	0.57	0.57	48 %	56 %	78 %

Galaxy compositions given in Table 4.4 that had three nodes. Each composition is identified by its corresponding Het_Degree (0.43, 0.22, 0.18, 0.12, 0.1 and 0).

Table 4.5 shows the VR obtained for each composition in the three tours tested and the Short and Long Timing Jumps. Together with the Het_Degree value, the W_{avg}, which is the average power (W_i) of the three nodes forming the cluster, and the $W_{slowest}$, which corresponds to the minimum W_i power among them, are shown.

In general, the results in Table 4.5 show a significant difference between Short and Long Timing Jumps, as already seen in the homogeneous test. Likewise, there are more differences between clusters in the Short Timing Jumps test. The reason of this behavior is that all the configurations for the Long Timing Jumps, apart from the $Het\_Degree = 0.22$ case, had enough time to download the data from the Internet and, as a consequence, the CPU advantage of some clusters was not highlighted. Thus, we can see that the Timing Jump of 60s was too long to differentiate the performance between clusters. The exception to this behavior was the case of the $Het\_Degree = 0.22$, which obtained very low values for the City and Town tours. This poor performance reveals that a low $W_{slowest}$, such as in this case, can lead the cluster's VR to very low levels. This shows that the performance of clusters with similar W_{avg}, such as the case of $Het\_Degree = 0.22$ and 0.18, can be significantly different because one of the nodes is very slow.

Focusing on the Short Timing Jumps tests, we can see that the differences between clusters were minimal in the Desert tour. In the Town tour, there were greater differences given the increase in the number of rendering operations and the consequent increase in the influence of W_{avg}. Likewise, we can point out that the fastest cluster ($W_{avg} = 0.71$) obtained the best VR for the Town tour, but not in the City tour. This was due to the broadband connection for the tests, this being a common resource of $10Mbps$, which did not allow the CPU speed advantage of the best clusters to be exploited when the amount of data to download was extremely high. Note that for the City tour, the homogeneous configuration ($Het\_Degree = 0$) gave slightly better performance, although this difference was not significant. Apart from the homogeneous case, it is worth emphasizing that the cluster with the highest heterogeneous degree ($Het\_Degree = 0.43$) obtained the best performance, this being the configuration with the highest $W_{slowest}$ and W_{avg} parameters.

In general, we can state that W_{avg} and $W_{Slowest}$ are the parameters with the greatest impact on the VR value, whereas the Het_Degree does not have as much significance as we might imagine. Thus, and taking our results into account, it could be established that heterogeneity is not a problem when building this kind of infrastructure while the Timing Jump is adjusted to the broadband connection and hardware characteristics of the available nodes.

Table 4.6: VR related to the Scalability

Short Timing Jump (30s)						
Node	Het_Degree	W_{avg}	$W_{slowest}$	City	Town	Desert
3	0.18	0.56	0.37	8 %	26 %	55 %
5	0.17	0.52	0.37	2 %	16 %	56 %
8	0.19	0.53	0.37	2 %	8 %	46 %
Long Timing Jump (60s)						
Node	Het_Degree	W_{avg}	$W_{slowest}$	City	Town	Desert
3	0.18	0.56	0.37	43 %	54 %	77 %
5	0.17	0.52	0.37	29 %	49 %	78 %
8	0.19	0.53	0.37	16 %	39 %	69 %

In order to analyze the scalability of a heterogeneous Liquid Galaxy platform, the number of nodes was increased from 3 up to 8 nodes. Likewise, with the aim of isolating the influence of size in relation to the others parameters, Het_Degree, W_{avg} and the $W_{slowest}$, were maintained close to constant throughout this experimentation.

Table 4.6 shows the results obtained. In general, we can see that, as expected, the overall VR decreases when the number of nodes increases. We can see very similar behavior as in the homogeneous case described in Section 4.2.1 with little differences. The results of this scalability study again show that heterogeneity is not a problem when building a cluster with different types of node. We can point out that the

Internet connection still limits the speed of rendering, thus becoming a bottleneck.

4.3 VR Related to User Behavior

As explained in the previous section, we performed a series of benchmarking tests in order to analyze the performance of Liquid Galaxy. The results showed that the type of environment affects the VR results. With these results, by comparing the VR obtained in the tests, it could be compared whether the use of these environments would have a positive or negative impact on performance. However, after studying the performance of Liquid Galaxy by means of the VR, which can only be compared to another VR value, the VR threshold above which a user would be satisfied had to be defined. Therefore, we could tell, by looking at the VR values, if Liquid Galaxy would have acceptable performance values from the user's point of view. In the next chapter, the usability of Liquid Galaxy is studied in order to relate the user behavior to the system performance and, thus, to be able to discern whether a VR value is acceptable or not.

Chapter 5

Usability Analysis of the Liquid Galaxy Platform

According to the performance results from previous chapters, we need to study Liquid Galaxy to know how users react to the system and how to try to predict an estimated overall satisfaction by relating it to the performance metrics. In order to achieve this, the Liquid Galaxy system was tested with real users by analyzing their user experience.

User eXperience (UX) is a widely-used term that deals with the person's perceptions of, and responses to, use and/or anticipated use of a product, system or service [54]. UX is a complex term mainly studied in the Human-Computer Interaction (HCI) and Human Factors fields as a consequence of a user's internal state (predispositions, expectations, needs, motivation, mood, etc.), the characteristics of the designed system (e.g. complexity, purpose, usability, functionality, etc.) and the context (or the environment) within which the interaction occurs (e.g. organizational/social setting, meaningfulness of the activity, willingness of use, etc. [43]). Therefore, as related researchers assert, evaluating UX is no easy task and frequently "traditional" usability (the extent to which a product can be used by specified users to achieve specific goals with effectiveness, efficiency and satisfaction in a specific context of use [54]) evaluation techniques are used to measure UX. The focus of the present work is not the evaluation of the whole UX, but is centered on knowing some aspects of the UX usability attributes of our system: satisfaction, effectiveness and efficiency [53].

For this purpose, we carried out user tests, in three different scenarios, in which the users had to answer some post-task questions about how they felt after completing each task and a post-test questionnaire about their feelings while using the system. At the same time, the system performance was monitored throughout the tests to study part of the user experience when using this kind of application in Liquid Galaxy.

5.1 Objectives of the Usability Analysis

According to [3] and [53], one way of analyzing the usability of a system is through the following three parameters: satisfaction, effectiveness and efficiency. In our context, we understand that satisfaction is a subjective parameter that measures the user's perception, effectiveness as the ability of the system to load all the images while running the application, and efficiency as the time required to load these images.

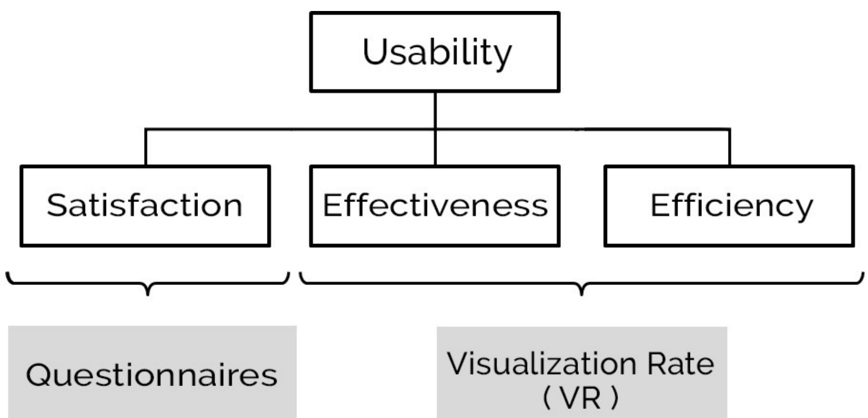


Figure 5-1: Measuring usability

The main objective of our analysis is depicted in Figure 5-1, where it can be seen that effectiveness and efficiency can be measured using the VR performance parameter, described in the previous chapter, and satisfaction can obtained by using questionnaires users had to answer.

Three different tests with different levels of freedom were carried out in order to achieve the proposed objective.

- A driven test that aimed to study the difference between real users and a benchmark. In order to do this, we analyzed how much time a real user needs to see a specific image.

- A laboratory test where the users had to complete a series of guided tasks given by a facilitator and answer a questionnaire about their experience.

- Finally, the last test was done in a travel agency to encourage users to navigate freely using the system to wherever they wanted to and answer some questions about it.

These tests and the results obtained are described in the next sections.

5.2 Driven test

This test was designed to resolve some issues with the benchmarking tests. In the previous chapter, the behavior of the VR when the system was working with the Google Earth in different environments was studied, but we needed to contrast the VR values with those that could be obtained from real users' tests. Thus, this enables us to decide what VR value could be defined as an acceptable minimum. In order to obtain the minimum acceptable VR value, a discount usability test [69] was carried out with 86 volunteers. They were visitors at the *fair Euskal Encounter fair (https://www.euskal.org/)*, where a homogeneous Liquid Galaxy was exhibited. The following subsections describe the environment, the participants and the results that correspond to this test.

5.2.1 Environment

Cluster Display-Wall

The environment where the tests were performed was a stand at the entrance of the main event of the fair. The stand was equipped with a Liquid Galaxy system made up of 8 N_1 nodes (see Table 4.3) with a 40" monitor for each node set vertically in a semi-hemispheric way. The nodes were connected to a gigabit local network with a 10-Mbps broadband connection and the Squid cache enabled.

Figure 5-2: 3D Space Navigator Movement Options

The interaction controller used in all of the user tests was a 3D Space Navigator (Figure 5-2), which is the most widely used controller in all Liquid Galaxy setups. This device is able to displace the view but also rotate it on all 3 axes, making it the most suitable controller for navigating through 3D scenarios.

Facilitator

A facilitator was running the test and guiding the participants through it. The facilitator attended the participants, explained the goals of the test, the consent form and everything to make them feel comfortable. Once the test started, the facilitator could only answer specific questions or give subtle advice when the participant was struggling to complete a task.

5.2.2 Participants

Beforehand, users were asked about their Multimedia eXperience Level (MXL) with an orientative value from 1 to 5 (1 being the lowest value and 5 the highest) that indicates the users' familiarity with multimedia applications. Table 5.1 shows the number of people involved in the test classified according to their MXL and age.

As could be expected, Table 5.1 shows that the lower the age of the participants in the test, the greater their familiarity with using multimedia applications. In general, we can see that the majority of the testers had a MXL in the (3,4) range, which

Table 5.1: Number of people in relation to age ranges and MXL

	MXL				
Age Range	1	2	3	4	5
12-16	-	-	2	5	3
17-24	-	-	4	7	7
25-35	-	2	7	8	4
36-50	1	7	10	5	2
51-75	5	3	3	1	-
Total	6	12	26	26	16

is logical if we take into account that the encounter was oriented towards computer enthusiasts.

5.2.3 Test

The test carried out, shown in Figure 5-3, was made up of six locations on a guided tour of the city of New York, where the user had to press a key when he/she wanted to go on to the next point on the tour. They were not allowed to change the camera position, angle or zoom. The reason for choosing New York is that we wanted to draw the user's attention to the city as a famous place with very important points of interest that any user would want to see. We asked the volunteers to take the tour with no time limit, spending as long as they liked looking at the buildings. In addition, we told them to only go to the next location when they were bored with the current one. This method would tell us how long a person usually spent on each point of the tour in a real environment, giving us real VR values to match our performance metrics. In this way, we were searching for the minimum VR threshold where the users had high satisfaction with the system.

5.2.4 Satisfaction Results

Note that while using discount usability testing [60], we cannot answer such questions as, "How usable is this system?" or "What is the User Experience level of the sys-

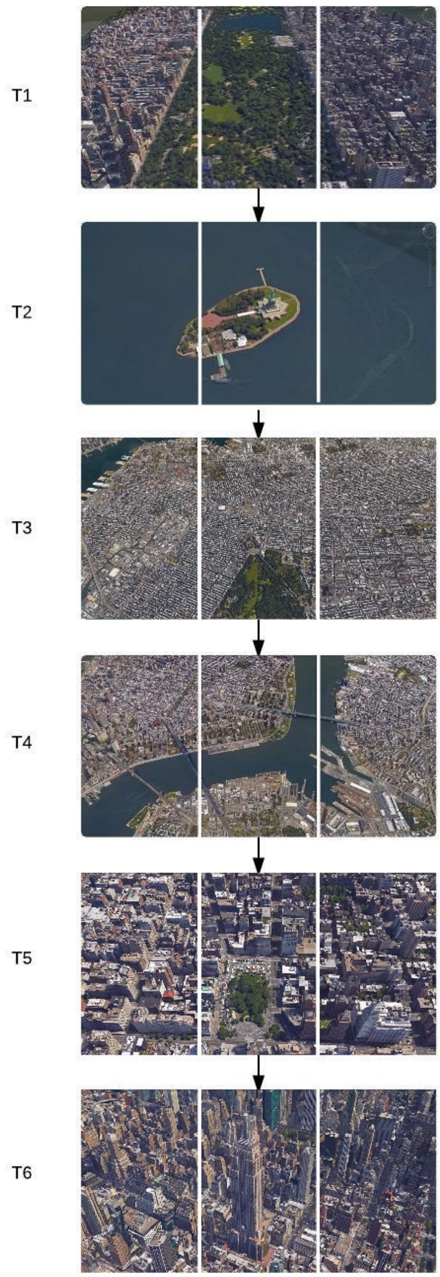


Figure 5-3: Tour Flowchart - T1 Central Park - T2 Liberty Island - T3 Brooklyn - T4 Brooklyn Bridges - T5 Union Square - T6 Empire State. Each picture is the composition of the 3 screens of the system

tem?". This was a preliminary test to give an overview of the VR behavior in relation to real users and to lead the study into new objectives. These new objectives related to satisfaction is explained in the next tests of this chapter. This test provides a quick way of observing people using our system which provided enough observations for the purposes of the test.

5.2.5 Effectiveness and Efficiency Results

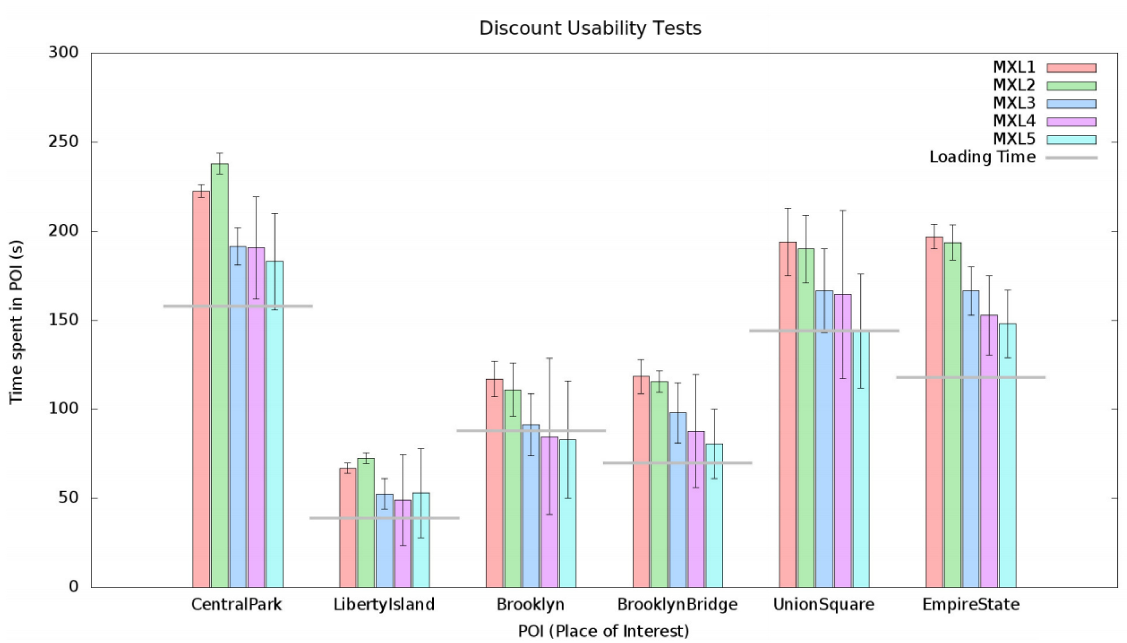


Figure 5-4: Discount usability test results

Figure 5-4 shows the results for the test with users, grouped by their MXL category, with the mean and deviation of the time spent by every MXL group at each of the six points on the New York tour in Figure 5-3. As a guideline, it also shows a horizontal line marking the time needed to load each of the points on the tour. Given that the city of New York has a higher density of 3D buildings than the three benchmarking tours described in Chapter 4 (Barcelona, Horsens and Sahara), the loading time of New York was much longer.

5.2.6 Discussion

According to the values obtained, users can be grouped into two sets:

- Group 1 (MXL1-2): This includes people with low technological skills but with a high attraction to this. They are impressed by the technology or are more interested in the tour. So, they spend longer looking at each location.

- Group 2 (MXL3-5): This includes young people but also those with high multimedia experience or knowledge of multimedia applications. In general, they do not wait long enough to visualize the fully-loaded data because they are too impatient or non-motivated. It is worth pointing out that people with higher MXL do not usually expect to be amazed by this kind of application. Therefore, their interest in the multimedia environment is low and they give lower visualization times. This statement also covers young people as their MXL tends to be very high or they are too impatient.

Both groups follow a homogeneous pattern of visualization in the sense that if one group tended to spend less time than another at a given point in the tour, this trend was continued at all the points. In general, we noticed that the users were more interested in the points of "Central Park", "Liberty Island", "Brooklyn Bridge" and "Empire State", as they spent more time visualizing them compared with their loading time. This confirms that if someone is interested in a place, he/she will spend more time visualizing this. Likewise, the graph also shows that some people did not need all the data to be downloaded in order to visualize it comfortably.

Regarding the deviations, there were greater deviations in the MXL4 and MXL5 groups. This was because many of the users from those groups were younger and behaved unpredictably, such as skipping through the tour fast.

From the visualization times shown in the Figure 5-4, we calculated the related values of the VR parameter. Table 5.2 shows the average VR parameter for every MXL at each point on the tour. The values that are below the average of the values in the table, which is 34%, are marked in red. From Table 5.2, we can see that high VR values were achieved in the points with higher interest on the tour (Central

Table 5.2: VR for every MXL category

	MXL1	MXL2	MXL3	MXL4	MXL5	AVG
CentralPark	41 %	51 %	21 %	21 %	16 %	30 %
LibertyIsland	72 %	86 %	35 %	26 %	36 %	51 %
Brooklyn	33 %	26 %	4 %	0 %	0 %	13 %
BrooklynBridge	69 %	65 %	40 %	25 %	15 %	43 %
UnionSquare	35 %	32 %	16 %	14 %	0 %	19 %
EmpireState	67 %	64 %	41 %	29 %	25 %	45 %

Park, Liberty Island, Brooklyn Bridge and Empire State) and for the groups of people with low MXL (MXL1 and MXL2). Likewise, it is worth pointing out a particular behavior for this specific tour in New York. On one hand, given the high density of 3D buildings in New York, the points chosen need a long time to load fully. On the other hand, they let the users imagine their content from a blurry and incomplete imagery while it is still loading. Additionally, the values achieved for some of the locations are relatively higher than for other locations due to the camera zoom and the number of buildings to be downloaded and displayed.

Table 5.3: Learning times according to MXL level

	Initial	Interm.	Start	Finish
MXL5	12s	10s	80%	100%
MXL4	15s	15s	70%	90%
MXL3	20s	20s	50%	80%
MXL2	30s	30s	20%	50%
MXL1	30s	40s	10%	40%

Table 5.3 shows the different times the users spent learning to use Liquid Galaxy autonomously. Assuming that having a 100% score means that the user is highly skilled, and having a 0% means not having any skill, the initial time is the time when the users start to learn something, independently of their previous skill. The intermediate time is the period when the users quickly acquire more skill until there are few advanced things they have yet to learn. From this point, the users navigate

fluently according to their initial MXL skills. As the table shows, users with a higher MXL are eager to learn faster than those with a lower MXL. The Start column indicates the percentage of skill from which the user starts the learning process, whereas the Finish column indicates the final skill achieved by a specific MXL user about the Liquid Galaxy application. Likewise, we can see that users with lower MXLs acquire more skill during the visualization process (Finish - Start). For the purpose of the present work, the next tests present studies where the users have trained before the tests. Nevertheless, the in-depth study of the learning behavior is presented in Section 6.3.

5.3 Laboratory test

With the Liquid Galaxy infrastructure, we carried out a second test with 27 people which consisted of navigating to some well known places around the world. Additionally, they answered some questions about their satisfaction while using the system. Then, we related these answers, which made up a set of qualitative performance metrics given by the Visualization Rate (VR) performance parameter, to give an idea about the efficiency and effectiveness of the system.

With these measurements related to system performance and users satisfaction, we aim to relate both metrics to be able to predict the average usability values for a specific cluster display wall with given hardware characteristics.

The different aspects involved in the usability tests that were carried out to analyze the system performance and the user satisfaction are presented next.

5.3.1 Environment

Cluster Display-Wall

The environment in which the tests were performed was a room equipped with a Liquid Galaxy system made up of 3 N_1 nodes (see Table 4.3). The monitors used in the cluster were a trio of 32" monitors set vertically in a semi-hemispheric way. The nodes were connected to a gigabit local network with a 8-Mbps broadband connection and the Squid cache enabled.

In order to acquire more specific data from the test, an eye-tracking device was used. This incorporates technology that tracks the exact point where the user's gaze is fixed on. This technology adds a powerful dimension to user research because it allows usability researchers to understand exactly what users look at and what they do not see when interacting with a user interface [76] [70].

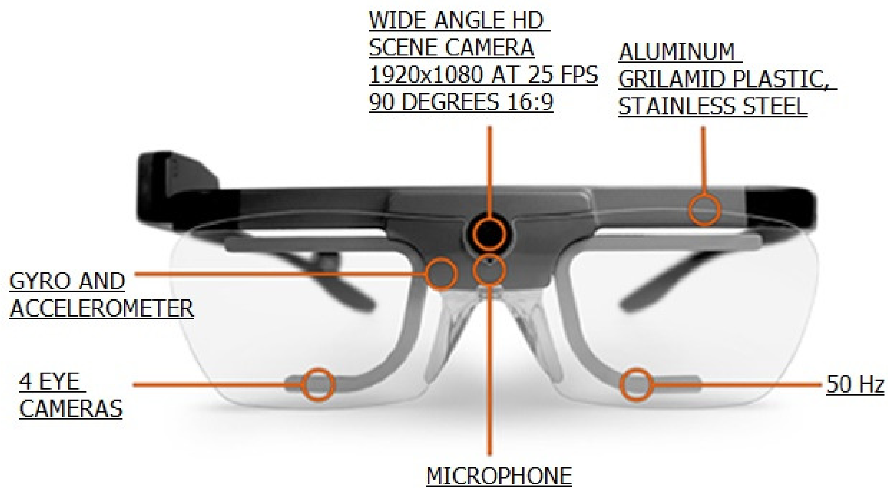


Figure 5-5: Tobii Pro Glasses 2

Traditional eye-trackers are integrated into a computer monitor. More recently, smaller and portable eye-trackers have appeared to facilitate usability research. However, these can only acquire data from one user screen, and, in our case, the system was made up of three large monitors. Accordingly, we decided to use the newer wearable Tobii Pro Glasses 2 [38]. This device, shown in Figure 5-5, enables researchers to capture truly objective and deep insights into human behavior in any real world environment. Thus, the Tobii Pro Glasses 2 allowed us to record the user's eye activity across multiple screens.

The data obtained includes video with audio, which also tells where the user is looking by depicting the point a tiny circle. This device was used to record all the user activity when using the system, which is valuable information to be used when analyzing the UX from their point of view. These data were processed and a heat map was obtained by drawing a static image with were the tiny circle was in every frame from the record. After that, this information was processed and timing statistics about the tests were obtained.

Facilitator

A facilitator was running the test and guiding the participants through it. He/she attended the participants, explained the goals of the test, the consent form and everything to make them feel comfortable. Once the test started, the facilitator could only answer specific questions or give subtle advice when the participant was struggling to complete a task.

5.3.2 Participants

The set of participants who carried out the tasks included a range of professional profiles, ages and skills with controllers. There were 27 participants in the test, 15 females and 12 males. Their age ranged from 12 to 68.

Table 5.4: Basic Participant Information

Age range	Participants	Occupation
12 to 16	4	4 Students.
17 to 21	3	2 Computer Science Programmers, 1 Computer Science Engineer.
22 to 35	9	3 Computer Science Students, 1 Pre-School Ed. Student, 1 Economics Student, 1 High School Student, 1 Restorer, 1 Industrial Engineer, 1 Computer Science Engineer.
36 to 68	11	5 Teachers, 4 management staff, 1 Customer Service Worker, 1 Shop Assistant.

Table 5.4 shows some information about occupation of the participants grouped by age range. This corresponds to information about their occupation.

Table 5.5: MXL in relation to age range

	MXL				
Age Range	1	2	3	4	5
12-16	1	-	1	2	-
17-21	-	-	-	1	2
22-35	3	2	-	3	1
36-68	5	2	2	2	-
Total	9	4	3	8	3

Table 5.5 shows the Multimedia eXperience Level (MXL) of the people involved in the test, grouped by age. Unlike the people chosen in the previous test, in this case we had a more limited public but we recruited people with different MXLs in all age ranges. In this case, the people who participated in the test were representative of the public, given that, according to our experience, it is more common that there are people with MXL1 and MXL4.

5.3.3 Test

While wearing the Tobii Pro Glasses 2 described in Figure 5-5, each participant was asked to do a series of four tasks. The test used to study the User Experience of the system was a semi-guided tour, using Google Earth, of different places across the globe.

The device used to carry out the tasks of the Liquid Galaxy was the 3D Space Navigator in Figure 5-2. As we knew that this is a different and more complex controller than a common mouse, we dedicated some minutes to helping the participants to use the device before starting the test to avoid any initial fears.

The users were also informed that all the information would be given through the monitors during the test and that they would have to answer the questions from the facilitator orally. The latter then wrote the responses down to avoid interference with the test.

For each participant, the test was composed of the following four tasks, each

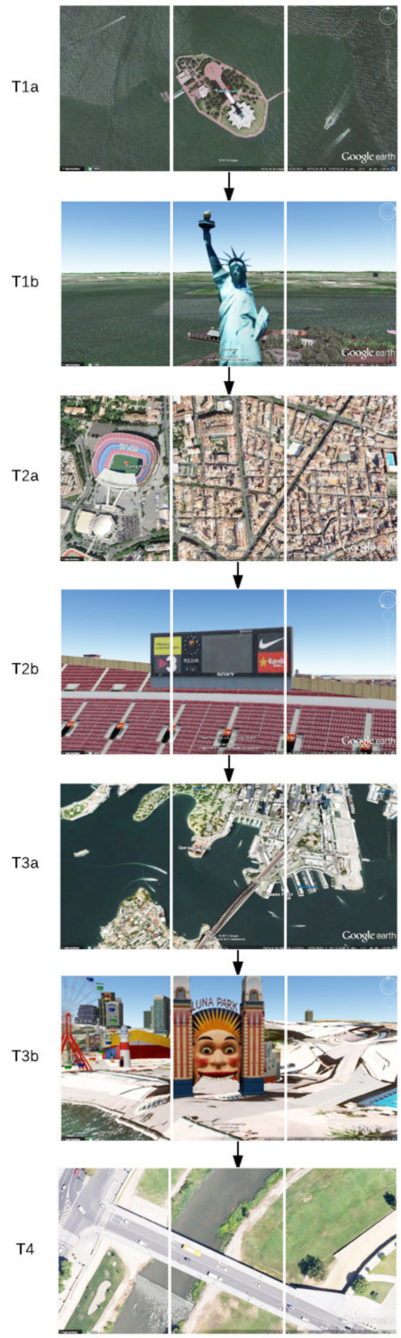


Figure 5-6: Tour Flowchart - T1a Statue of Liberty from above - T1b Statue of Liberty crown - T2a Barcelona Football Club stadium - T2b The stadium screen - T3a Sydney Harbour - T3b Luna Park - T4a Lleida bridge. Each picture is the composition of the 3 screens in the system

corresponding to navigating to one of the following well-known places:

- T1: New York (USA): Statue of Liberty.
- T2: Barcelona (Spain): Barcelona Football Club Stadium.
- T3: Sydney (Australia): Bay of the Opera House.
- T4: Lleida (Spain): City where the participants live.

Tasks T1, T2 and T3 were composed of two sub-tasks. In the first sub-task (Tia), the system positioned itself automatically at the first place in the city, so that participants had to answer the related question. This way, the participants did not use the controller for the first part of each task. However, in the second sub-task (Tib), they had to use the controller to reposition the view to be able to answer the related question. Task T4 consisted of a free flight from Sydney (T3b) to a specific point of interest in Lleida. Figure 5-6 depicts the 7 sub-tasks that corresponded to the flow of activities in the test.

Table 5.6 shows the questions the users had to answer to complete each sub-task, thus completing each of the tasks. Although the right answers were not so important, they were a way of forcing the user to interact with the system and show interest when doing the tasks. The tasks were designed to gradually increase the difficulty at each step and thus help the users' ability to control the 3D Space Navigator to progress.

After each task, with the aim of acquiring information to evaluate the UX with the minimum set of questions, we issued the same two questions to the participants:

- Q1: "From the following drawings, mark which one that better explains how you felt when performing the task". The drawings (Figure 5-7), based on the LemTool emotional tool [47], are designed to acquire the user's emotional state straight after solving each task. Among others, LemTool is an auto-report tool that can be used during the interaction with the interface for its evaluation. It allows the interface to be related to the emotion evoked. The tool consists of

Table 5.6: Description of test tasks

Task	City	Sub-task	Positioning	Questions
T1	New York	T1a	Automatic	How many vertices does the base of the Statue of Liberty have?
		T1b	Manual	How many points does the crown of the Statue of Liberty have?
T2	Barcelona	T2a	Automatic	What does it say on the Barcelona Football Club stadium stands?
		T2b	Manual	What make is the screen in the Barcelona Football Club stadium?
T3	Sydney	T3a	Automatic	How many buildings is the Sydney Opera House made up of?
		T3b	Manual	Find a structure nearby with a clown's face on it. What does it say above it?
T4	Lleida	T4	Manual	How many buses are crossing the bridge in front of the cathedral in Lleida?

Figure 5-7: Emotional choices (source LemTool)

eight figures that represent four positive and four negative emotions, combining facial expressions and body postures.

- Q2: "From 0 to 10, mark your satisfaction with the loading time of the images". The possible answers were 0, 2, 4, 6, 8 and 10. The goal of this question was to obtain a fast first-hand opinion related to the satisfaction with the image loading time. This is directly related to the VR parameter as explained in Section 5.3.5.

Table 5.7: Post-test Questionnaire

Questions	Description
FQ1	"From 0 to 10, mark your personal skill at using joysticks and remote control devices"
FQ2	"From 0 to 10, mark the ease of use when using the system"
FQ3	"From 0 to 10, mark how much you think you had to learn to use the system"
FQ4	"From 0 to 10, mark how much technical help you think would be needed to use the system"
FQ5	"From 0 to 10, mark if you believe that everyone could learn the system quickly"
FQ6	"From 0 to 10, mark your personal satisfaction when using the system"
FQ7	"From 0 to 10, mark how secure you felt when interacting with Liquid Galaxy"
FQ8	"From 0 to 10, mark how much would you be eager to use the system again"
FQ9	"From 0 to 10, mark your perception of system complexity"

Finally, after finishing all four tasks, participants had to answer the post-test questionnaire, shown in Table 5.7. This last questionnaire is inspired by the System

Usability Scale (SUS), which has long been accepted as an industry standard [15] and adapted to our tests with the Liquid Galaxy system.

5.3.4 Satisfaction Results

By collecting the data of the eye-tracker and processing the answers to the questions given by the participants (see Tables 5.6 and 5.7), we obtained a set of qualitative measures of the test. These are presented next.

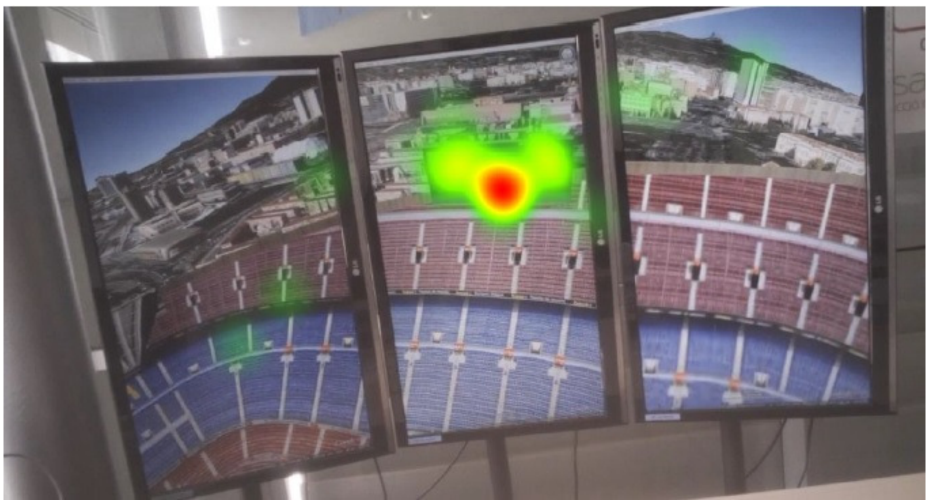


Figure 5-8: T2b Heat map - Common Pattern in the whole study

Heat maps show where the user was looking most of the time. After processing the information from the Tobii Pro Glasses 2 eye-tracking device, we can say that all users always looked at a single spot, the most interesting one and the potential answer to their task question. Figure 5-8 shows an actual user's heat map while performing task T2b. This heat map represents the common pattern found with this device throughout this test. Moreover, another pattern that we found is that, in general, the participants moved the target to the central screen whenever possible, independently of the number of screens Liquid Galaxy was made up of.

Table 5.8 shows the level of satisfaction (question Q1) when performing the tasks in New York (NY), Barcelona (BCN), Sydney (SYD) and Lleida (LL). Furthermore, Table 5.9 shows the feeling of the users about the waiting time for each of these tasks

Table 5.8: Results of Q1 Question

Tour	Joy	Desire	Fascination	Satisfaction	Sadness	Disgust	Boredom	Dissatisfaction
NY	7.1%	7.1%	35.7%	42.9%	7.1%	0%	0%	0%
BCN	7.1%	14.3%	21.4%	42.9%	0%	7.1%	0%	7.1%
SYD	7.1%	0%	28.6%	57.1%	0%	0%	0%	7.1%
LL	7.1%	14.3%	35.7%	42.9%	0%	0%	0%	0%

Table 5.9: Results of Q2 Question

Tour	0	2	4	6	8	10
NY	0%	0%	0%	14.3%	42.9%	42.9%
BCN	7.1%	7.1%	7.1%	28.6%	42.9%	7.1%
SYD	0%	7.1%	7.1%	7.1%	50%	28.6%
LL	0%	0%	7.1%	14.3%	35.7%	42.9%

(question Q2). We can see that the majority of the participants provided positive responses when using the system while doing the tests whenever everything was functioning and the waiting times were short. In general, 90% of the participants gave positive responses to the tours of NY, SYD and LL, with a few cases of discomfort. However, this percentage was lower in BCN because the user had to move the Google Earth view into a position showing a significant portion of the city buildings, thus, making the application download a considerable amount of data. Because of this, some people felt that they had to wait much longer than in other tasks, especially the most demanding users, including the youngest participants or those with more technology knowledge. As a consequence, the results for BCN in Table 5.8 show that 14.2% of participants had negative feelings. This correlates with question Q2 about the waiting time shown in Table 5.9, where 21.3% considered the waiting time unacceptable in the case of BCN (values lower than 5). Despite some people being frustrated by the wait, they answered more positively in question Q1 than expected. This leads us to think that they were enthusiastic about the system as it was new and fun for them. We can also say that many of them found the system challenging, but not impossible, forcing themselves to perform better and feel with a big sense of accomplishment, which could affect the results by giving higher values.

The answers obtained for the post-test questionnaire described in Table 5.7 can

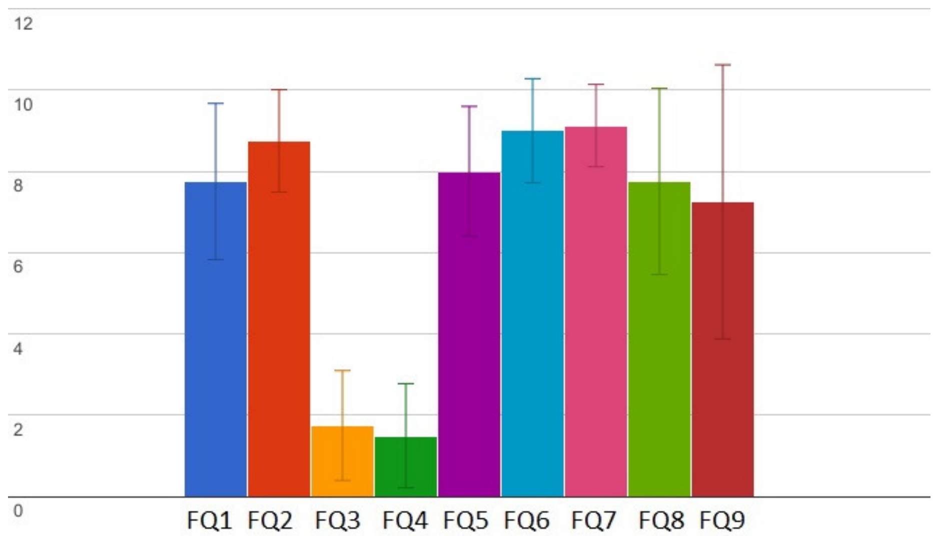


Figure 5-9: Mean and error bars from post-test questions - Skilled group (16 people)

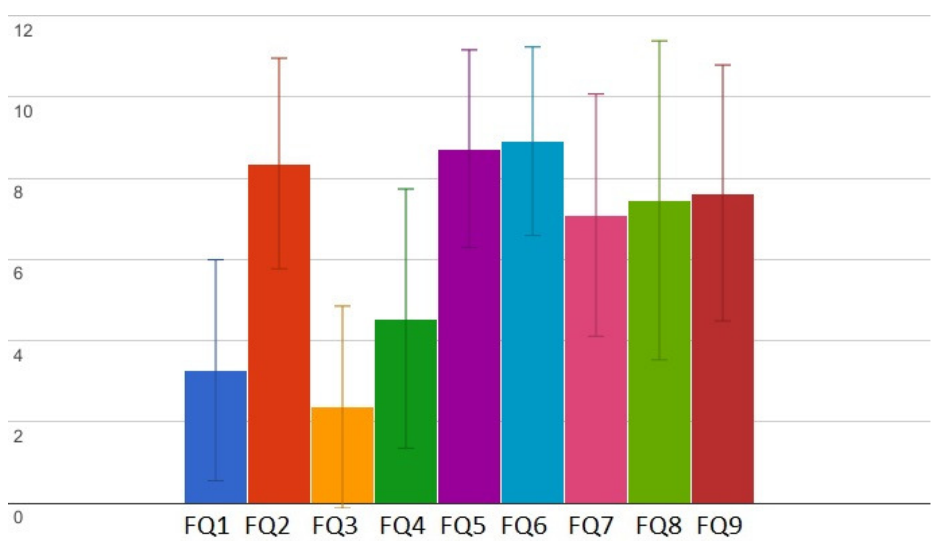


Figure 5-10: Mean and error bars from post-test questions - Less skilled group (11 people)

be divided into two main groups: a more skilled profile corresponding to MXL3 to MXL5 in Table 5.4, whose answers, are presented in Figure 5-9, and a less skilled group corresponding to MXL1 and MXL2, whose answers are presented in Figure 5-10. This skill difference can be observed in the answers to question FQ1 in both Figures. Taking the answers to FQ2 and FQ3 into account, both groups stated that they felt no need for prior knowledge and that the system was easy to learn and use, regardless of their experience. The main difference was about how secure they felt while using the system (FQ7), where the second, less experienced, group felt less secure while also noting that they needed more technical help. Also, the differences between the responses of the two groups to the question about whether they thought technical help would be needed (FQ4) were remarkable, and correlated with their technical skill level.

In relation to the deviation of the answers, we can see that was generally higher in Figure 5-10 due to the differences in age and occupation among participants in this group. Likewise, it is worth pointing out that the question about how complex they felt the system was (FQ9) generated the widest range of opinions in both groups. This is because some of the participants did not know what to answer as it was a new system and it led them to think that it would be too complex for others. Thus, some answered thinking about other people with less or similar experience to themselves. This behavior was expected as the questions were designed to take into consideration other people evaluating the system, and not how they used it.

5.3.5 Effectiveness and Efficiency Results

One of the goals of this section is to establish a relationship between the satisfaction (associated with the subjectivity of the participants) and the effectiveness and efficiency measures regarding the system performance (with no subjectivity associated). The Visualization Rate (VR) metric presented in Chapter 4 was used to obtain the effectiveness and efficiency of the system.

The system performance metric was monitored throughout the test, but only the tasks where comparisons could be made were recorded. Those tasks are the ones that are guided, because Google Earth follows the same path from one place to another in a guided task independently of the user, thus providing fully comparable values. This is not applicable when the user moves between locations freely, as it is almost impossible to use the same flight path for the same step, even with the same user. This means that only tasks T1a (NY), T2a (BCN) and T3a (SYD), depicted in Figure 5-6, were monitored to calculate the VR.

Table 5.10: VR Ranges for T1a, T2a and T3a sub-tasks

	VR Ranges			
Tour	0-5%	5-10%	10-15%	15-30%
NY	28.6%	42.9%	21.4%	7.1%
BCN	57.1%	35.7%	7.1%	0%
SYD	14.3%	57.1%	21.4%	7.1%

Table 5.10 shows the percentage of the VR categorized in the VR ranges obtained by different users for NY, BCN and SYD for the first part of their respective tasks (Tia). For simplicity, these values are categorized into ranges shown as columns in the table. For the case of NY and SYD, the highest VR values were near 25%, but for BCN, the VR gave lower values. This is due to the fact that in Task 2a in BCN, a lot of 3D buildings had to be rendered in order to view the desired place and, as a consequence, the loading time was higher and that users did not need to have the images fully loaded to answer the question. Another point to highlight is that all VR values were below 30%. This is because, in our test, the user only wanted to answer the questionnaire, and was not interested in the specific imagery. In a free flight around points of interest by a given user, he/she would spend more time looking at a specific point, which would increase the VR metric. Another point to take into account is that this test achieved lower VR values than the previous test of Section 5.2. This was due to the nature of the test, as the answers to the questions set out by the test could be obtained as soon as the essential part of the visualization media had loaded.

5.3.6 Discussion

The results of VR obtained in the user tests presented before were analyzed in order to study the relation between the system performance and personal satisfaction of the users when using the Liquid Galaxy system. In general, the results showed that there is a relation between the VR values and the users' satisfaction levels. From our study, we can see that SYD had the highest VR and also the highest satisfaction level, whereas BCN had the lowest VR and level of satisfaction. Despite this, people were generally satisfied and happy to use the system, even though the VR values were rather low. This behavior is reflected in the results obtained from BCN.

Table 5.10 shows the VR for the three first tours (NY, BCN and SYD). We proceeded to study the correlation between the VR parameter, given in Table 5.10, and both the satisfaction parameter (Q1 results of Table 5.8), and performance (Q2 results given in Table 5.9).

Table 5.11: Variances and Correlations for Q1, Q2 and VR

Tour	σ^2_{Q1}	σ^2_{Q2}	σ^2_{VR}	$\sigma_{Q1,VR}$	$\sigma_{Q2,VR}$	$r_{Q1,VR}$	$r_{Q2,VR}$
NY	0.88	1.40	6.48	3.70	5.59	0.65	0.62
BCN	5.34	1.91	3.04	2.49	1.73	0.15	0.30
SYD	1.23	1.69	4.16	2.67	4.43	0.52	0.63

Table 5.11 shows the variances (σ^2), covariances (σ) and linear correlations (r) for the answers to Q1 and Q2, and the VR performance metric. Correlation values closer to 1 or -1 mean that there is respectively a strong direct or inverse relation between the metrics, while having values closer to 0 means that there is no relation between them. We can see that, in general, there is a positive correlation in all the cases. Likewise, as expected, both correlations ($r_{Q1,VR}$ and $r_{Q2,VR}$) in NY and SYD are very strong, while correlations in BCN are the weakest. The reason for this lower correlation in the BCN case is that, on one hand, it had the worst VR values due to the high number of 3D buildings to be loaded, but, on the other hand, people maintained a high interest and satisfaction when flying above Barcelona because it was the best known tour for Spanish users. In addition, it is worth pointing out that,

in general, the $r_{Q2,VR}$ correlation is slightly stronger than the $r_{Q1,VR}$. This behavior is normal given that the Q2 question asked to the users about their feelings in relation to the loading time and the VR metric was calculated from the same loading time.

These correlation results lead us to think that the VR metric constitutes an orientative value to know the minimum required performance to guarantee satisfaction for the user with the system.

5.4 Field test

It is known that users tend to behave a bit different in a controlled environment than when they are not observed. Moreover, if the user does not have an actual personal objective to complete, the final outcome is not usually the same as when the user has a genuinely personal goal to achieve. Our approach to try to reduce this behavior was to take the experimentation into a real environment, a travel agency, where users could visit the places they were going to travel to beforehand. This test was very similar to the test described in Section 5.3 to enable comparison between the two. In order to achieve this, the same assumptions and classifications of participants were also applied in this analysis.

5.4.1 Environment

Cluster Display-Wall

The environment where the tests were performed was a travel agency named Ilertravel, located in Molleussa (Spain), equipped with a Liquid Galaxy system made up of 3 N_1 nodes (see Table 4.3). The cluster used a trio of 24" screens monitors set horizontally in a semi-hemispheric way. The nodes were connected to a gigabit local network with a 5-Mbps broadband connection and the Squid cache enabled.

Facilitator

A facilitator ran the test and guided the participants through it. He/she attended to the participants, explained the goals of the test, the consent form and everything to make them feel comfortable.

In this test, we wanted to lessen the impact of controller usage, so, the facilitator was able to help when needed. With this measure, it did not matter if the user was unsure how to use the controller properly.

5.4.2 Participants

There were 13 participants, 8 females and 5 males. The multimedia skills of the users were balanced and the ages ranged from 21 to 70. In this case, we did not focus on age, as we consider the MXL a good comparison parameter. Also their professions were not focused on multimedia expertise, although there were some people who worked with computers as users everyday. Once the test started, the facilitator was able to help when needed.

Table 5.12: MXL information grouped by age and profession

Age range	Participants	Occupation
17 to 21	1	1 clerk.
22 to 35	1	1 housewife.
36 to 68	11	1 firefighter, 2 travel agents, 1 clerk, 2 housewives, 1 supermarket director, 2 teachers, 2 technicians

Table 5.12 shows the age and occupation of the people in each age range.

Table 5.13 shows the MXL of the people in the test, grouped by age. We observed that this travel agency tends to have adult customers, which is the reason for there being people with lower MXL values than in the other tests.

Table 5.13: MXL in relation to age ranges

	MXL				
Age Range	1	2	3	4	5
17-21	-	1	-	-	-
22-35	-	-	1	-	-
36-68	4	3	2	1	1
Total	4	4	3	1	1

5.4.3 Test

The test involved a questionnaire with the same Q1 and Q2 questions described in Section 5.3, but for two places that the user would choose by himself/herself. After both tasks were finished, the user was given the same series of post-test questions as in the previous test and described in Table 5.7.

The procedure was the following:

1. The users filled in a short pre-test questionnaire (name, age, profession and estimated skill level) and were given instructions on how to use Liquid Galaxy.

2. The facilitator trained the users, focusing in the interaction with the 3D Space Navigator mouse and let them try the system for a convenient time to ensure that they knew how to use it.

3. The facilitator changed the view of the system to show the place chosen by the user.

4. Then, the users moved freely using the Google Earth application while asking for assistance when needed or when the facilitator thought they might need help.

5. When the user said they had had enough time to visualize the place of choice, they were given two short questions (Q1 and Q2) about how they felt during the task. This was repeated for another place chosen by the user.

6. After both tasks are completed and answered, the users answered a series of post-test questions (FQ1 to FQ9 from Table 5.7).

5.4.4 Satisfaction Results

In this test, the environments were separated into high (above 160 MB), medium (between 160 MB and 110 MB) and low (below 60 MB) data density locations, as each user chose a different place to visit. Therefore, the results show the average achieved in the different environments visited by the users in this test. The number of places in each category was as follows: 9 high-density locations, 8 medium-density locations and 9 low-density locations.

Table 5.14: Results of Q1 Question

Data Density	Joy	Desire	Fascinat.	Satisfac.	Sadness	Disgust	Boredom	Dissatisfac.
High	11.1%	11.1%	55.5%	22.2%	0%	0%	0%	0%
Medium	37.5%	12.5%	12.5%	60%	0%	0%	0%	0%
Low	22.2%	11.1%	11.1%	55.5%	0%	0%	0%	0%

Table 5.15: Results of Q2 Question

Data Density	0	2	4	6	8	10
High	0%	0%	0%	0%	66.7%	33.3%
Medium	0%	0%	0%	0%	62.5%	37.5%
Low	0%	0%	0%	11.1%	22.2%	66.6%

Table 5.14 shows the level of satisfaction (question Q1) when performing the tests in the different environments chosen by the users, while Table 5.15 shows the feelings of the users about the waiting time for each task (question Q2). The results show that all the participants had positive feelings when performing the different tasks.

As for the results in Q2 (Table 5.15), the users responded positively regarding the waiting times of the system. It is notable and expected that for the low data-density locations there would be a higher value of satisfaction than for the other two types of environment.

We find again that the users were enthusiastic about the system but, moreover, they found it more satisfying as they could visit places of real interest for them instead of locations given by the facilitator. This is reflected in the values of both tables with higher levels of satisfaction than in the previous test.

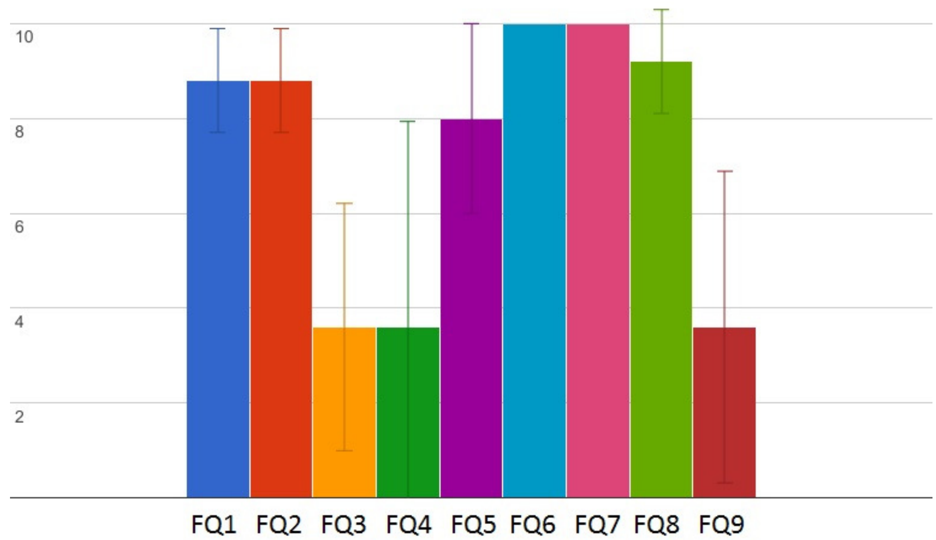


Figure 5-11: Mean and error bars from post-test questions - Skilled group (5 people)

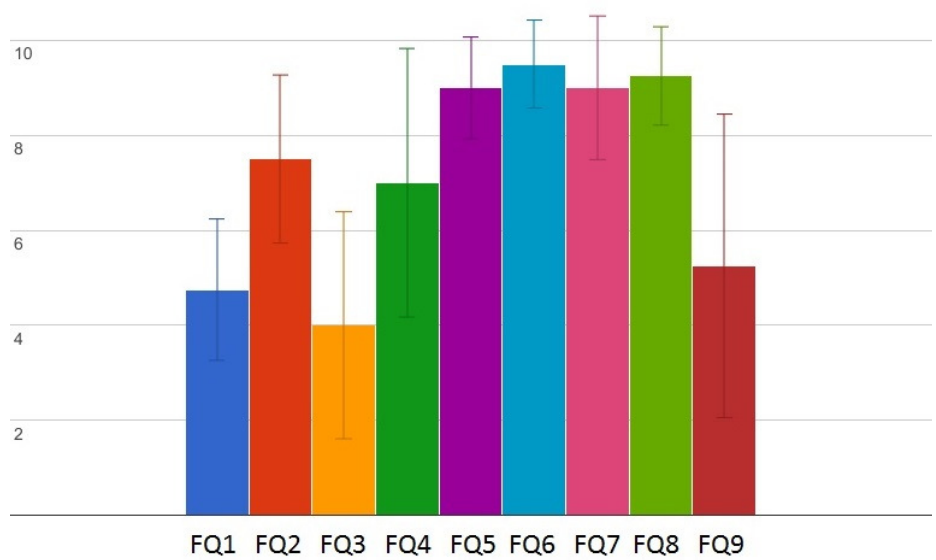


Figure 5-12: Mean and error bars from post-test questions - Less skilled group (8 people)

The answers obtained for the post-test questionnaire described in Table 5.7 were divided into the same two main groups: a more skilled profile corresponding to MXL3 to MXL5 in Table 5.13, whose answers are presented in Figure 5-11, and less skilled people, corresponding to MXL1 and MXL2, whose answers are presented in Figure 5-12. This skill difference can be observed with the answer to question FQ1 in both figures. As can be seen, the results for both groups were very similar. All users agreed that the system was easy (FQ2, FQ9) and pleasant to use (FQ8), and felt satisfied (FQ6) using it. The main differences was that the skilled group felt that they might not need as much technical help (FQ4), believed that everyone could learn quickly how to use the system (FQ5) and felt more secure (FQ7) than the less skilled group.

In comparison with the previous test, the users in this test felt the system less complex (FQ9), probably because the facilitator was guiding them fully. This also led to an increase in the feelings of satisfaction (FQ6) and security (FQ7). In contrast, this made the participants think that they had to learn more how to use the system (FQ3).

In relation to the deviation of the answers, we can see that, in general, this was higher in Figure 5-12 due to the low number of skilled participants.

5.4.5 Effectiveness and Efficiency Results

Table 5.16: VR Ranges for the Different Environments

	VR Ranges			
Data Density	0-5%	5-10%	10-15%	15-30%
High	66.7%	33.3%	0%	0%
Medium	75%	0%	25%	0%
Low	22.2%	22.2%	0%	55.5%

Table 5.16 shows the VR values, categorized into VR ranges, obtained by different users for the environments, categorized in density ranges, they visited. As can be seen, higher density environments tend to have lower VR values, which is expected behavior. It is worth pointing out that, again, all VR values were below 30%. This may be caused by the slow Internet connection available in the travel agency.

In comparison with Table 5.10, the values obtained in this test were generally lower due to the bad broadband connection available in the travel agency.

5.4.6 Discussion

The answers from the post-test questionnaire confirmed that the people were satisfied with the system in every aspect, although some answered that they would need the help of a technician to use it or that they had to learn a lot to use the multimedia application properly. Also, it is worth pointing out that all users felt that they improved their multimedia skill after using the system.

Table 5.17: Variances and Correlations for Q1, Q2 and VR

Tour	σ^2_{Q1}	σ^2_{Q2}	σ^2_{VR}	$\sigma_{Q1,VR}$	$\sigma_{Q2,VR}$	$r_{Q1,VR}$	$r_{Q2,VR}$
High	1.03	1.13	3.11	2.02	2.63	0.63	0.74
Medium	1.32	0.97	3.92	3.19	3.35	0.61	0.88
Low	1.05	1.37	12.88	11.15	15.59	0.82	0.88

Table 5.17 shows the variances and correlations for the Q1, Q2 and VR. In general, the results achieve a high positive correlation for both the related parameters and the three data density images. Note that the highest correlation (near 90%) was obtained for the relation between Q2 and the VR parameter for the cases of low and medium density, which are the cases that had the lowest loading time and, as a consequence, the users were better satisfied with the load time. This explanation is consistent given that the highest correlation between Q1 and VR was obtained for the images with the lowest density.

In relation to the results obtained in the case described in the previous test, we can see that we obtained much better correlation given that, in this case, the users navigated wherever they wanted and, as a consequence, all the answers had higher satisfaction values.

5.5 Conclusion

To acquire knowledge of system performance, we used the Visualization Rate (VR) metric that indicates the average CPU idle time of system nodes, in such a way that the time intervals when the CPU is idle (under a certain threshold) mean that the images were fully loaded. A VR value equal to 0% means that the system is ineffective and any value above 0% describes the percentage of efficiency of the system. During the tests, we observed that the VR was generally below 30% and above 0%. Although this could seem a low performance, it was due to the behavior of the participants when completing the tasks. Some users in tests 1 and 2 only had to wait for the image to load partially to be able to answer the questions (usually on the center screen, excluding the others) and that is why the VR values were relatively low. In relation to the field test, the main reason for the low VR was the poor Internet connection used in that test.

The experimentation also shows a clear difference between the two groups of people. The first group, usually younger, was more experienced and tended to be more impatient, as they have better skill of this kind of technology, whereas the people in the second group were usually less experienced and tended to be more impressed.

Additionally, we analyzed the existing relationship that could be established between the results of the study: satisfaction and VR performance. We were able to confirm that the VR constitutes an indicative measure of the satisfaction level when using a specific system infrastructure. This was confirmed by the statistical correlation between VR and the questionnaires. It shows how the answers to both post-task satisfaction questions are related to the VR values achieved in those tasks.

Moreover, we found that satisfaction levels could be higher than expected, taking low VR values into account, whenever the tour was of high interest for the users.

Likewise, the results varied with the setup of the different types of tests we carried out. They showed that the participants may be apprehensive about using a new controller device, but will feel more satisfied when the nature of the tests reveals information of more interest for them.

Our results reveal that with the knowledge of the system performance, which can be calculated from objective metrics, we can estimate the suitability of a cluster display-wall for use under certain user requirements. This is a very encouraging result for us, as it can facilitate the spread the use of the Liquid Galaxy platform to a broad number of users and fields (education, professional, research, etc.).

Chapter 6

Modeling

Having stated in previous studies that the performance of a cluster display wall built with commodity hardware is adequate to run Google Earth, our objective in this chapter is to model the behavior of the system in a way that the performance for any given Liquid Galaxy cluster could be predicted compared to a reference system.

From the study of the different tests carried out with Liquid Galaxy in previous chapters, we define a theoretical schema used to model the behavior of the cluster-based system running Google Earth. The following sections explain how we defined this model, the definition of its different parts and the relationship between them.

6.1 Performance modeling

We assume that the performance of the Liquid Galaxy system is calculated according to the VR metric, as defined in Chapter 3. In order to do this, there are two parts to be considered when calculating the VR: the time when the CPU is processing and the time when the CPU is idle. The time when the CPU is processing data, named Processing Time (T_{CPU}), is directly related to the infrastructure on which the application is running, together with the resolution of the image to be loaded. On the other hand, the time when the CPU is idle (T_{idle}) is more related to the user, who decides how long he/she wants to visualize the imagery. Note that the total visualization time (T_{total}) is the sum of T_{CPU} and T_{idle}, so it depends on the factors

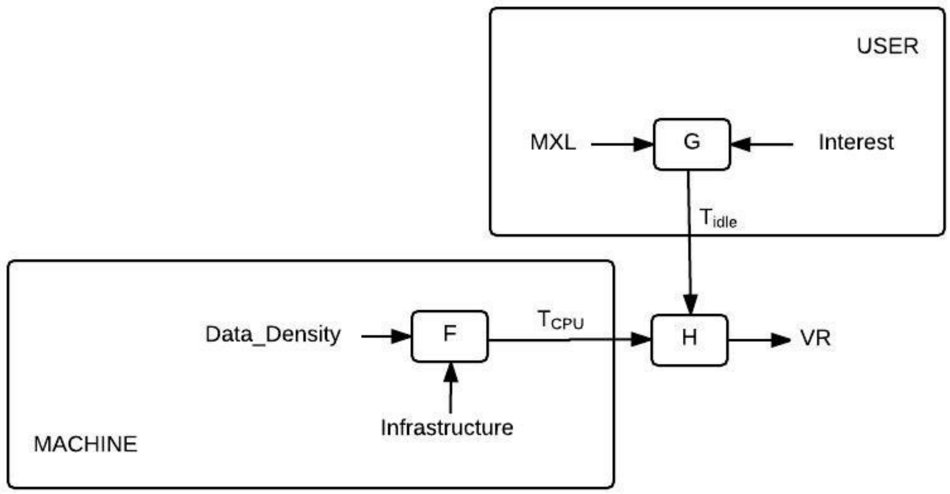


Figure 6-1: Scheme to model VR

previously described.

According to the previous reasoning, the model to be used to calculate the system performance is depicted in Figure 6-1, where two separated parts can be distinguished: Machine and User modules.

- The *Machine* module represents how efficiently a given infrastructure processes multimedia data. This module will deliver the T_{CPU} value as a function of the density of the data to be processed and the characteristics of the infrastructure:

$$T_{CPU} = F(Data\_Density, Infrastructure) \tag{6.1}$$

- The *User* module represents how much a user is interested in the visualized data, taking into account his/her Multimedia eXperience Level (MXL). So, the output function is as follows:

$$T_{idle} = G(Interest, MXL) \tag{6.2}$$

According to this, to obtain the overall system performance, the inputs of these modules are separated into four key factors:

- *Infrastructure.* When trying to model the performance of Liquid Galaxy, we found that there were some minimum hardware requirements. As shown in Chapter 4, those requirements are identified as the minimum power of the nodes in the cluster and the minimum broadband bandwidth. Chapter 4 explored how the power of the slowest node in the cluster constitutes the major issue in the performance of the Liquid Galaxy system. Likewise, the Google Earth application has a list of recommended minimum system requirements, which are as follows:

  - CPU: Pentium 4 2.4GHz or AMD 2400xp
  - RAM: 1GB
  - Network Speed: 768 Kbits
  - Graphics Card: DirectX9 and 3D capable with 256MB of VRAM

  Using these requirements and our experimentation with the Liquid Galaxy environment, we defined our reference cluster configuration in order to compare other systems to it. Our reference cluster was defined inside the following ranges:

  - CPU: from 1.6GHz to 3.0GHz
  - RAM: 8GB at 1600MHz
  - Network Speed: from 6Mbits to 100Mbits
  - Graphics Card: NVidia GT620

- *Data density.* The amount of data to be processed by the system is an important parameter to be taken into account when calculating its performance. The more data to be downloaded, the longer the CPU will take to process it. This amount of data is characterized by the number of calls to the *Draw function* given by Google Earth to represent the imagery of the application. Note that this number of calls is given by the Google Earth application and is totally independent of the hardware.

- *Multimedia eXperience Level (MXL)*. When calculating the VR metric in previous tests, we concluded that the T_{idle} was different for every user who interacts with the system. While doing some tests, we noticed that the multimedia knowledge of, or ability with, this kind of system was a significant parameter, given that the higher the MXL, the lower the T_{idle}.

- *Interest*. The interest of a user depends on how keen he/she is to visualize a city or place and, with the MXL, this is reflected in the time that the system is idle.

  The user tests carried out in the laboratory and field tests in Chapter 5 show how interest varies depending on the type of test, with higher interest in the field test. Thus, the higher the interest of the user, the higher the T_{idle}. Taking previous experimentation into account, throughout the chapter, we consider "low interest" as when the user has to complete an task imposed by the test facilitator, whereas "high interest" corresponds to when the users have freedom to navigate and are therefore more interested in the imagery.

6.2 Machine modeling

In this Section, we develop the modeling of the *Machine* module described in Figure 6-1 by defining the T_{CPU} function as $F(Data\_density, Infrastructure)$. We performed some tests to analyze the parameters involved in the Machine module. The tests consisted of carrying out different tours using a very long jump time between the locations to be loaded, so the system was able to load the imagery completely at every step in the tour. By doing so, the impact of the user is negligible. We chose 6 different tours with different imagery size and polygon complexity. All the tours were run in a Liquid Galaxy cluster made up of 3 N_1 nodes (see Table 4.3) and a broadband connection of 100-Mbps broadband connection, which is inside the ranges of the reference cluster described in Section 6.1.

Table 6.1 shows the results obtained with the following parameters:

- T_{CPU} is the CPU time in seconds spent processing the images.
- $Draw\_calls$ are the number of Draw function calls needed to draw all the images.
- $Size$ is the amount of data downloaded from the Internet.
- T_{dl} is the time needed to download the imagery, which is directly dependent on the size of the imagery.

Table 6.1: Test results

Tour	$T_{CPU}(s)$	Draw_Calls	Size (MB)	$T_{dl}(s)$
Sahara	120s	154	23	7s
Horsens	206s	277	28	9s
Alps	250s	377	87	27s
Barcelona	254s	310	123	38s
Venice	455s	588	152	47s
Paris	654s	644	280	87s

As can be observed, T_{CPU} is much higher than T_{dl}. Thus, we can assume that an image is downloaded and processed simultaneously, making T_{dl} negligible. So, if we take this assumption into account, we can state that the actual time used to call a single Draw function (T_{Draw}) can be defined as the rate between the T_{CPU} and the number of Draw calls:

$$T_{Draw} = \frac{T_{CPU}}{Draw\_Calls} \tag{6.3}$$

Table 6.2: Time per Draw function (T_{Draw_ref})

Power of slowest node	Broadband Bandwidth					
	100Mbits	80Mbits	40Mbits	20Mbits	10Mbits	6Mbits
3GHz	0.34s	0.36s	0.37s	0.71s	0.83s	1.03s
2.4GHz	0.39s	0.40s	0.42s	0.66s	0.84s	1.02s
2GHz	0.48s	0.50s	0.52s	0.64s	0.80s	1.02s
1.6GHz	0.52s	0.55s	0.56s	0.70s	0.83s	1.03s

To establish a reference value of T_{Draw}, we performed some tests with different configurations of the infrastructure. The tests were carried out using 24 configurations of homogeneous clusters varying two characteristics: the CPU, ranging from 1.6GHz to 3GHz, and the broadband connection, ranging from 6 Mbits to 100 Mbits. Table 6.2 shows the results for the T_{Draw_ref} from the mentioned tests. These show a clear difference under and over 40 Mbits of broadband connection, which indicates that the broadband bandwidth is a bottleneck when its value is below this threshold. Therefore, the results obtained for each system under the 40Mbits have similar values independently of the CPU power. Statistically, the loss of time between 6 Mbits and 10 Mbits is 20%, 18% from 10 Mbits to 20 Mbits and 31% from 20 Mbits to 40 Mbits on average. On the other hand, the left side of the table shows the low influence of the connection bandwidth when the network resources are abundant. In this case, the CPU speed is the dominant parameter, where configurations with better CPU achieved lower T_{Draw_ref} values.

After having tested different configurations and obtained reference values for every broadband connection, we can state that Equation 6.4 can be used to calculate an approximation to the Processing Time (T_{CPU}).

$$T_{CPU} = Draw\_Calls \times T_{Draw\_ref} \tag{6.4}$$

where T_{Draw_ref} is a value that depends on the configuration of the broadband connection and is given by Table 6.2, while $Draw\_Calls$ depends on the complexity of the imagery and is given directly by Google Earth in real-time.

6.3 User Modeling

This section studies the users interest on the multimedia application and their skill or knowledge of using those applications (MXL) to explain how the user affects the T_{idle}.

6.3.1 Modeling of the Learning Period

The MXL is a value that measures the skill of a person at using multimedia applications. It is worth pointing out that, although users have some knowledge of the system they are going to use, they could have different starting knowledge and also different learning curves.

In our tests, we found that there was a difference between having users who had already used Liquid Galaxy before the tests and those who had not. This behavior can be represented by a learning curve that gives information about the point where the user is when talking about knowledge of the multimedia system. This curve shows how much time an individual needs to start learning, how much time it takes him/her to learn and how much he/she can potentially learn.

From our tests and experience, we defined a formula from a four parameter logistic regression formula that approximates the learning curves to every MXL by using the values given in Table 5.3 in Chapter 5.

$$MXL(t) = d + \frac{a - d}{1 + (\frac{t}{c})^b} \tag{6.5}$$

where a is the minimum asymptote (Start value of Table 5.3, d is the maximum asymptote (Finish value of Table 5.3, b is the hill's slope (calculated using a statistical tool) and c the inflection point (half-way through the Intermediate period of Table 5.3 adding the Initial period). Five different MXL levels were defined and a curve was calculated for each one. The a, b, c an d parameters were calculated for each curve and they were as follows:

$$\begin{aligned}
MXL1(t) &= 40 + \frac{10-40}{1+(\frac{t}{50})^5} \\
MXL2(t) &= 50 + \frac{20-50}{1+(\frac{t}{45})^5} \\
MXL3(t) &= 80 + \frac{50-80}{1+(\frac{t}{30})^6} \\
MXL4(t) &= 90 + \frac{70-90}{1+(\frac{t}{22})^{5.3}} \\
MXL5(t) &= 100 + \frac{80-100}{1+(\frac{t}{17})^4}
\end{aligned} \tag{6.6}$$

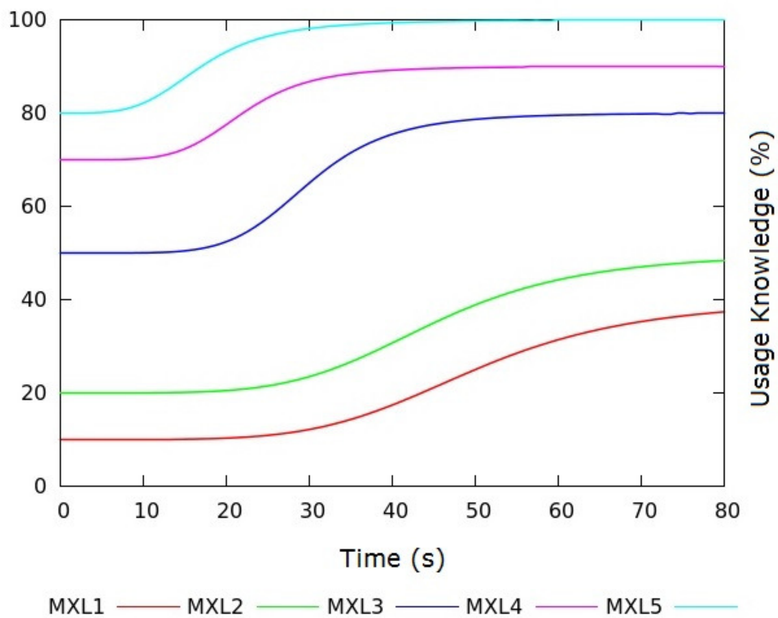


Figure 6-2: MXL learning curves

Figure 6-2 depicts the learning curves corresponding to each of the formulas in Equation 6.6. These tell the skill level of a person with a given MXL throughout the learning period. As can be seen, the curves have different slopes depending on the MXL the user is in, as people with a higher MXL will tend to learn faster. Also, the final value of the learning curves match our other studies in which we identified 2 groups of users, ones with high skill (above 50% of the knowledge, corresponding to MXL3, MXL4 and MXL5) and others with low skill (below 50% of the knowledge, MXL1 and MXL2). Moreover, the number of things someone can learn is higher if their starting point is lower, as there are more things to learn than there are for someone who already knows most of the system. For instance, if we look at MXL5, the starting point is 80% and the final is 100%, while for MXL1, the starting point is 10% and it ends near 40%.

6.3.2 Modeling of T_{Idle}

The experimentation carried out in Section 5.2 was taken in order to analyze the relationship between the MXL and the T_{idle} parameter, given that in that test we classified the users by their MXL and this allows us to discriminate between them according to their skills.

Table 6.3: T_{idle} average according to MXL and interest

Interest	MXL1	MXL2	MXL3	MXL4	MXL5
High (HI)	59s	56s	32s	22s	17s
Low (LI)	36s	33s	13s	10s	6s
MXL_factor_HI	1.84	1.75	1	0.69	0.53
MXL_factor_LI	2.77	2.54	1	0.77	0.46

Table 6.3 shows the average T_{idle} the users achieved in the test. It can be observed that the T_{idle} is higher when interest is also high. At the bottom of the table, the MXL factor for both high interest (MXL_factor_HI) and low interest (MXL_factor_LI) is shown. This is calculated as the relation between the T_{idle} of each MXLi and MXL3 (used as reference). The MXL_factor will be used to weight the influence of each users MXL in relation to the T_{idle}. According to the MXL_factor, we can observe that the higher the interest, the lower the factor (with the exception of MXL5). Moreover, there is a clear trend that shows that users with a lower MXL have a higher T_{idle} which thus increases the MXL_factor values.

Table 6.4 summarizes the data obtained in the laboratory and field tests in Chapter 5, carried out in a cluster made up of 3 N_1 nodes (see Table 4.3). The table shows the T_{idle_ref} values that correspond to the average T_{idle} achieved for a reference machine in

Table 6.4: T_{idle_ref} according to data density and interest for a reference cluster

	Data Density		
Interest	High	Medium	Low
High	8s	10s	19s
Low	8s	8s	7s

the laboratory and field tests described in Chapter 5. The values are categorized by the interest level of the user and the data density of the test. As with the interest, we categorized the values for the different scenarios tested depending on their download size into high (above 160 MB, medium (between 160 MB and 110 MB) or low (below 60 MB) data density. The results show that, for high interest, the average T_{idle_ref} significantly increases as the data density decreases from high to low. This is not the case for low interest, as the value achieved remains at about 8s, which is due to the user trying to answer the questionnaire as quickly as possible. Also, we can observe that the more interest the user has in the test, the more time he or she will observe the imagery.

Thus, using the values from the table, we can define the T_{idle} for any imagery with the following equation:

$$T_{idle} = T_{idle\_ref} \times MXL\_factor \tag{6.7}$$

where T_{idle_ref} is the idle time for our reference machine and the MXL_factor is defined using Table 6.3.

6.4 VR Modeling

This Section combines the models of the *Machine* and *User* modules in order to obtain a mathematical model to calculate the value of the VR (VR_Model) in ideal scenarios.

The VR_model is calculated using Equation 4.1, which is $VR = \frac{T_{idle}}{T_{total}}$, where

$T_{total} = T_{CPU} + T_{idle}$, and thus the final equation is as follows:

$$\begin{aligned} VR\_Model &= \frac{T_{idle}}{T_{CPU} + T_{idle}} = \\ &= \frac{T_{idle\_ref} \times MXL\_factor}{Draw\_Calls \times T_{Draw\_ref} + T_{idle\_ref} \times MXL\_factor} \end{aligned} \tag{6.8}$$

where T_{CPU} can be easily obtained using Equation 6.4 by relating the $Draw\_Calls$ for a given imagery, supplied by Google Earth itself, and the T_{Draw} obtained from the values in Table 6.2.

For T_{idle}, the user interaction is taken into account and is calculated using Equation 6.7, taking the values for T_{idle_ref} from Table 6.4 and the MXL_factor from Table 6.3.

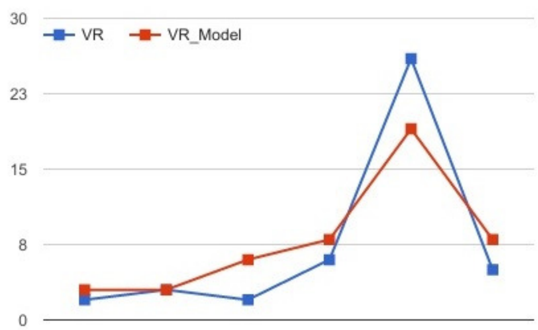
In order to compare our model with real values, we synthesized the values for both VR and VR_Model in the two tests of Chapter 5, laboratory test and field test into Table 6.5.

Table 6.5: VR model estimation and VR real values

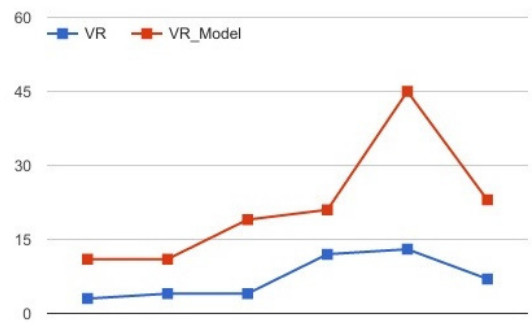
MXL	**Data Density**	**Interest**	**VR**	**VR_Model**	**Relative Dev.**
3-5	High	High	2	3	-0.33
3-5	High	Low	3	3	0
3-5	Medium	High	2	6	-0.67
3-5	Medium	Low	6	5	0.2
3-5	Low	High	26	19	0.37
3-5	Low	Low	5	8	-0.38
1-2	High	High	3	11	-0.73
1-2	High	Low	4	11	-0.64
1-2	Medium	High	4	19	-0.79
1-2	Medium	Low	14	16	-0.13
1-2	Low	High	13	45	-0.71
1-2	Low	Low	7	23	-0.70

Besides, this table shows the VR and VR_Model values categorized by the MXL, Data Density and Interest. Although the VR and VR_Model values are not identical, a similar trend, which gives an approximation of the behavior of the system when displaying images, can be observed. In addition, the *Relative Dev.*$(\frac{VR - VR\_Model}{VR\_Model})$

column shows the error deviation of the VR_Model in relation to the real VR. By observing this last column, we can say that there is less deviation for values with high MXL (3-5) compared with lower MXL (1-2).



(a) VR for High MXL(3-5)



(b) VR for Low MXL(1-2)

Figure 6-3: Values of VR and VR_model for for MXL3 to MXL5 (left) and MXL1 to MXL2 (right)

This is reflected in Figures 6-3a and 6-3b, which depict the values of the VR and VR_Model for both high MXL (3-5) and low MXL (1-2) respectively. In Figure 6-3a, it can be seen that the values of VR_Model approximately resemble the real VR obtained in the tests. By observing Figure 6-3b, we can see that the same values of VR_Model also have a trend similar to the real VR values, with the exception of the second to last value, but with more deviation. We already observed this behavior in Section 5.2, where users with low MXL achieved a greater dispersion of the VR compared with the average VR. This is due to the categorization the MXL of a user, which is a subjective task and is thus, a challenge that can be tackled in a new line of research.

Chapter 7

Conclusion and Future Work

This work is focused on the cluster display wall infrastructures and the study of their ability to act as low-cost visualization platforms for high-resolution images. In this scope, we worked on the specific infrastructure, named Liquid Galaxy, a cluster display wall developed by Google to run Google Earth. We analyzed its performance and carried out user tests on different configurations of Liquid Galaxy while running different kinds of application in order to study its possibilities and limitations. According to the results obtained in this study, we explored the feasibility of extending the use of Liquid Galaxy to a broad range of fields, such as research, gaming, entertainment or education, using commodity hardware to build the infrastructure. This implied analyzing the platform from two different sides. These were the side of the built infrastructure with the performance, scalability and heterogeneity and the user side, regarding its experience and interest. Additionally, we developed a model to approximate the performance of the platform given the main characteristics of its nodes, the connecting network and the kind of user.

To reach the aforementioned aims, we carried out the following steps, which are followed in the published papers.

1. **With regard to performance**, we analyzed the main parameters that influenced the system, namely CPU, RAM and broadband connection, in order to know its limitations for different kinds of applications. We also defined the

VR (Visualization Rate) parameter that is defined as the visualization time in relation to the total time used in each test. After studying different kinds of application in both the master-slave and client-server architectures, we concluded that almost all the studied applications could run in the Liquid Galaxy system without any problem. The Google Earth application presented higher resource consumption and, thus, we studied its performance in-depth by carrying out different tests. These were done using several locations with different time intervals between jumps and varying the amount of data to be loaded. It was observed that for this system, the VR value is mainly influenced by the environment (number of 3D buildings to be visualized) and user behavior (interval of time between consecutive displacements). This work was published in the following paper [8]:

Arroyo, I., Giné, F., Roig, C.
Analysis of the performance of the Google Earth running in a cluster display wall.
Proceedings of the 2013 International Conference on Computational and Mathematical Methods in Science and Engineering (CMMSE), Vol. 2, 24-27 June 2013, pp. 146-156.
ISBN: 978-84-616-2723-3

With the aim of extending the use of the Liquid Galaxy to a broad range of fields and also users, we introduced heterogeneity into the Liquid Galaxy platform and performed additional tests by analyzing the impact of having different kinds of node. The results showed that the slowest CPU had a negative impact on the VR, whereas the degree of heterogeneity had low influence. Thus, the heterogeneity of the cluster is not a problem when building this kind of system whenever the power of the slowest node is above a certain threshold. Likewise, we tackled the scalability problem for both types of system, the homogeneous and the heterogeneous, scaling up to 8 nodes. From this analysis, we can say that the bottleneck is located in the Internet connection, as it greatly limits the renderization speed, which is reflected in the VR values. The results of this research are reported in the following work [9]:

Arroyo, I., Giné, F., Roig, C., and Gonzalez, M.
User experience on heterogenous Liquid Galaxy cluster display walls.
Proceedings of the IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM). June 2014.
ISBN: 978-1-4799-4786-7

With the evaluation of the performance based on the elements of system configuration, we were able to verify that the Liquid Galaxy cluster, built with commodity hardware, had enough computing capacity to run and visualize any kind of application. Therefore, the first part of our hypothesis was achieved, giving us the opportunity to extend the use of the system to all kind of user and environment by using commodity hardware.

2. **With regard to usability**, we wanted to introduce the user behavior in a visualization environment to study the real performance and perception of a user of the Liquid Galaxy system. We carried out three different tests to analyze the user behavior on Liquid Galaxy. Those three tests were a *driven test*, to study the minimum acceptable value of the VR in relation to the interest and MXL (Multimedia eXperience Level) of the users; a *laboratory test*, to study the relation between the VR and the three attributes of the usability: satisfaction, efficiency and effectiveness; and a *field test* to study the user behavior in a less intrusive environment. In the driven test, we studied the time that the user spent when observing the imagery of the application when it was fully loaded. The results showed that there were two clearly distinct groups of people, which showed that users with higher skill using this kind of system or application achieved lower VR values, while users with lower skills obtained higher VR. This led us to conclude that the MXL of the user is a key factor when calculating the VR.

   In the following paper it is presented the mentioned work [4]:

Arroyo, I., Giné, F., Roig, C., and Granollers, T.
Analyzing Google Earth application in a heterogeneous commodity cluster display wall.
Multimedia Tools and Applications (MTAP) Vol 75, Issue 18, pp 1139111416. 2016
DOI: 10.1007/s11042-015-2859-z
Print ISSN: 1380-7501
Online ISSN: 1573-7721

In order to extend the analysis into a more detailed work, we also carried out a laboratory orientated test where the users had to answer a questionnaire while using an eye-tracking device and navigating in a autonomous way. The results showed that the user did not wait long enough for the imagery to load fully, as they only waited for a specific area to load. This was observed by using the eye-tracking device, which showed that the users were watching a single screen in order to complete the tasks. This specific area was the one needed to answer the question, which they answered as soon as they could, ignoring the surrounding area. That is why the VR was generally lower than we expected.

This work was published in the following paper [10].

Arroyo, I., Giné, F., Roig, C., and Granollers, T..
Usability Analysis in the Liquid Galaxy platform.
Proceedings of The Ninth International Conference on Advances in Computer-Human Interactions (ACHI), pp. 345-352. 2016
ISBN: 978-1-61208-468-8

After studying the users with constrained tests, we wanted to carry out the tests in a friendlier environment where users had freedom of choice without stress. Accordingly, we carried out a test where users could search for anything that interested them and observe it as long as they liked without any task to complete. We found that there was a correlation between the interest shown towards the test and the VR achieved, showing a higher VR when there was more interest.

The results showed that our first thoughts about the possible VR optimum value ranges were higher than the results obtained in these usability tests. Despite of this and given that there were cases where users had different experience, we could find a correlation with the system performance. With this information we could confirm that the VR was an appropriate parameter to study the usability of the system.

3. **With regard to modeling**, we wanted to model the behavior for any kind of Liquid Galaxy, taking the infrastructure and user behavior into account, in order to predict the performance of the system. Using the parameters that affected the performance and the usability of the Liquid Galaxy, we defined an equation to model the VR behavior. The parameters that we use to model the VR are the T_{CPU}, which is the time that the system is processing the imagery to be displayed, and the T_{idle}, which is the time spent by the user to observe a loaded imagery. After performing some benchmark tests, we could define some reference values for the time spent on each drawing of the imagery (T_{Draw_ref}), which, combined with the density of the imagery, can be used to calculate the T_{CPU}. By observing the T_{idle} parameter in relation to the MXL and the user interest, we could define reference values (T_{idle_ref}) to be used in the model. Note that this model takes into account the complexity and download size of the tour, which affects both the CPU and the network. It also allows users to be analyzed in the process of learning by using the MXL curve formulas described in Equation 6.6. Likewise, we validated the model accuracy by comparing the real values with the model values, achieving values with similar trends. Thus, this model can predict the behavior of the VR for different cluster configurations and different user stereotypes with a relative low error.

   The process of modeling and the study that led to these results was presented to the following paper:

   Arroyo, I., Giné, F., Roig, C., and Granollers, T.

Modeling the Performance and Usability of the Liquid Galaxy system running the Google Earth Application.
IEEE Computer Graphics and Applications.
In process of review.

After studying the possible parameters that affected the VR behavior, we could give reference tables and a mathematical model to calculate the VR using those parameters. With this model we could estimate the performance and the satisfaction of users in a given environment and hardware configuration, which enables the possibility of studying the necessities of such characteristics before building the system. Therefore, our initial hypothesis about that it was possible to model the performance of a Liquid Galaxy system in relation to a specific environment was also achieved.

7.1 Future Work

In relation to performance, we would like to extend this work to introduce features such as deciding the order in which the imagery will load. This will allow the user to decide what parts to load first in order to complete a specific objective faster.

In relation to usability, we would like to extend the usability analysis to more professional environments, such as medical consultation, which would provide a higher quality in the study of the VR. We are also interested in the study of the usability in professional environments with high stress, such as the command center in a firefighting station.

We are also interested in developing and/or analyzing both performance and usability of new applications such as multimedia applications to visualize complex 3D models and animations.

In relation to the model, we would like to extend the relationship between MXL and usability to other kinds of system, such as smart watches, which would allow the study and modeling of other systems using the VR metric.

Likewise, we would like to extend the methodology used in our analysis and, specifically, use the VR metric to measure the performance of other visualization systems, such as CAVE [24].

In the early 2017, a post in the Google Developers Blog appeared to announce that Google Earth goes open-source [39]. This opens up new possibilities when monitoring parameters that Google Earth was not designed to show at first, giving accurate statistics and new parameters to study. Also, this could mean that how the system work can be adjusted to synchronize the data and events even faster and more optimized. Another work that could be done is to change the system behavior to let the main display to be loaded first or even load the 3D imagery before loading any other texture. Moreover, other visualization engines could be used instead of the default one.

Appendix A

Appendix

A.1 VR modeling

The first approach to modeling the Liquid Galaxy by using a given hardware with a benchmark that calculates the VR in different tours is explained in this test. This would represent the calculation of the function H in Figure 6-1. Each testing tour executed in Google Earth was characterized by the average number of calls made to the Draw() function during the tour. We use the Draw() calls, given by Google Earth, because this is a representative function of the density of 3D buildings in an environment. Draw() is called every time an object has to be drawn, including the vertexes, triangles and textures. Likewise, the artificial user behavior was simulated and characterized by the frequency of jumps between the points on the tour. We obtained the VR parameter in relation to a range of Draw() calls between 100 and 600 and a frequency of jumps between 15 and 150s.

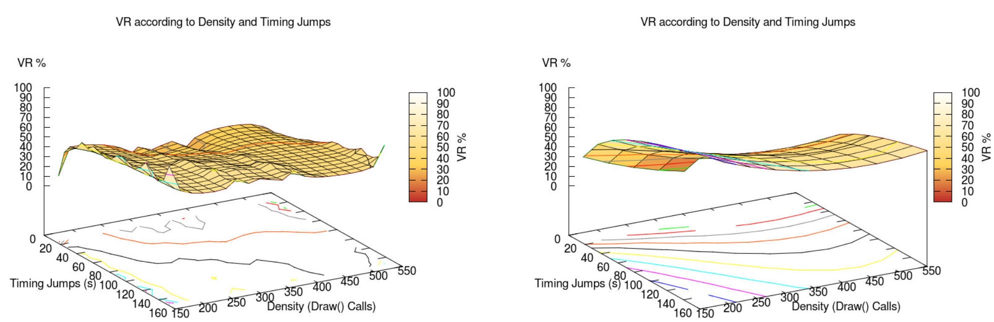
Taking different ways of modeling into account, we decided to use a linear regression because it is a simple method and already achieves a high percentage of accuracy for this model. We used the statistical tool R [77] to determine the linear regression formula that best fits the real VR values. The result of the statistical study with an accuracy of 98% gives the following formula:

$$
\begin{aligned}
VR = {} & a * draw\_calls \\
& + b * draw\_calls^2 \\
& + c * log(timing\_jump)
\end{aligned}
\tag{A.1}
$$

where *draw_calls* and *timing_jump* are the variables, and a, b and c are the constants calculated by R and have the following values:

$$
\begin{aligned}
a &= -3.985e - 01 \\
b &= 4.357e - 04 \\
c &= 3.056e + 01
\end{aligned}
\tag{A.2}
$$

With this formula, we can estimate values of the VR for the hardware studied.



(a) 3D graph of the data obtained (b) 3D graph of the estimation

Figure A-1: 3D representations of the real and estimated VR

Figure A-1 shows the VR value that was obtained by varying the density of the tours (number of Draw() Calls) and the period between timing jumps (seconds). This is shown for the real data obtained from the tests (left graph) and the estimated VR derived from equation A.1 (right graph). As can be observed, in both graphs, the VR value decreases as the density of the tours increases and, on the contrary, the increase of the timing jump causes an increase in the VR value. The graph on the left also shows some peaks that correspond to real calculated values. Regarding the estimated VR values, on the right graph, it can be observed that they follow the same trend as the real VR values. As could be expected, this graph presents a smoother pattern, without peaks, as it corresponds to the simulated process. However, even so, the trend is highly accurate and reliable enough to be used as a prediction mechanism for the VR values that could be expected for a given hardware.

This work tries to find a relation between the performance and simulated user behavior by using the data density and the timing jumps, respectively. After testing this model, we found that the error range was higher than our expectations, which lead us to think that there were additional parameters to be used in the modeling of the VR. Likewise, we discarded this model for a more complex and accurate one, described in this thesis.

Bibliography

[1] 3DConnexion. Space Navigator. http://www.3dconnexion.es/products/spacemouse/spacenavigator.html. retrieved at 2016-05-27.

[2] Christopher Andrews, Alex Endert, and Chris North. Space to think: Large high-resolution displays for sensemaking. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '10, pages 55–64, New York, NY, USA, 2010. ACM.

[3] L Ardevol. *User experience methodology for the design and evaluation of interactive systems*. PhD thesis, Ph. D. dissertation, University of Lleida, 2013.

[4] I. Arroyo, F. Giné, C. Roig, and T. Granollers. Analyzing Google Earth application in a heterogeneous commodity cluster display wall. *Multimedia Tools and Applications*, pages 1–26, 2015.

[5] Ismael Arroyo. System, network and caching performance benchmarking. https://www.google-melange.com/archive/gsoc/2013/orgs/lg/projects/asherat.html, 2013. retrieved at 2016-07-11.

[6] Ismael Arroyo. System, network, and performance monitoring for Liquid Galaxy web-based app. https://www.google-melange.com/archive/gsoc/2014/orgs/lg/projects/asherat.html, 2014. retrieved at 2016-07-11.

[7] Ismael Arroyo. Cloud point visualization for Liquid Galaxy. https://summerofcode.withgoogle.com/projects/#6613839659401216, 2016. retrieved at 2016-07-11.

[8] Ismael Arroyo, Francesc Giné, and Concepció Roig. Analysis of the performance of the Google Earth running in a cluster display wall. In *Proceedings of the 2013 International Conference on Computational and Mathematical Methods in Science and Engineering (CMMSE)*, volume 2, pages 146–156. IEEE, 2013.

[9] Ismael Arroyo, Francesc Giné, Concepció Roig, and Marc Gonzalez. User experience on heterogenous Liquid Galaxy cluster display walls. In *2014 IEEE 15th International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, pages 1–3. IEEE, 2014.

[10] Ismael Arroyo, Francesc Giné, Concepció Roig, and Toni Granollers. Usability analysis in the Liquid Galaxy platform. In *Proceedings of The Ninth International Conference on Advances in Computer-Human Interactions (ACHI 2016)*, pages 345–352. IARIA, 2016.

[11] Robert Ball and Chris North. Effects of tiled high-resolution display on basic visualization and navigation tasks. In *CHI'05 extended abstracts on Human factors in computing systems*, pages 1196–1199. ACM, 2005.

[12] Ignasi Barri, Concepció Roig, Francesc Giné, and Francesc Solsona. Mapping mmofps over heterogeneous distributed systems. *The Journal of Supercomputing*, 58(3):341–348, 2011.

[13] Xiaojun Bi and Ravin Balakrishnan. Comparing usage of a large high-resolution display to single or dual desktop displays for daily work. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1005–1014. ACM, 2009.

[14] Nanette J Boden, Danny Cohen, Robert E Felderman, Alan E Kulawik, Charles L Seitz, Jakov N Seizovic, and Wen-King Su. Myrinet: A gigabit-per-second local area network. *IEEE micro*, 15(1):29–36, 1995.

[15] John Brooke. Sus: a retrospective. *Journal of Usability Studies*, 8(2):29–40, 2013.

[16] William Buxton, George Fitzmaurice, Ravin Balakrishnan, and Gordon Kurtenbach. Large displays in automotive design. *Computer Graphics and Applications, IEEE*, 20(4):68–75, 2000.

[17] Y. Chen, H. Chen, D.W Clark, Z. Liu, G. Wallace, and K. Li. Software environments for cluster-based display systems. In *Proceedings - 1st IEEE/ACM International Symposium on Cluster Computing and the Grid*, CCGrid 2001, pages 202–210. IEEE, 2001.

[18] Haeyong Chung, Christopher Andrews, and Chris North. A survey of software frameworks for cluster-based large high-resolution displays. *IEEE Transactions on Visualization and Computer Graphics*, 20(8):1158–1177, 2014.

[19] Gerald Combs. Wireshark 2.0. `https://www.wireshark.org/`, 2015. retrieved at 2017-03-10.

[20] EndPoint Corp. Videogames for liquid galaxy. `https://github.com/LiquidGalaxy/liquid-galaxy/wiki/VideoGames`. retrieved at 2016-07-11.

[21] Wagner T Corrêa, James T Klosowski, and Cláudio T Silva. Out-of-core sort-first parallel rendering for cluster-based tiled displays. *Parallel Computing*, 29(3):325–338, 2003.

[22] Cyviz. Vizwall. `http://www.cyviz.com/`. retrieved at 2016-05-27.

[23] Ryan Lienhart Dahl. Node.js. `https://nodejs.org/`, 2009. retrieved at 2015-06-07.

[24] Thomas DeFanti, Daniel Acevedo, Richard A Ainsworth, Maxine D Brown, Steven Cutchin, Gregory Dawe, Kai-Uwe Doerr, Andrew Johnson, Chris Knox, Robert Kooima, et al. The future of the CAVE. *Central European Journal of Engineering*, 1(1):16–37, 2011.

[25] Thomas A DeFanti, Gregory Dawe, Daniel J Sandin, Jurgen P Schulze, Peter Otto, Javier Girado, Falko Kuester, Larry Smarr, and Ramesh Rao. The star-cave, a third-generation cave and virtual reality optiportal. *Future Generation Computer Systems*, 25(2):169–178, 2009.

[26] K. Doerr and Falko Kuester. CGLX: a scalable, high-performance visualization framework for networked display environments. *Visualization and Computer Graphics, IEEE Transactions on*, 17(3):320–332, 2011.

[27] Stefan Eilemann, Maxim Makhinya, and Renato Pajarola. Equalizer: A scalable parallel rendering framework. *Visualization and Computer Graphics, IEEE Transactions on*, 15(3):436–452, 2009.

[28] End Point Corporation. Panoramic Video Streaming. `https://code.google.com/p/liquid-galaxy/\\wiki/PanoramicVideo`, 2012. retrieved at 2015-06-07.

[29] End Point Corporation. Liquid Galaxy by End Point. `http://liquidgalaxy.endpoint.com`, 2013. retrieved at 2015-06-07.

[30] Alex Endert, Christopher Andrews, Yueh Hua Lee, and Chris North. Visual encodings that support physical navigation on large displays. In *Proceedings of Graphics Interface 2011*, GI '11, pages 103–110, School of Computer Science, University of Waterloo, Waterloo, Ontario, Canada, 2011.

[31] Robert E Felderman, Alan E Kulawik, Charles L Seitz, J Seizovic, Wen-King Su, et al. Myrinet: A gigabit-per-second local area network. *IEEE micro*, (February):29–36, 1995.

[32] Andrew Forsberg, Graff Haley, Andrew Bragdon, Joseph Levy, Caleb I Fassett, David Shean, James W Head III, Sarah Milkovich, Mark Duchaineau, et al. Adviser: immersive field work for planetary geoscientists. *IEEE computer graphics and applications*, (4):46–54, 2006.

[33] AS Forsberg, JW Head, N Petro, G Morgan, et al. A 3d geoscience data visualization system for mars applied to undergraduate laboratories. In *Lunar and Planetary Science Conference*, volume 38, page 1297, 2007.

[34] Google. Google summer of code. `https://summerofcode.withgoogle.com/`, 2005. retrieved at 2016-07-11.

[35] Google. Google Earth. http://www.google.com/earth/index.html, 2013. retrieved at 2015-06-07.

[36] Google. Peruse-a-rue for liquid galaxy website. http://blog.endpoint.com/2013/11/liquid-galaxy-and-its-very-own-street.html, 2013. retrieved at 2015-06-07.

[37] Google. Liquid galaxy website. http://www.google.com/earth/explore/showcase/liquidgalaxy.html, 2015. retrieved at 2015-06-07.

[38] Google. Tobii website. http://www.tobii.com/es/eye-tracking-research/global/products/hardware/tobii-glasses-eye-tracker, 2015. retrieved at 2015-07-08.

[39] Google. Open-sourcing google earth enterprise, 2017. retrieved at 2017-05-02.

[40] Silicon Graphics. OpenGL 4.5. https://www.opengl.org/, 2014. retrieved at 2017-03-10.

[41] Greggman and Human Engines. Aquarium WebGL. http://webglsamples.org/aquarium/aquarium.html, 2009. retrieved at 2015-06-07.

[42] Naoki Hashimoto, Seungzoo Jeong, Yasutoyo Takeyama, and Makoto Sato. Immersive multi-projector display on hybrid screens with human-scale haptic and locomotion interfaces. In *Cyberworlds, 2004 International Conference on*, pages 361–368. IEEE, 2004.

[43] M. Hassenzahl and N. Tractinsky. User experience - a research agenda. *Behavior and Information Technology*, 25(2):91–97, 2006.

[44] Kip Haynes and Eric Chance. CaveSL: A Large Format Scalable Multi-display System for Social and Scientific Visualization in Second Life. In *13th Annual IEEE EuroGraphics/Visualization (EuroVis) Conference*, Bergen, Norway, June 2011.

[45] JW Head, A Van Dam, SG Fulcomer, A Forsberg, and Rosser Prabhat. G., and milkovich, s.(2005). adviser: Immersive scientific visualization applied to mars research and exploration. *Photogrammetric Engineering & Remote Sensing*, 71:1219–1225.

[46] Thomas T. Hewett, Ronald Baecker, Stuart Card, Tom Carey, Jean Gasen, Marilyn Mantei, Gary Perlman, Gary Strong, and William Verplank. ACM SIGCHI curricula for human-computer interaction. Technical report, New York, NY, USA, 1992.

[47] G Huisman and M Van Hout. The development of a graphical emotion measurement instrument using caricatured expressions: the lemtool. In *Emotion in HCI–Designing for People. Proceedings of the 2008 International Workshop*, pages 5–8. Citeseer, 2008.

[48] Greg Humphreys, Ian Buck, Matthew Eldridge, and Pat Hanrahan. Distributed rendering for scalable displays. In *Supercomputing '00 Proceedings of the 2000 IEEE conference on Supercomputing (CDROM) Article No. 30*, 2000.

[49] Greg Humphreys, Matthew Eldridge, Ian Buck, Gordan Stoll, Matthew Everett, and Pat Hanrahan. Wiregl: a scalable graphics system for clusters. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 129–140. ACM, 2001.

[50] Greg Humphreys, Mike Houston, Ren Ng, Randall Frank, Sean Ahern, Peter D Kirchner, and James T Klosowski. Chromium: a stream-processing framework for interactive rendering on clusters. In *ACM Transactions on Graphics (TOG)*, volume 21, pages 693–702. ACM, 2002.

[51] Tommi Ilmonen, Markku Reunanen, and Petteri Kontio. Broadcast gl: An alternative method for distributing opengl api calls to multiple rendering slaves. *JOURNAL OF WSCG*, 13(2):65–72, 2005.

[52] Yoshio Ishiguro and Jun Rekimoto. Peripheral vision annotation: Noninterference information presentation method for mobile augmented reality. In *Proceedings of the 2Nd Augmented Human International Conference*, AH '11, pages 8:1–8:5. ACM, 2011.

[53] ISO DIS. Ergonomic requirements for office work with visual display terminals (VDTs)-Part 11-Guidance on usability. ISO DIS 9241-11:1998, International Organization for Standardization, Geneva, Switzerland, 1998.

[54] ISO DIS. Ergonomics of human system interaction Part 210: Human-centered design for interactive systems (formerly known as 13407). ISO DIS 9241-210:2008, International Organization for Standardization, Geneva, Switzerland, 2008.

[55] ISO/IEC. Systems and software engineering-Systems and software Quality Requirements and Evaluation (SQuaRE) System and software quality models. ISO/IEC 25010:2011, International Organization for Standardization, Geneva, Switzerland, 2011.

[56] Shahram Izadi, Harry Brignull, Tom Rodden, Yvonne Rogers, and Mia Underwood. Dynamo: a public interactive surface supporting the cooperative sharing and exchange of media. In *Proceedings of the 16th annual ACM symposium on User interface software and technology*, pages 159–168. ACM, 2003.

[57] Andrew Johnson, Jason Leigh, Paul Morin, and Peter Van Keken. Geowall: Stereoscopic visualization for geoscience research and education. *Computer Graphics and Applications, IEEE*, 26(6):10–14, 2006.

[58] Sung-Jin Kim. *The diva architecture and a global timestamp-based approach for high-performance visualization on large display walls and realization of high*

quality-of-service collaboration environments. California State University at Long Beach, 2006.

[59] Naveen K Krishnaprasad, Venka Vishwanath, Shalini Venkataraman, Arun G Rao, Luc Renambot, Jason Leigh, Andrew E Johnson, and Brian Davis. Juxtaview-a tool for interactive visualization of large imagery on scalable tiled displays. In *Cluster Computing, 2004 IEEE International Conference on*, pages 411–420. IEEE, 2004.

[60] S. Krug. *Rocket Surgery Made Easy: The Do-It-Yourself Guide to Finding and Fixing Usability Problems*. New Riders., 2009.

[61] J. Leigh, A. Johnson, L. Renambot, T. Peterka, and et al. Scalable resolution display walls. *Proceedings of the IEEE*, 101(1):115–129, 2013.

[62] Inc Linden Research. Second Life. `http://secondlife.com//`. retrieved at 2016-05-27.

[63] Can Liu, Olivier Chapuis, Michel Beaudouin-Lafon, Eric Lecolinet, and Wendy E Mackay. Effects of display size and navigation type on a classification task. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 4147–4156. ACM, 2014.

[64] G. Lui-Kwan. In-flight entertainment: the sky's the limit. *Computer*, 33(10):98–101, Oct 2000.

[65] NASA. NASA Liquid Galaxy. `https://open.nasa.gov/blog/nasas-liquid-galaxy-an-overview`, 2011. retrieved at 2016-05-27.

[66] B. Neal, P. Hunkin, and A. McGregor. Distributed opengl rendering in network bandwidth constrained environments. *Eurographics Symposium on Parallel Graphics and Visualization*, 2011.

[67] Tao Ni, Greg S Schmidt, Oliver G Staadt, Mark A Livingston, Robert Ball, and Richard May. A survey of large high-resolution display technologies, techniques, and applications. In *Virtual Reality Conference, 2006*, pages 223–236. IEEE, 2006.

[68] J. Nielsen. *Usability Engineering*. Morgan Kaufmann Publishers Inc., 1993.

[69] Jakob Nielsen. Guerrilla HCI: Using discount usability engineering to penetrate the intimidation barrier. *Cost-justifying usability*, pages 245–272, 1994.

[70] Jakob Nielsen and Kara Pernice. *Eyetracking web usability*. New Riders, 2010.

[71] Nirnimesh, P. Harish, and P. Naraayanan. Garuda: A scalable tiled display wall using commodity pcs. *IEEE Transactions on Visualization and Computer Graphics.*, 13(5):864–877, 2007.

[72] OpenArena Team. OpenArena (Quake 3 Arena). `http://www.openarena.ws`, 2005. retrieved at 2014-12-01.

[73] OpenGL. Opengl: Drawing polygons. `https://open.gl/drawing`, 2012. retrieved at 2016-12-27.

[74] OSG. Openscene graph, 2014. retrieved at 2015-06-07.

[75] Ponent 2002. G-liquid galaxy. `http://www.g-liquidgalaxy.com`, 2012. retrieved at 2015-06-07.

[76] Alex Poole and Linden J Ball. Eye tracking in HCI and usability research. *Encyclopedia of human computer interaction*, 1:211–219, 2006.

[77] R Team. R project. `http://www.r-project.org`, 1997. retrieved at 2015-06-07.

[78] Luc Renambot, Arun Rao, Rajvikram Singh, Byungil Jeong, Naveen Krishnaprasad, Venkatram Vishwanath, Vaidya Chandrasekhar, Nicholas Schwarz, Allan Spale, Charles Zhang, et al. SAGE: the Scalable Adaptive Graphics Environment. In *Proceedings of WACE*, volume 9, pages 2004–09. Citeseer, 2004.

[79] Nicholas Schwarz. *Distributed Volume Rendering of Very Large Data on High-Resolution Scalable Displays*. 2007.

[80] Nicholas Schwarz and Jason Leigh. Distributed volume rendering for scalable high-resolution display arrays. In *GRAPP*, pages 211–218, 2010.

[81] Rajvikram Singh, Nicholas Schwarz, Nut Taesombut, David Lee, Byungil Jeong, Luc Renambot, Abel W Lin, Ruth West, Hiromu Otsuka, Sei Naito, et al. Real-time multi-scale brain data acquisition, assembly, and analysis using an end-to-end optiputer. *Future Generation Computer Systems*, 22(8):1032–1039, 2006.

[82] Squid Team. Squid web proxy cache. `http://www.squid-cache.org`, 2002. retrieved at 2015-06-07.

[83] O. G. Staadt, J. Walker, C. Nuber, and B. Haman. A survey and performance analysis of software platforms for interactive cluster-based multi-screen rendering. *Eurographics Workshop on Virtual Environments*, pages 261–270, 2003.

[84] Roger Stevens. Testing the norad command and control system. *IEEE Transactions on Systems Science and Cybernetics*, 1(4):47–51, 1968.

[85] N Taesombot, Frank Uyeda, Andrew A Chien, Larry Smarr, Thomas A DeFanti, Phil Papadopoulos, Jason Leigh, Mark Ellisman, and John Orcutt. The optiputer: high-performance, qos-guaranteed network service for emerging e-science applications. *IEEE Communications Magazine*, 44(5):38–45, 2006.

[86] Desney S. Tan, Jeanine K. Stefanucci, Dennis R. Proffitt, and Randy Pausch. The infocockpit: Providing location and place to aid human memory. In *Proceedings of the 2001 Workshop on Perceptive User Interfaces*, PUI '01, pages 1–4, New York, NY, USA, 2001. ACM.

[87] Hardinfo tools. `https://help.ubuntu.com/community/HardInfo`. retrieved at 2016-07-21.

[88] Waikato University. The symphony cluster. `http://symphony.waikato.ac.nz/`, 2011. retrieved at 2015-06-07.

[89] Wouter van Oortmerssen. Cube 2: Sauerbraten. `http://sauerbraten.org//`. retrieved at 2016-05-27.

[90] VideoLAN. VLC Media Player. `http://www.videolan.org/vlc/`. retrieved at 2016-05-27.

[91] Grant Wallace, Otto J Anshus, Peng Bi, Han Chen, Yuqun Chen, Douglas Clark, Perry Cook, Adam Finkelstein, Thomas Funkhouser, Anoop Gupta, et al. Tools and applications for large-scale display walls. *Computer Graphics and Applications, IEEE*, 25(4):24–33, 2005.

[92] Webgl applications. `https://webglsamples.org`. retrieved at 2015-06-07.

[93] Bin Wei, Claudio Silva, Eleftherios Koutsofios, Shankar Krishnan, and Stephen North. Visualization research with large displays. *IEEE Computer Graphics and Applications*, 20(4):50–54, 2000.

[94] Wikipedia. 4k Resolution. `https://en.wikipedia.org/wiki/4K_resolution`. retrieved at 2016-05-27.

[95] UWS Wonderama. Opengl: Textures. `https://open.gl/textures`, 2012. retrieved at 2016-12-27.

[96] UWS Wonderama. X-wing fly-by on liquid galaxy. `https://www.youtube.com/watch?v=cbTQPJtqLxg`, 2012. retrieved at 2016-07-11.

[97] Beth Yost, Yonca Haciahmetoglu, and Chris North. Beyond visual acuity: The perceptual scalability of information visualizations for large displays. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '07, pages 101–110, New York, NY, USA, 2007. ACM.

[98] Zygote. ZygoteBody. `http://www.google.com/earth/index.html`, 2010. retrieved at 2015-06-07.