

Semester: MED 10

Title: A study on the perceived quality of smoke effects in virtual environments, using fluid systems and imposters.

Project Period: Spring 2011

Semester Theme: Master thesis

Supervisor: Rama Hoetzlein.



Aalborg University Copenhagen
Lautrupvang 15, 2750 Ballerup, Denmark

Semester coordinator: Stefania Serafin
Secretary: Lisbeth Nykjær
Phone: 9940 2471
lny@create.aau.dk
<https://internal.media.aau.dk/>

Created by:

Morten Flyvholm Iversen

Copies: 3

Pages: 75

Finished: October 3rd - 2011

Abstract

This report is based on a set of concerns regarding the limitations in performance of home computers compared to the endless possibilities of modern visual effects, whose stunning looks often result in heavy stress on the user's hardware. This inspired an analysis where alternative methods for portraying effects, and various methods for additionally reducing the amount of effect variables were researched. A technique was put together to attempt to create and portray a smoke effect with the use of billboarding, and a limited amount of possible viewing angles, so that the potential stress on the hardware was minimized, without the user noticing any difference in the perceived quality of the effect. Further research was initiated by looking into previous attempts to control the perceived quality of effects, and also on various methods for the actual creation of effects. Two different smoke effects were created, and each was rendered and tested in three versions, one viewed from all angles, one viewed from every second angle and one viewed from every fifth angle. The purpose was to attempt to conclude that effects could be perceived of similar quality when viewed from fewer angles, compared to being viewed from all angles. A number of test subjects watched and ranked the various examples and rated the quality in a quantitative questionnaire composed of a mixture between subjective and objective questions. It was concluded that the ability to reduce visual quality and still maintain the same perceived quality can be done, however it did not work with all the possible solutions. Additionally, limitations to the method were found, as once the test subjects watched an effect which crossed their threshold, the illusion was again broken and the quality perceived as worse.

1 Content

4 INTRODUCTION & MOTIVATION	3
4.1.1 Project Angle	3
5 PRE-ANALYSIS	4
5.1 The Problem	4
5.1.1 My Thesis Statement.....	4
5.1.1.1 Hypothesis.....	5
5.2 Methods for Creating and Portraying Visual Effects	6
5.2.1 Real 3D Rendering.....	6
5.2.2 Image-Based Rendering.....	6
5.2.2.1 Skyboxes	7
5.2.2.2 Sprites	9
5.2.2.3 Billboards	10
5.2.2.4 Imposters.....	13
5.3 Related Studies	14
5.3.1 A Three Dimensional Image Cache for Virtual Reality	14
5.3.2 The use of Imposters in Interactive 3D Graphics Systems	14
5.3.3 Animated Impostors for Real-time Display of Numerous Virtual Humans.....	15
5.3.4 Real-Time Cloud Rendering	16
5.3.5 Real-Time Tree Rendering.....	16
5.3.6 Spherical Billboards and their Application to Rendering Explosions.....	17
5.3.7 Additional Inspiration.....	17
5.4 Changeable Elements	17
5.4.1 Shadows	18
5.4.2 Movement/Viewport.....	19
5.4.3 Compression/Level of Detail	19
5.4.4 Selection.....	19
5.4.5 Elements Not Included	19
5.5 Delimitation	20
5.6 Final Problem	21
5.7 Hypothesis.....	21
5.8 Target Group.....	21
6 ANALYSIS	22
6.1 Research on Fluid Simulation.....	22
6.1.1 Stable Fluids	22

6.1.2 Visual Simulation of Smoke	22
6.1.3 Simulation and Animation of Fire and Other Natural Phenomena in the Visual Effects Industry	23
6.1.4 Interactive Fluid-Particle Simulation using Translating Eulerian Grids	23
6.1.5 Limitations Still Exist.....	23
6.2 Simulation Techniques	24
6.3 Fluid Algorithms.....	24
6.3.1 Navier-Stokes	25
6.3.2 Solvers.....	25
6.3.3 Summary	26
6.4 Software Options.....	26
6.4.1 Autodesk – Maya	27
6.4.2 Alternatives.....	27
6.5 Billboards and Fluids Combined	27
6.5.1 Billboard Clusters.....	27
6.5.2 Particle Billboards	28
6.5.3 Fully Rendered Billboards	29
7 TEST METHODOLOGY	30
7.1 Definitions	30
7.1.1 Quality	30
7.1.1.1 Side Note	31
7.1.1.2 Subjective Feelings or Objective Facts.....	31
7.2 Subjective Test	32
7.3 Objective Test.....	32
8 DESIGN	33
8.1 Reference Material	33
8.1.1 Reference One.....	33
8.1.2 Reference Two.....	34
8.1.3 Reference Three.....	35
8.2 Smoke	36
8.3 Surroundings	37
8.4 Camera	37
8.5 Rendering	37
8.6 Summary	38

9 IMPLEMENTATION	39
9.1 Smoke.....	39
9.1.1 Emission, Colour and Opacity.....	39
9.1.2 Movement.....	41
9.1.3 Wind.....	42
9.1.4 Smoke Detail.....	42
9.1.4.1 Resolution.....	42
9.1.4.2 Texture.....	43
9.1.4.3 Complications	44
9.2 Environment.....	44
9.2.1 Surroundings.....	44
9.2.2 Lights	45
9.2.3 Camera	45
9.3 Rendering	47
9.3.1 Complications	47
9.4 Compositing	48
9.5 Time	49
9.6 Optimizations	49
10 TEST	50
10.1 Clarification.....	50
10.2 Procedure and Design	50
10.2.1 Design	50
10.2.1.1 Video	50
10.2.1.2 Questionnaire.....	52
10.2.2 Test 1 – The <i>Fan</i> Test.....	52
10.2.3 Test 2 – The <i>Cone</i> Test.....	53
10.3 Observations and Results.....	54
10.3.1 Test 1 – The <i>Fan</i> Test.....	54
10.3.2 Test 2 – The <i>Cone</i> Test.....	55
10.3.3 Comparison	56
10.3.4 Discussion	57
10.3.4.1 Context and Focus.....	57
10.3.4.2 Design	57
11 CONCLUSION.....	58
12 DISCUSSION	59

13 PERSPECTIVE.....60

14 FUTURE DEVELOPMENT61

15 LIST OF FIGURES62

16 REFERENCES.....64

16.1 Bibliography..... 64

17 APPENDIX.....68

17.1 Appendix 1: Questionnaire 68

2 Abstract

This report is based on a set of concerns regarding the limitations in performance of home computers compared to the endless possibilities of modern visual effects, whose stunning looks often result in heavy stress on the user's hardware. This inspired an analysis where alternative methods for portraying effects, and various methods for additionally reducing the amount of effect variables were researched. A technique was put together to attempt to create and portray a smoke effect with the use of billboarding, and a limited amount of possible viewing angles, so that the potential stress on the hardware was minimized, without the user noticing any difference in the perceived quality of the effect. Further research was initiated by looking into previous attempts to control the perceived quality of effects, and also on various methods for the actual creation of effects. Two different smoke effects were created, and each was rendered and tested in three versions, one viewed from all angles, one viewed from every second angle and one viewed from every fifth angle. The purpose was to attempt to conclude that effects could be perceived of similar quality when viewed from fewer angles, compared to being viewed from all angles. A number of test subjects watched and ranked the various examples and rated the quality in a quantitative questionnaire composed of a mixture between subjective and objective questions. It was concluded that the ability to reduce visual quality and still maintain the same perceived quality can be done, however it did not work with all the possible solutions. Additionally, limitations to the method were found, as once the test subjects watched an effect which crossed their threshold, the illusion was again broken and the quality perceived as worse.

3 Readers Guide

It is advised that the reader has followed a course in image processing or 3D animation at Aalborg University, or at least a similar course at another research facility. As well as the reader should have a small amount of knowledge about video games and terms used within that field.

Keywords: Visual Effects, Fluids, Maya, Dynamics, Smoke, Fire, Billboards, Imposters, Image-based Rendering.

4 Introduction & Motivation

The inspiration and motivation for this project comes from different goals set by hardware producers and video game developers. The graphical quality of video games are increasing rapidly, and as a result of that, hardware producers are constantly striving to release new products that contain the newest technology, thus enabling the users to draw full benefit of these new graphical wonders. However, always being up to date, and never having to settle with settings below the video games absolute maximum is a rather expensive hobby, this i have experienced myself first hand.

Game producers have a large variety of methods for creating graphical spectacles in their productions. They also possess a lot of methods for scaling these graphical spectacles, either showing them in their fullest quality, in a decreased quality or as an illusion which mimics some of the graphical spectacles but in reality is completely different, dependant on the capabilities of the user's game station.

One of the newer features which have been added to games is the ability to process an effect with a real-time renderer. This means that the effect is not created beforehand but is processed directly when it happens, thus being more unique and responsive to the game environment, but also a lot heavier on the hardware of the game station. An interesting question arises: is a real-time rendering of an effect really that much more desirable than one created beforehand compared to the added stress to the game station which easily can result in a drop in frame rate within the game? I am an eager consumer of video games myself, and often these issues appear to result in a lot more than just graphical annoyance. Other results such as personal irritation, game stations overheating or drop in user performance within the game are all possible consequences of wrong graphical settings. This happens because some users simply swear only to use the best settings provided within the game, without accepting the limitations of their game station. This is where the problem manifests itself, some users simply does not accept their limitations, and I do not believe it is of any interest to hardware producers or game developers to limit their products, only to accommodate users which does not correctly acknowledge their computers limits.

I will in this project attempt to highlight what technologies within games that are currently pushing the boundaries of both the user's perception and also the hardware producers. What effects are most riveting and stunning to the user, and what effects are easily replaced with something more acceptable towards older hardware without the user actually noticing? After finding the most optimal effect to recreate and perform a test on, I will try to recreate a scenario where multiple versions of the same effect will occur, and then let a group of users experience this variation and observe their reaction. This observation will be the foundation of a statistical analysis between various graphical effects and their fallout towards the user, and finally answering the question; are the newest graphical settings really a necessity for the user, or are they simply only being used because the user is aware of their availability, which in the end is making the user handicap him or herself as a result of this.

4.1.1 Project Angle

In this project it has been chosen that the analytical angle will not be that of a hardware tester, performance analyst, game critic or anything of the likes. Those are not in the interest of the problem, and analysis will solely be performed with the angle of the user's perception towards video game graphics and effects.

5 Pre-Analysis

5.1 The Problem

In modern computer graphics both in movies and in games, there exist various ways of creating and portraying effects to the viewer, all which have strengths and weaknesses that makes them more or less appealing towards the various platforms in which they can be used. One of the major issues which separate these methods is the computers performance capabilities. Some techniques can provide stunning effects, but result in a very heavy load on the hardware used to produce it.

Some very common effects in both games and movies are effects such as fire and smoke. This is however where the problematic starts to arise. Smoke and fire are often used very extensively, and thus becomes an expensive thing to produce, both economically for when creating real fire and smoke in movies, but perhaps more interestingly when used as a visual effect in both movies and games. There exist solutions such as particle and fluids systems which can simulate very realistic replications of these effects, but they are computationally heavy and often not suitable for real-time rendering. So the question is; are those systems the best solution to create these effects, or are there any solid alternatives? Movies like Lord of the Rings 1-3, (1) which are as of 2011 still considered fairly state of the art, are known to use other options such as imposters which mimic real 3D effects, but on the contrary are consisting of simple 2D images which are much easier to render and produce. See section 5.2.2.4 for an explanation of imposters. They do this even though they are not running them real-time. So why do they do that, are real 3D effects not more desirable in every situation, especially when they are within your funds and hardware range?

5.1.1 My Thesis Statement

As I have stated earlier, my concerns are these:

"Is a real-time rendering of an effect really that much more desirable than one created beforehand compared to the added stress to the game station, which easily can result in a drop in frame rate within the game?"

This led to the following thesis statement:

To what degree can the creation of virtual effects such as smoke or fire be simplified while still maintaining the same graphical satisfaction of the viewer?

It has been chosen to narrow the problem down to effects containing either smoke or fire at this very early stage, as the area covering computer effects is very large. Nevertheless, fire and smoke are used often and a lot, which makes both an attractive effect, but also an effect which in theory, is easier to find information about.

As Akenine-Möller et al. explains in their book *Real-Time Rendering*:

"There is no single correct way to render a scene. Each rendering method is an approximation of reality, at least if that is the goal." (2)

This is important to keep in mind. Some methods might provide a higher possibility of creating realism or perhaps quality, but that does not make the method more correct. A selection of methods for creating effects will be briefly explained in Section 5.2.2.

5.1.1.1 Hypothesis

As a supplementing basis for this analysis, it is the belief that certain effects can within certain boundaries be changed to a similar effect that uses a more hardware tolerant method, and still provide the same amount of perceived quality for the viewer.

5.2 Methods for Creating and Portraying Visual Effects

There are a few different ways of creating realistic effects, followed by a few different ways of portraying them inside a production. The two main methods may be categorized as real 3D renders, and image-based renders. In this section these ways of creating effects will be examined, along with how they can be portrayed inside the virtual environment.

5.2.1 Real 3D Rendering

The most obvious way to create effects such as fire, smoke and the likes are through fluid or particle systems which have inbuilt mathematical formulas that tells the fluid or particles how to behave, so that is mimics the real world effects as much as possible. This is done so that the effects are created in a live 3D render, making it a very versatile use - and enables the fluid or particle system to interact easily with other objects or forces within the scene.

The benefit of full simulation is that the systems are very dynamic, responsive and adaptive towards the environment that they are in, and the forces which they are affected by, if any. Another positive element of these types of renders are that they can be viewed from any angle or orientation and still look credible and maintain the same quality. The downside of this method is that it can be computationally heavy on the hardware. If the systems become too complex or too detailed, the computation time becomes proportionally higher, making the render and simulation time of the simulation very long. This is especially a problem for games where these effects have to be rendered in real-time. *Akenine-Möller et al.* also describe this, though they talk about polygon models and not particle or fluid systems; however the observation is the same. They write:

“Modelling surfaces with polygons is often the most straightforward way to approach the problem of portraying objects in a scene. Polygons are good up to a point, however. Image-based rendering (IBR) has become a paradigm of its own. As its name proclaims, images are the primary data used for this type of rendering. A great advantage of representing an object with an image is that the rendering cost is proportional to the number of pixels rendered, and not to, say, the number of vertices in a geometrical model.” (2)

In what they write here the number of polygons and vertices can be compared to the amount of fluid or particles released in a system. The description here is actually a possible solution to the problematic which is described in the introduction of this report, or more accurately, a way to minimize the problem by using alternative methods of portraying these effects. One possible method of doing this is what is referred to as image-based rendering (IBR).

5.2.2 Image-Based Rendering

Heung-Yeung Shum et al. writes as the introduction for one of their reposts:

“Image-based rendering (IBR) refers to a collection of techniques and representations that allow 3D scenes and objects to be visualized in a realistic way without full 3D model reconstruction. IBR uses images as the primary substrate.” (3)

This description is a concise overview of what IBR is about in general. There exist a vast amount of methods within the boundaries of IBR, but the overall idea is that objects are replaced with images, in one form or another. These images then imitate either real world effects or 3D effects, but portray them on a 2D image.



Figure 1 – A screenshot from *Doom 2* by *id Software*, copyright © 1994, showing an early usage of image-based rendering, which is portraying a character and two fire effects. (4)

Specific methods within this category can be skyboxes, sprites, billboards and imposters. All of these are more or less the same, but with small variations of their functionality. In the following section a small walkthrough of their functionality will be described in order to determine the relevance of each method to the solution of this projects problem. Figure 1 shows a screenshot from the old game *Doom 2* released by *id Software*. (5) The character in the centre, and the two fire effects (foreground and background) are all created with the use of image-based rendering. These fire effects are fairly simple, and can be considered to be poor quality now, but at the time of their release, there were different criteria and the effects were considered to be fair in quality.

5.2.2.1 Skyboxes

Skyboxes are not as much a method for displaying effects as it is a method for portraying an overall environment, which may be implemented with environment maps. Skyboxes take advantage of distant objects lack of movement when the viewer is moving around. *Akenine-Möller et al.* describes a scenario in their book based on mountains being viewed from a distance:

"... A distant mountain itself does not normally look appreciably different if you move a meter, or even a thousand meters. It may be blocked from view by nearby objects as you move, but take away those objects and the mountain itself looks the same." (2)

Environment maps can be used for various things, but the focus here is on its ability to act as the environment of a scene by using a texture mapped on an object surrounding the scene. The use of skyboxes becomes relevant when you need to portray a very distant environment, this can be anything and can even include effects or the likes, and the only limitation is that it is something which is supposed to be far away. Very distant objects do not suffer the *parallax* effect. (6) The parallax effect only occurs on objects which change depth and orientation when we move past or around them. Distant objects are only affected by this in a very limited sense, which enables the use of skyboxes.

The relevance of skyboxes towards this project is connected through skyboxes ability to reduce the amount of elements being portrayed in an environment, thus reducing the amount of processing tasks on the hardware; this could mean a performance improvement.

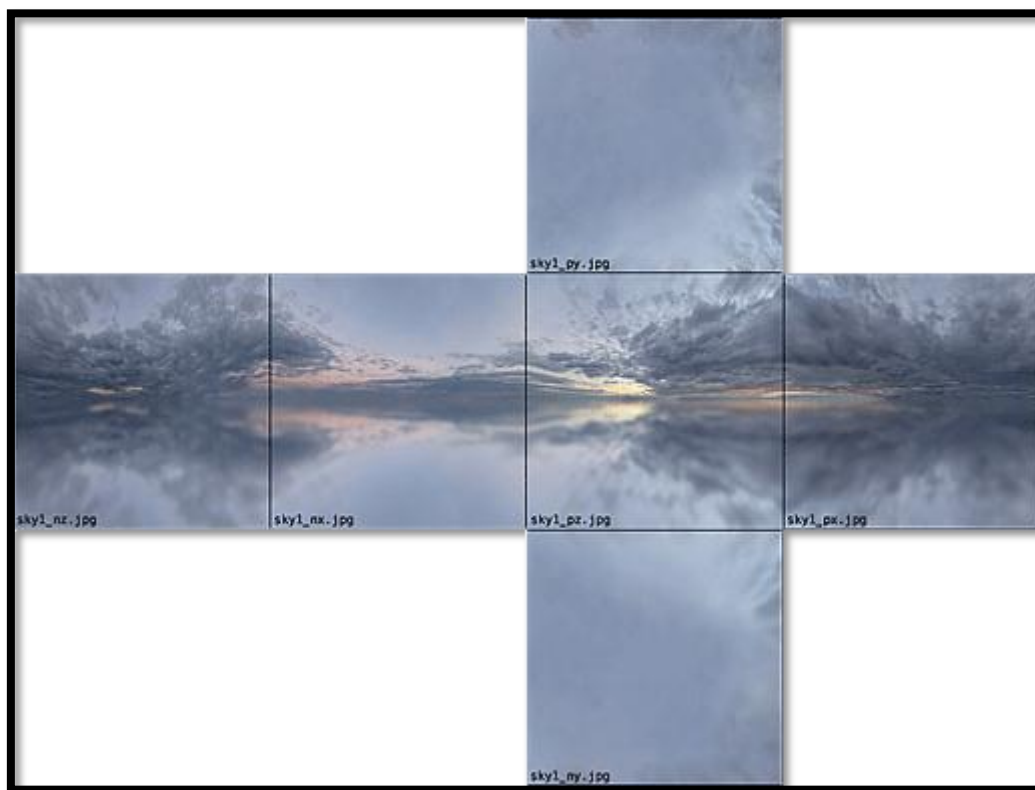


Figure 2 – Image showing the example of an environment map for a skybox. (7)

Figure 2 shows what an environment map for a skybox could look like. It is consisting of six squared images taken in a panoramic fashion, making each of them work as one side of a squared box in which a sky scenario will take place. The images are connected such in a way that makes it hard to notice the edges of the box when it is wrapped with textures.

A skybox is not specifically a way to portray effects and is not described as a way to portray effects, however it is important to notice that the possibility of doing that is there, and it is a viable way of portraying effects with less render time as opposed to real 3D renders. Effects can be put into the

skybox's texture; this becomes especially valid when the amount of time where the skybox is viewed at a time is limited, so that the change in the visual effect is close to zero, or if the skybox's texture is somehow animated to portray the effect.

5.2.2.2 Sprites

A sprite is basically a 2D image which can be placed anywhere on the screen, imagine a mouse cursor or a dialogue box within a game. When talking about sprites we come closer to a render type which is much more specific towards the possibility of portraying effects. Sprites are also not limited to being squared, parts of the image can in most render engines be portrayed as transparent.

George Geczy refers to sprites as *"The king of 2D Graphics"*, and he continues:

"At its very simplest, a 2D engine is capable of drawing a background image and then drawing individual graphic objects -- sprites -- onto that image. The use of colour-keying allows a specific colour in a sprite to be considered transparent, which allows sprites to be any irregular shape. And the uses of a series of sprites to represent an object, for instance a number of different facings of a unit, allow the simulation of animation and action when they are drawn in sequence." (8)

Putting all this together, a sprite is best explained as an image or image sequence, which is placed upon or inside the environment. It never changes orientation, but can only differ in size and position, a technique only utilizing two dimensions. However, it can be made to mimic 3D to a certain degree.



Figure 3 - Screenshot taken from the game *Command & Conquer: Red Alert 2* by Westwood Studios, Copyright © 2000, showing a great example of how sprites can be used to mimic 3D objects. (9)

Figure 3 portrays a screenshot taken from the game *Command & Conquer: Red Alert 2* published by Westwood Studios in 2000. (10) This is a good example of what sprites are capable of in terms of mimicking 3D objects. Sprites can also contain a z-depth value, which means that one sprite can be placed on top of another, based on the z-depth value assigned to each of them.

The sprite technique might be the king of 2D graphics, but when the game changes to actual 3D, sprites lose some of their advantages, and become less desirable in most situations. One of the hurdles of using sprites in 3D are their fixed orientation. As sprites cannot be placed on a 3D dimensional position, they cannot be used to display objects or effect within that environment without compromising the illusion of being a 3D object. This is where the billboarding technique comes into play.

5.2.2.3 Billboards

Billboards or billboarding is not very different from using sprites and consist more or less of the same technique. However billboards take use of polygonal content to portray their images. The basic principle of billboarding is to take an actual 3D polygonal plane and plant a texture on it, much like a sprite, but what makes the difference is that billboards can be stuck in a 3D position; this allows billboards to be used in a 3D settings. It also enables the possibilities of allowing objects to change perspective and orientation in 3D while maintaining the advantages which makes it more or less as fast to render as a sprite.

George Geczy explains the render technique from a 3D perspective as:

"In 3D terminology, what needs to be done is to "render a textured quad with an orthogonal view." This is actually quite simple in plain English -- a quad is a square or rectangle, and a texture is just a bitmap image. Drawing textures on polygons (triangles, squares, whatever) is a basic function of 3D hardware. The orthogonal view means that you'll be looking straight down on it, no perspective effects." (8)

Another feature of the billboard is that it can be adjusted to face the camera, giving it several options for displaying various elements. Akenine-Möller et al. lists a number of objects and effects which are often created with billboards, particularly they mention the use of smoke and fire, as I did earlier in this report. They write:

"Billboarding, combined with alpha texturing and animation, can be used to represent many phenomena that do not have smooth solid surfaces. Vegetation, especially grass is an excellent candidate for this type of technique. Smoke, fire, fog, explosions, energy shields, vapour trails, and clouds are just a few of the objects that can be represented by these techniques." (2)

In Figure 4 a number of billboards portraying a fire can be seen, this is taken from the game *Counter-Strike* from Valve in 1999. (11) The fire looks great and realistic; however it is easy to notice that these are billboards, especially if you notice the clipping between the fire and the wooden box in the right side of the screen.



Figure 4 – Screenshot from *Counter-Strike* by Valve, Copyright © 1999.

António Ramires Fernandes provides some comments to the use of billboards, while at the same time referring to a similar problem as the one presented in this project:

"This technique is quite popular in games and applications that require a large number of polygons. Regardless of how fast graphic cards get (Thanks NVidia!), it seems that it is never enough. Games take advantage of the hardware advances to push things further to the next level in complexity and detail (In turn hardware manufacturers try to keep up with games designers in a never ending cycle). 3D graphics programmers in general, and game programmers in particular, have a constant struggle with the amount of polygons that can be displayed with a decent frame rate." (12)

He also confirms the various possibilities and advantages of billboarding:

"Billboarding can be used to give the illusion of a 3D object without compromising the polygons budget. However billboarding is not limited to saving polygons. In general, the billboarding technique can be applied to make any object face a target, where the target may be the camera, another object, or simply a 3D location." (12)

Having gathered this general information about billboarding, it is clear that billboarding can be a great tool for reducing hardware stress. But there are a number of different types of billboards which each have a different usage, and a different number of advantages and disadvantages.

5.2.2.3.1 Screen-Aligned Billboard

The most common billboard is a *screen-aligned billboard*. Here the image is always parallel to the screen. This means that the direction from the billboard to the camera is constant, as well as the up direction is constant, as they are both defined by the placement of the camera in the scene. The last direction is determined by finding the cross product between the two known directions or vectors as they are called.

This is a good way to show elements like text, heads up displays (HUD) or other elements which needs to be constantly facing the screen, while also maintaining the same up direction. (2) The reason for this is that these types of elements are not presenting objects in respect for the worlds up (*y*) direction but with respect for the cameras up direction. (13) This type of billboard is effective for portraying effects that are not required to be constrained in any way towards the virtual environment, but only towards the orientation of the camera.

5.2.2.3.2 World-Oriented billboard

World-oriented billboards are different from screen-aligned billboards only because of the up direction in the billboards general orientation. Here the objects are rendered with respect to the worlds up direction and not the cameras up direction as screen-aligned billboards are. This is basically the only difference; these images still aligns themselves towards the camera but the up direction of the billboard remains constant towards the world and not towards the camera, often used for objects like clouds or lens flares. A small downside of using world-orientated billboards is that when viewing sprites that are of a little larger size the object can be skewed when aligned with the view plane. A solution to this is to make the direction vector point towards the viewer's position, so that all sprites are aligned towards the viewpoint.

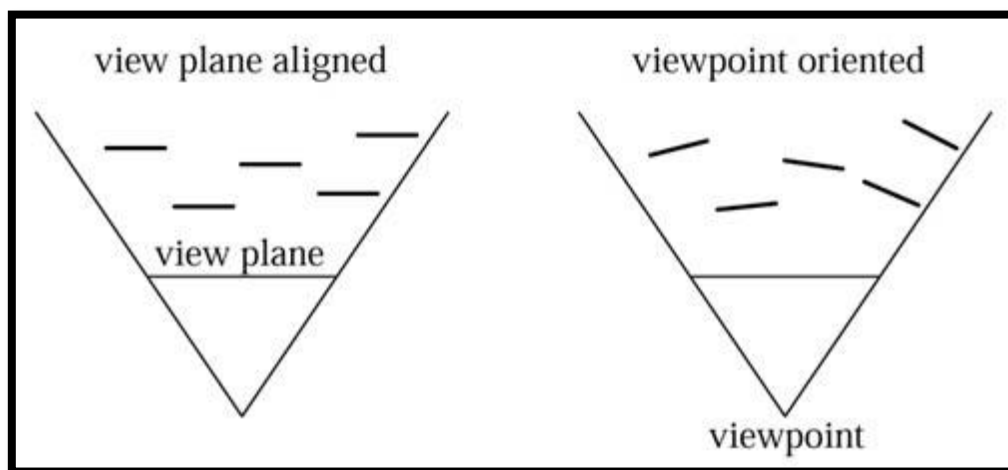


Figure 5 – Image showing the difference between view plane and viewpoint billboards. (14)

Figure 5 shows an image from the book *Real-Time Rendering* which perfectly portrays this issue. When the sprites are small, this issue can partly be ignored as the skewing then becomes harder to notice, however when the sprites are above a certain size the viewpoint oriented solution has to be implemented. (2) The difference between these two types of billboards are not noticeable when the camera moves around the objects, moves up or down or moves closer or further away, but only noticeable when the camera rolls around the view axis.

5.2.2.3.3 Axial Billboard

Axial billboards are different from world-oriented billboards and screen-aligned billboards (also sometimes referred to as point-billboards), as they are only able to rotate around some fixed angles, while the last angle is locked in relation to the world. This is a good technique to portray objects such as grass, trees, general vegetation, and other cylindrical objects or similar which can be found in a scene.



Figure 6 – Screenshot from *World of Warcraft* by *Blizzard Entertainment*, Copyright © 2004, portraying various grass and tree billboards.

Figure 6 shows a screenshot from the game *World of Warcraft* released by *Blizzard Entertainment*. (15) In this game both the grass and the crowns of the trees are consisting of axial billboards. However the grass rotates around the Y-axis to constantly face the viewer, the tree crowns do not, and they are fixed in both Z and Y direction, but are consisting of two or more billboards which are crossing each other.

5.2.2.4 Imposters

Billboards are often referred to as imposters, however imposters are slightly different. An imposter consists of real 3D objects which is rendered on an off-screen image, and then saved as a texture which is placed on a billboard within the game; this is done to improve performance.

Imposters also have downsides though, one of them is when the orientation or distance changes, either the illusion breaks because of the different viewing angle, or the illusion breaks because of the pixels growing too large when the imposters comes too close to the viewer. Imposters can circumvent this by setting a threshold, like a maximum angle or maximum distance in which the viewer can move, and once this threshold is passed, the imposters are re-rendered.

Imposters can also be pre-rendered and cached, this may be classified as a billboard but that is not of importance. What is important is to recognize this method as being very clever. A pool of different angles of the same animation is then rendered, and the billboard or imposter is then portrayed differently, depending on which angle or distance it is viewed from. The billboard or imposter simply contains all methods. This might be larger in data size, but a lot faster in frame rates.

5.3 Related Studies

5.3.1 A Three Dimensional Image Cache for Virtual Reality

This report might not be exactly categorized as a related study, but it was one of the first studies to introduce the concept of imposters, as because of that it has been found interesting and somewhat relevant as a knowledge database for this report. It was written by *Gernot Schaufler* and *Wolfgang Stürzlinger* in 1995, (16) and presents the use of imposters to maximize frame rates in virtual environments. They present impostors and 3D virtual caches as a possible solution to this problem, which is directly linking to the problem in this report, and thus an interesting read.

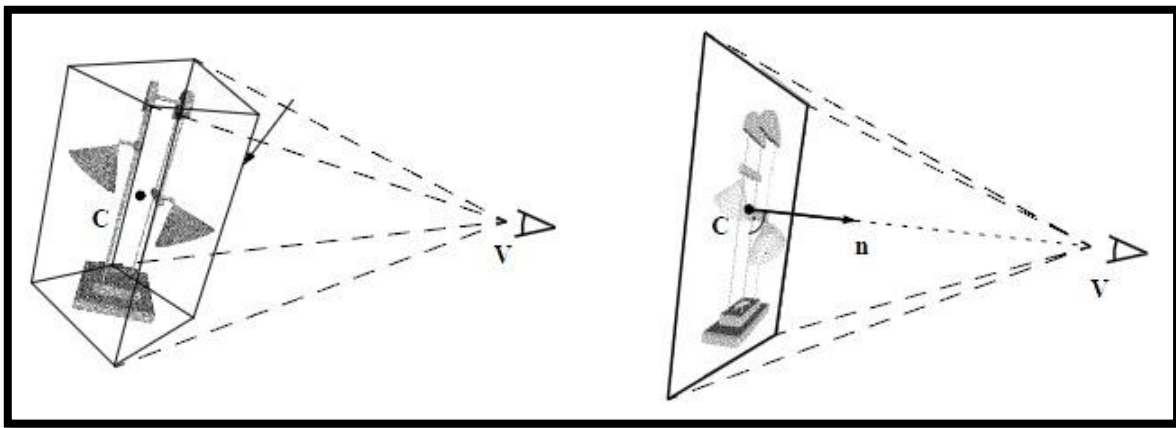


Figure 7 – Image showing the differences between a real 3D object and its corresponding imposter. (16)

Figure 7 shows an image from their report, here a visual interpretation of the concept is presented and afterwards carefully explained both conceptually, but also mathematically and technically, as how to create, use and implement this method.

This report is also interesting as several other reports read during this project refers to this report, and gives generally gives good references towards this report, making it a desirable read for information.

5.3.2 The use of Imposters in Interactive 3D Graphics Systems

In a report by *Kenneth Rohde Christiansen*, (17) he wishes to examine the use of imposters to maintain frame rates. He bases his analysis on the concept created by *Gernot Schaufler & Wolfgang Stürzlinger*. (16) In their studies they introduce what they refer to as *dynamically generated impostors*, which is explained further in section 5.3.4 and 5.3.5. In *Kenneth Rohde Christiansen's* report, he examines whether or not it is viable to use imposters for rendering distant occurrences of a 3D object, while close occurrences are rendered in real 3D, and whether or not this difference is noticeable.

The basis for his study reminds us of the basis and problematic described for this project. He describes the process of his report as:

"In this paper we have looked at a method for speeding up the rendering of complex scenes in order to improve the frame-rate and thereby the user experience. The idea was to use so-called imposters and reuse these over a series of frames. The imposters were re-generated when they diverged too much from

the original object due to the user/camera having turned more than a given degree.” (17)

And then concludes his report by saying:

“In order to look at the usability of the method we did a proof-of-concept version and conclude that the negative visual impact is neglectable. Dynamic generated imposters are an easy-to-implement method with almost no visual impact that with a good implementation has the ability to speed up the rendering of complex interactive 3D environments.” (17)

There are some differences between his study and this study, one of them is the fact that he is talking about 3D objects. However the problem and the method of achieving the goal are very similar. While he is reducing polygon count by showing billboards instead of 3D objects, one can imagine that in a smoke effect, particles or voxels can be reduced in the same manner, thus reducing calculation and render time in the same way as with 3D objects. He concludes his report as seen in the above quote, by saying that imposters are an easy-to-implement method which has the ability to speed up rendering of 3D environments. This conclusion is not limited to 3D objects.

5.3.3 Animated Impostors for Real-time Display of Numerous Virtual Humans

This report was published not long after *Schaufler* and *Stürzlinger* published theirs. It is published by *Amaury Aubel et al.* (18) This study is focusing on the use of imposters as a replacement option for virtual environments that contains a large number of the same objects, in this case human’s models. They focus directly on the use of imposters to improve frame rates.

Taking my thesis statement into consideration, it is clear that this project attempts to solve some of the same problems as they do here. They strive to find a solution which is a sufficient substitute to a real object, in this case a human model, which can aid in reducing the amount of hardware stress, in order for the frame rates in the virtual environment to be more consistent.

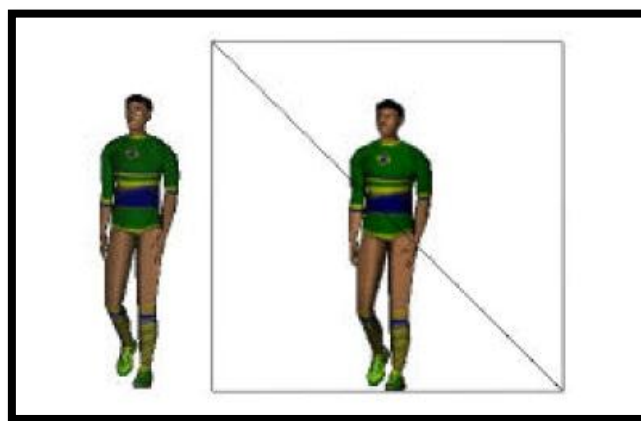


Figure 8 - Image showing a 3D football player and its corresponding imposter. (18)

The report has a successful conclusion, in which they write:

This paper shows that image-based rendering can be applied successfully to virtual humans. In particular, we have explained how to generate a texture

representing a virtual human and proposed an image cache invalidation algorithm that works for any articulated character and executes reasonably fast. (18)

It validates the initial thoughts and pre-analysis in this project, as it shows IBR as a viable technique for reducing hardware stress and speeding up rendering of objects which would otherwise limit real-time display.

5.3.4 Real-Time Cloud Rendering

This is a report written by *Mark J. Harris et al.*, (19) in which they present a method where a particle system is used to generate a cloud like effect within a game. This is an interesting study as it introduces a particle system to generate a virtual effect, but also stumbles upon the problem which can arise when particle systems are becoming too heavy on the hardware, which forces them to find an alternate solution.

The concept of this report is to use particle systems to create a cluster of smaller billboards which as the end product looks like a somewhat realistic cloud. The particles are pre-generated, which means that within the game, only the shading of the clouds is rendered. A problem arises when their scene becomes too complex, and the frame rate starts to suffer. The decision to implement imposters is then taken, and they put up two solutions: either to use pre-computed imposters which contain images of the object generated from multiple viewports, or to use dynamically generated imposters which are then generated and rendered when only when needed. The latter is chosen.

The result was very positive, and even in a scene where the imposters had to be redrawn every frame, the frame rate and performance was still better than without the imposters. It can be seen from their conclusive graphs that the performance increase was very real and substantial. An example taken at around 640000 particles shows a frame rate of ~8 without imposters, and a frame rate of ~20-30 with the use of imposters. That is a substantial increase of frame rates, especially in this area revolving around such low amount of frames, as the human eye is said to perceive fluent motion at around ~24 frames per second, give or take in special circumstances. (20)

This is an interesting report, both because of their great results, but also because they discuss and reflect on several different uses of imposters in their argumentation, before implementing a solution.

5.3.5 Real-Time Tree Rendering

This report is written by *I. Remolar et al.* (21) It focuses on real-time rendering of trees and vegetation, and is based on a problem that arises when complex models contains too many polygons and thus slows down the system, limiting the frame rate.

This is again a report where fully or partial usage of IBR is used to replace objects in a virtual environment in order to reduce hardware stress and ensure consistent frame rates.

They focus their research on methods which allows them interactively either to lower the resolution of the polygon models, depending on the distance, or to completely replace the model with a dynamically generated imposter. They implement a clever use of both, but particularly their use of imposters is interesting. They create a movable imposter, which gradually moves through the polygon model, thus replacing more and more of the actually model with the imposter, the further away from the object

you get. Naturally there is a difference between a smoke effect and a tree model, mostly because the smoke effect in most cases would be animated. This changes how it could be used. Also when animated, the centre of the effect or model would be likely to shift position, thus this technique would have to be altered. These difficulties are not experienced in this report, but have to be kept in mind when relating this report to my project.

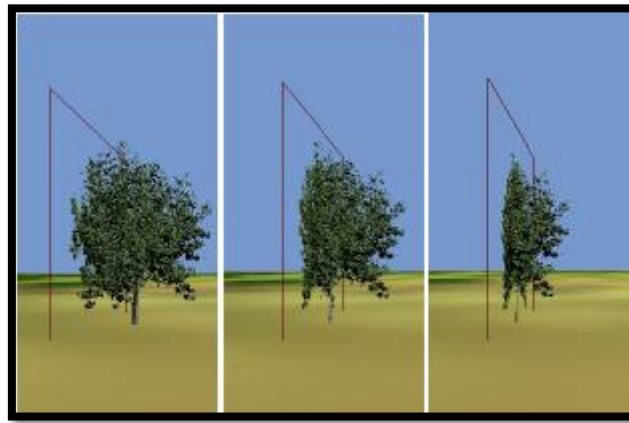


Figure 9 - Image from their report showing how their use of imposters gradually replaces more of the actual model. (21)

Again, a study is shown where the use of imposters can improve the frame rate, and in this case, with great focus on not reducing the quality of the scene. An example from their conclusion is taken at a scene with eight trees, with fully polygon models, the scene renders at ~26 frames per second, but with the use of imposters they manage to get a frame rate of ~54 frames per second.

5.3.6 Spherical Billboards and their Application to Rendering Explosions

This report is written by *Tamás Umenhoffer et al.* (22) and it does not share any of the same specific problems as this report; however it is an interesting read as it clearly shows billboards obvious ability to portray effects such as fire and smoke, and does it well. It also presents a more modern way of using billboards. They have developed a new method which uses multiple billboards on particles to render a volume billboard that provides a better graphical finish than using regular billboards. This is particularly useful as it shows that using billboards or imposters is not an out dated technique and can be altered and used in various ways to portray great visuals.

5.3.7 Additional Inspiration

A few selected other reports has also been used for inspiration in this project, however as they are of less relevance to the actual problem they will not be explained in detail as the previously explained reports. This concerns *Realistic Real-Time Rain Rendering*, (23) and *True Imposters*. (24)

5.4 Changeable Elements

Reducing a real-time effect to one being used in an image-based render method is a great technique for relieving stress on the computer. However, there are still areas in which the effects can be altered in order to minimize the hardware stress. Calculation processes, transfer processes and loading processes have to be taken into account. This means that processing unit, hard-drive and memory will

all be considered when finding possible areas where various IBR techniques can be combined to relieve hardware stress. However implementing a multitude of solutions might have a tendency to reduce the quality too much, thus making the effect undesired because of poor quality. A delimitation of solutions has to be made, in order to choose those most relevant.

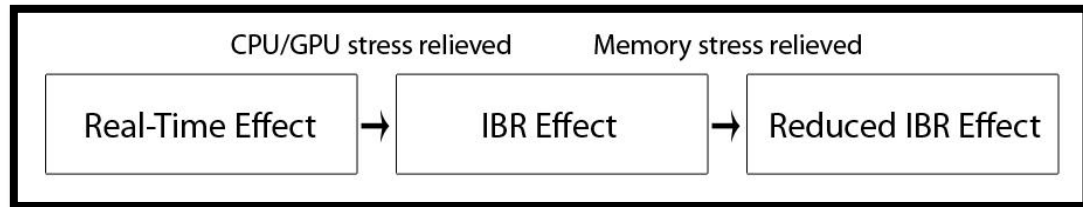


Figure 10 - Image showing the effects on hardware by changing method for portraying effects. Image by the author.

Figure 10 shows an image which shows the possible areas of hardware which can be relieved of additional stress when changing how effect are portrayed. The challenge had two aspects, first finding the optimal IBR method for reducing the effects, secondly the challenge lies in finding aspects of that IBR method which furthermore can reduce the stress, and this last part is equally difficult. This section will attempt to highlight some of the areas in which experiments can be done to attempt to also relieve memory stress when using IBR techniques to show effects within a game or virtual environment.

5.4.1 Shadows

One possible method for reducing hardware stress is to change the method of portraying shadows for a specific effect. Shadows can be portrayed in different ways, one method is to calculate the shape of the shadow directly inside the game, and this uses processing power. Another technique is to use a pre-rendered shadow map, much like an image sequence in the same way the effect is portrayed, and this takes up memory instead. However reducing a shadow map to only contain one image, or a small looping image sequence could be an attempt to reduce the shadows effect on hardware.



Figure 11 - Screenshot from *Counter-Strike* by Valve, Copyright © 1999, showing how shadows are presented. (25)

In Figure 11 a screenshot from *Counter-Strike* by Valve (11) shows how shadows can be reduced to simple circles below the character, requiring minimum hardware to run.

5.4.2 Movement/Viewport

Generally this means how the objects are viewed when taking direction, speed of the camera and rotation into account. Objects viewed from a static camera can more easily be replaced with imposters or billboards as they are only required to portray the object from one angle and in one type of render. However if the camera moves, or the object moves, the objects animation should change accordingly to accommodate this change. Dynamic imposters has been seen to be able to do this, they load multiple sequences of the effect, but only portrays the one which fits the angle at which the effect is being viewed. Does the camera move to a different angle, the effect changes sequence and shows the effect which fits that angle. A method for reducing hardware stress here could be to add a threshold for when the angle changes, making one sequence valid for several angles. This could reduce memory load, and perhaps be hard to detect by the viewer.

5.4.3 Compression/Level of Detail

The last element which I found interesting towards this project, in terms of optimizing and reducing image-based rendering is compression and level of detail. Right now many games use LOD (Level of Detail) to minimize render times. This means that they show low quality renders in the distance, and high quality renders when portrayed close to the camera. This is very effective and can be used both for billboards as well as real 3D objects, and can even be combined. See section 5.3.5 for an example. Imagine billboards being shown at distance where the angle and size of the object changes very slowly and then real 3D animations will be shown when the camera is close to the object. It can also be used solely on billboards, showing a billboard with a high compression method at the distance where pixilation and the likes are harder to notice, and then loading a billboard with less compression when moving closer. This would be able to reduce the memory as images and sequences would take up less memory when compressed more.

5.4.4 Selection

Using all these variables in a test is not desirable; it creates a very large amount of possible variations, and makes it extremely difficult to pinpoint which change is causing what result. So delimitation is a necessity in order to be able to receive proper test results.

5.4.5 Elements Not Included

There are some elements which do not work at all or very well with IBR techniques. One of these is the new PhysX technology, (26) which is a technique that control physics inside a game, and allows objects to be destroyed and manipulated. An example of the uses of the PhysX technology can be found in the game *Crysis 2: Be the Weapon* from *EA Games*. (27) This is another reason as to why it appears more relevant to choose an effect such as smoke or fire, as those are not yet being contributed much by new technologies such as the PhysX technology. Thus smoke is becoming an even more attractive effect to optimize and work on.

5.5 Delimitation

A lot of different angles, researches and alteration methods have been described in this pre-analysis, and to bring every aspect of those into the actual problem and solution would be an immense project which would not fit into the scale of this master thesis, additionally it would be hard to derive anything from so many different variables in a test scenario, so a few sacrifices and delimitations towards the research and production area has to be made.

Initially, it would be ideal to take a look at the different methods which has been presented on how to create effects, and how to portray them inside the virtual environment. Fluid systems are hard to compute but can also create stunning effects, and can still be used to pre-generate billboards or imposters. This might be the most obvious choice to go with according to the research found, and it is also a commonly used method within virtual environments as it is at the moment. However, comparing a fluid system with a billboard portraying exactly the same does not make much of a difference. Therefore the section about changeable elements was presented. There are a lot of variables that can change, both between fluid systems and billboards, but more importantly, what can change between two billboards themselves. The presented elements can make a huge difference in both appearance and performance, they are thus equally important to the problem as the discussion about creation method is.

Comparing the changeable elements and the creation methods there are already a few obvious things in common which are good for specifying a narrower problem, especially the section about imposters and movement/viewport. Imposters come in different forms and the differences between them are mostly focused on the way they behave compared towards the viewport direction and camera position. Looking back on the presented methods in this section, some elements appear less desirable than others. The most appealing one based on personal interests, appears to be the solution about reducing viewing angles.

Concluding on all this, without naming every detail of the selection, it would be ideal towards this project, to focus on **fluid systems** as the tool for the creation of the effect, **billboards** for the portrayal of the effects, and **movement/viewport** as the changeable elements which in the end creates the difference in the effect which will be tested upon, taking my knowledge towards the different aspects, and my skills in various creational elements into account.

5.6 Final Problem

In conjunction with the delimitation, a final problem has been created. This reflects the decisions that were made in order to narrow down the area of research and information which the project will span over. This problem describes specifically what the analysis will delve into, and exactly what needs to be researched before the design section can create a possible solution. The final problem is as follows:

To what degree can the amount of billboard image sequences representing different viewing angles of a smoke simulation be lowered, without the user perceiving a drop in quality?

This is to be understood as following: imagine a smoke effect portrayed on a billboard, here the threshold is 1 degree meaning that the billboard viewing angle is static as long as the camera does not move more than 1 degree to the left or right. However, when the camera does move more than 1 degree, a new billboard is loaded where the viewing angle of the effect on the billboard has changed. Here 1 is the threshold. The problem which arises here is that these types of billboards with such a low threshold are very hardware and memory intensive, so what can be done to reduce this? Making the threshold larger would mean that the billboard requires less memory to run as there are less sequences to be loaded, however if the threshold is too big, the illusion and the quality of the effect can be compromised. So what is the golden number of the threshold, and how far can you go in the creation process in order to solve this problem without compromising the quality?

5.7 Hypothesis

Following up on the hypothesis formulated in the pre-analysis, however more specific this second time around, as with the actual problem. Taking the hypothesis a little further from the initial one, it is the belief that even with minimal elements in a scene in addition to the effect under examination, there has to be significant changes in the quality of the effect for the user to notice. Small changes to quality can relieve a lot of hardware stress, and if the user does not notice a drop in quality, this change would be most welcome.

5.8 Target Group

There are two possible target groups that could benefit for this type of knowledge.

1. **Extremely high quality virtual environments for home use:** This means games and other virtual environments, that utilize very high quality graphics and animations which require a lot of memory, memory not yet found in all home hardware solutions.
2. **Effects used for mobile platforms, such as tablet computers and cell phones:** On these platforms where the hardware is limited, the ability to cut corners like the idea presented in this report are most welcome, especially if they can be used in a way that does not let the user believe he or she is viewing a lower quality version of an effect.

Creating this product and researching various ideas, there does not have to be a distinction of whether the effect is used in one or the other form of target group, as they would both follow the same developing procedure, and both benefit equally from the result (should the result prove to be a success).

6 Analysis

This analysis takes a step further from the pre-analysis and initiates with an examination of previous researches, however here the researches are much more specific towards the description given in the final problem. What are being researched in this section are articles about fluid smoke creation and billboards which change angles, articles that does not provide any insight into these two areas are not relevant.

6.1 Research on Fluid Simulation

This section will present specific papers which contribute to the understanding of fluids and their implementation using billboards.

6.1.1 Stable Fluids

This report is written by *Jos Stam* back in 1999. (28) He was the very first person to create fluids which were unconditionally numerical stable, which could provide a stable fluid output for use in the computer graphics, even though the time steps in between of calculations were raised. Previously, too large time steps would resolve in the fluid moving out of control or the system crashing.

In very short; through the *Navier-Stokes* equation he had created a stable advection scheme which allowed larger time steps to be achieved without loss of control of either the system or the fluid, or what he refers to as *blow-ups*. There are a lot more minor alterations to the solver in his report than what is mentioned above, and the technical details of their solution are not important to this thesis. However it is important to notice how this report has been accepted and used for implementation in software afterwards. As an example, high end animation software such as *Autodesk Maya* (29) currently solves the *Navier-Stokes equation* as well, and appears to be taking advantage of *Jos Stam's* research, as *Maya* fluids are not prone to *blow-ups*.

6.1.2 Visual Simulation of Smoke

Two years after the *Stable Fluids* report, *Jos Stam* released another report together with *Ronald Fedkiw* and *Henrik Wann Jensen*; which was published in 2001. (30) In this report they attempt to create a fluid solver that allows them to render realistic smoke, and their focus is to do this with a low grid value so that it renders more quickly. Additionally they want to the smoke to interact with objects in the scene.

There are two interesting elements worth noticing in this report. First, we here see that fluid systems in fact can be used for rendering smoke, and does it quite well even as of 2001. However, the second elements which we need to notice is that they do not use the *Navier-stokes* equation as he described in his first article titled *Stable Fluids*, he uses what is called *inviscid Euler equations*, which is different way of solving the *Navier-Stokes* fluid equation. The main difference is the fact that the *Euler equation* is inviscid, meaning that there are no resistance in the fluid; this is also referred to as the *thickness* or *internal friction*. Inviscid fluids are also referred to as *ideal fluids*. An article from Princeton University suggests an example where the difference in viscosity is compared to the difference of water and honey, water being low on viscosity and honey being high. (31)

6.1.3 Simulation and Animation of Fire and Other Natural Phenomena in the Visual Effects Industry

The simulation of fire and smoke are several places seen to be of somewhat similar nature, and has been seen to use more or less the same algorithms and attributes as each other. In this report by *Duc Nguyen et al.*, (32) they introduce a list of techniques for creating very realistic fire, smoke and explosion effects for high-end productions. They use a technique referred to as the *ghost fluid method*. They are solving the *Navier-stokes* equation, and they end up with some astounding looking results, which have also been used in feature movies such as *Terminator 3: Rise of the Machines*. (33)

This is a great report which shows the possibility of creating great looking effects through a fluid solver, especially the possibilities of solving the *Navier-stokes* equation. It is also important to notice both the *Lagrangian solver* and the *Euler solver* is mentioned which are two alternatives which both need to be explored. This report is valid for my study both in terms of confirming the possibilities of creating fire and smoke through the use of a fluid system, but also by providing alternative solution methods which can help provide insight to the best possible solution for creating an effect.

6.1.4 Interactive Fluid-Particle Simulation using Translating Eulerian Grids

This report by *M. Cohen et al.* is a report released by NVidia and presents one of the newest methods for creating real-time smoke. (34) This is done with a fluid system that is visualized through its influence on particles within the system. It shows how fairly complex smoke mimicking movement now can be rendered real-time; using specific solvers and GPU accelerated calculations.

The solution presented here uses what they refer to as *translating Eulerian grids*; this basically means that their solution builds on the *Euler equation*. They themselves describe their method as:

"We demonstrate a method that combines the high fidelity and high performance of modern Eulerian methods with the flexibility and controllability of particle systems. By moving all data-intensive computation onto the data-parallel processor, the GPU, we achieve results that will scale nearly linearly with the number of GPU processing cores." (34)

They simply use state of the art *Euler equation* solvers, and integrate them in a system that allows the calculations to be done on the GPU apart from the CPU which currently has been the main calculation unit for processing fluid and particle systems. They also a different grid system which does not have the space limitations as other solutions as it is allowed to *move* along with the car, but they limit elements like turbulence only to the vicinity of the object releasing or influencing the fluid, this attribute becomes more static the further away it gets, thus computationally lighter.

6.1.5 Limitations Still Exist

Even with the great real-time results provided by NVidias modern hardware and fluid solvers presented in their article *Fluid-Particle Simulation using Translating Eulerian Grids*, there are still a number of limitations which makes these methods unsuitable for implementation in newer real-time virtual scenarios and games. *M. Cohen et al.* state themselves:

"While our system could be dropped into current games at low resolution, because we are riding an exponential performance curve, these methods will be practical on midrange consumer hardware at high quality in a few years." (34)

Their solver scales according to them, linearly with the amount of processing cores on the GPU, however quality in virtual environments can end up scaling unexpectedly high. So acknowledged by themselves, these methods can right now function in low resolution environments, but are not yet ready for the common consumer hardware for a couple of years. This is an important observation as it makes the problem of this report viable for years to come. Very high quality renders of smoke effects are still not quite ready to be processed real-time, this the solution of pre-rendered billboards are still one of the leading options when implementing smoke effects into a real-time virtual environment.

6.2 Simulation Techniques

Scouting the market for software products that contain a fluid system which can be used for the creation of smoke and similar effects, a large number of potential solutions quickly presents themselves. We have *Autodesk Maya*, we have *3D studio Max* and its plugin *FumeFX* and *Houdini Pyro*, however none of them provide absolutely clear information of what solver they use. Then we have *RealFlow* which uses a particle and rigid body system which generates a large amount of particles and then creates a mesh around them, however it is not a fluid system as those found in *Autodesk Maya*. *Softimage ICE* is also a system consisting of rigid bodies, soft bodies and particles into some form of liquid animation. These are just some of the solutions out there, and some of them even contain several solvers, *Houdini Pyro* can also use an *Eulerian/Lagrangian* solver, while *Autodesk Maya* also has a simpler *Spring Mesh* solver.

So how does one find the right solution in this confusing web of possible solutions, and what settings are the best to use? In a report from 1997 by *Nick Foster et al.* they use the *Navier-Stokes equations* to approximate the movement of gas in space, (35) what appears to be the most common non-commercial approach to use some of the systems which are referred to as regular fluid systems (*Maya*, *Houdini* & *FumeFX*). (36) Yet what solver-algorithm is desired? Here computational speed, appearance, realism and complexity are a factor.

6.3 Fluid Algorithms

This section will go through the various common algorithms provided in software for calculating and simulating motion and behaviour of fluid systems. The list of methods has been generated based on the algorithms used in reports presented in the previous research sections, and have thus all been used for fluid simulation before with varying results. The standard equation algorithm is the *incompressible Navier-Stokes*. The other equations are most of them a derivative of the *Navier-Stokes*, or simply a different way of solving the *Navier-stokes*.

The equations for dealing with fluids are all deriving from *Newton's second law of motion* which is also referred to as the conservation laws, and specifically the conservation of mass and conservation of momentum, while also including the *First law of Thermodynamics* which deals with the conservation of energy. (37)

6.3.1 Navier-Stokes

The *Navier-Stokes equations* were discovered in 1822 by *Claude-Louis Navier*, (38) and then later modified by *Sir George Gabriel Stokes* in the years from 1842 to 1850. (39)

The *Navier-Stokes equations* has in the early years of its use in computer graphics and simulations been prone to cause blow-ups, the complexity of the equations at times simply caused the equations to be numerically unstable and create an uncontrollable fluid chaos. This was solved by *Jos Stam* which is also described in the article called *Stable Fluids* (28) in the previous research section 6.1.1, which now makes the *Navier-Stokes* solvers quite reliable and precise. The *Navier-Stokes* is presented by *Robert Bidson* as the famous *incompressible Navier-Stokes equation*. (40) He presents it like this:

$$\begin{aligned}\frac{\partial \vec{u}}{\partial t} + \vec{u} \cdot \nabla \vec{u} + \frac{1}{\rho} \nabla p &= \vec{g} + \nu \nabla \cdot \nabla \vec{u}, \\ \nabla \cdot \vec{u} &= 0.\end{aligned}$$

Figure 12 - The *Navier-Stokes* formula. (40)

At first glance, the *Navier-stokes equation* looks rather complicated, however once the different elements are explained it becomes easier to know what the actual formula deals with.

- The letter \vec{u} is used for the velocity of the fluid.
- The Greek letter ρ is used for the density of the fluid.
- The normal letter p is used for pressure of the fluid, meaning the force which the fluid exerts on anything.
- The letter \vec{g} is the gravity acceleration.
- And finally, the Greek letter ν represents what is referred to as *kinematic viscosity*. This determines how viscous the fluid is. Thin like water or thick like honey.

The first equation in Figure 12 describes what is known as the *momentum equation*, which is the equation which tells us how the fluid moves as a result of the forces inflicted upon it. The last part is called the *incompressibility condition*, which is a formula which helps make sure that fluids does not change volume. As fluids rarely in the real world change volume, it is easier and more idea in animation to work with *incompressible fluids*. (40)

Additional information and resources used for the understanding of the *Navier-Stokes* can be found in these references: educational slideshow from *Washington University*, (41) report by *Charles L. Fefferman*, (42) and explanation from *NASA*. (43)

6.3.2 Solvers

The first way of solving fluid motion is through the *Euler equations*, which was discovered by the pioneering Swiss mathematician and physicist *Leonhard Euler*. (44) It was first published in *Mémoires de l'Academie des Sciences de Berlin* in 1757. (45) The *Euler equations* are a set of equations which describes the physical dynamics of inviscid fluids.

What makes the *Euler equation* stand out from the other equations listed here is that it deals with inviscid fluids. This means that the factor dealing with the resistance of the fluid, or more specifically the *thickness* or *internal friction* is not included in this equation. (31)

Another solver is the *Lagrangian equations*. It was invented as a result of *Joseph-Louis Lagrange's* study on classical mechanics from 1788. (46) (47)

The major difference between the *Lagrangian equations* and the *Euler equations* is the way that measurements of the fluid data are gathered. The *Lagrangian equations* view on fluid flow follows a single fluid parcel as it moves. This means that each point inside the fluid is treated as a separate particle with a position and a velocity. Whereas *Euler* view on flow is static and views the fluid flow through a fixed position in space. *Robert Bidson* has managed to simplify the explanation to something easily understandable:

"One way to think of the two viewpoints is in doing a weather report. In the Lagrangian viewpoint you're in a balloon floating along with the wind, measuring the pressure and temperature and humidity, etc., of the air that's flowing alongside you. In the Eulerian viewpoint you're stuck on the ground, measuring the pressure and temperature and humidity, etc., of the air that's flowing past. Both measurements can create a graph of how conditions are changing, but the graphs can be completely different as they are measuring the rate of change in fundamentally different ways." (40)

It can be said that the *Euler* way is a fixed grid, which does not change position in space as the fluid moves through it, while the *Lagrangian* way reminds more of a particle system.

6.3.3 Summary

The *Euler* and *Lagrangian* methods for solving the fluid dynamics are simply two ways of calculating and solving the same mathematical problem. The result *can* vary depending on circumstances, however when it comes down to choosing what to use, it appears that a lot of software solutions only provide one way of solving it, based on either *Euler*, *Lagrangian* or a mixture of them both. As a result of this, I am inclined to conclude that the choice should lie in the software, and not the solver within the software.

Autodesk Maya appears to be solving through *semi-Lagrangian* solution, however I have been unable to find any official statements providing any exact information about this. (48)

6.4 Software Options

As already briefly mentioned in section 6.2, some software solutions are more suited for specific purposes. Some programs are designed with specific solvers, equations; physics and mathematics built into them, and thus are more suited for this specific task. Other can have the same options through add-ons and updates, and some use totally other solutions. What matters in choice of software is settings and options, but also difficulty, availability, price and learning curve.

6.4.1 Autodesk – Maya

Autodesk Maya is a 3D production software program that can solve a large variety of jobs, and is easily integrated with other software's and solutions. Maya is mostly used for animation in movies and games. It has a lot of tools which allows the creation workflow to be optimized; it can create fur, cloth, water and particles among other things. Utilizing the *Maya Dynamics* section of Maya, it provides a great opportunity for creating realistic smoke and fire effects. Learning the simple tasks in Maya is fairly easy, but Maya contains a big set of advanced features that allows more experienced users to create really complex and specific animations and effects.

Maya is provided by Aalborg University, and also taught during classes which give Maya a big advantage. I was taught in *Maya* at Aalborg University from the 5th semester which gives *Maya* an easier learning curve compared to other solutions.

6.4.2 Alternatives

There are a lot of alternatives to *Autodesk Maya*, however as *Maya* provides all the necessary tools, are provided freely by the university, and has contributed with a lot of previous experience which is useful for this project, the alternatives are all less appealing. The alternatives which were investigated are; *RealFlow*, *Autodesk 3D Studio Max*, *Sitni Sati – FumeFX*, *Exotic Matter – Naiad* and *Alembic – Houdini*, however *Maya* were still considered to be the optimal choice.

6.5 Billboards and Fluids Combined

Another issue which needs to be addressed is to determine which method is best for portraying the generated effect within the desired environment. Creating and rendering a realistic looking smoke effect is don't most effectively via a specific method, but as we already concluded, those solutions are not suited for real-time and thus the effect has to be portrayed otherwise. There are three types of billboards which are suited for portraying smoke.

1. Billboard Clusters.
2. Particle Billboards.
3. Fully Rendered Billboards.

Each of these can be used for portraying a smoke effect effectively within a real-time environment. However they need to fit the render method and be able to portray the generated effect exactly as it looks within the software generating it. This is an important question as some of these methods are better than others in certain environments, and vice versa. There are examples of good looking smoke effects with all of these methods.

6.5.1 Billboard Clusters

Billboard clusters is a collective billboard which is joined together as a whole from a set of smaller billboards. This type of billboarding is effective for portraying effects which needs to somehow interact in 3D with itself, or to portray an effect which needs a form of random variation, as each part of the cluster can be selected randomly. Also used to portray more static things, like trees where the crown consists of two billboards with an angle between them to give the illusion of 3D. This can also be used to portray smoke effects, clouds and steam. However, when the effects become larger, more

complex and most of all, more specific in its appearance, this effect is too random, there is no benefit from sectioning the effect at that point.

However there is one big benefit from this type of billboard that other methods such as fully rendered billboards cannot provide, which is the actual presence of 3D within the effect. A great example is when virtual environments try to portray very puffy smoke, fog or clouds. Instead of portraying them through one billboard, it is consisting of several ones which are separated from each other through the Z direction, often seen in flying simulators. When flying through a cloud, a normal fully rendered billboard would instantly turn from cloud to no cloud, where as a cloud shown through billboard clusters can be more gradient in its appearance when passing through.

6.5.2 Particle Billboards

Particle billboards are in some ways exactly like billboard clusters, it consists of a number of smaller billboards which makes up a whole. However, particle billboards are directly chained to a particle system within the environment. This allows a certain amount of dynamic within the effect, as you actually can get some of the same variables in real-time as you do when pre-rendering effects. The clever thing is that you can limit the number of particles to a fairly small amount, and then attach a texture to each particle, making the appearance resemble that of a pre-rendered fluid or particle system a lot. To me, this sounds ideal, yet this method *can* end up facing some of the same performance problems described earlier when the effects gets really extensive and of very high quality.



Figure 13 - Screenshot from *Call of Duty: Black Ops* by Activision, Copyright © 2010, showing a fire and smoke effect which appears to be created through a particle billboard system. (49)

Also, at times I find it somewhat easy to notice that these types of effects are actually composited of several smaller image sequences, which for me breaks the illusion of a *complete* effect.

6.5.3 Fully Rendered Billboards

This method is consisting of a single billboard which is portraying the entire effect. The upside for this type of effect is that the only limitation to the quality of the effect is those created by the software generating the actual effect, and the resolution of the virtual environment and the texture. The downside is that this billboard only has one layer, and when moving around, or moving through, the illusion of 3D is easy to break. Also, very large textures can be quite intensive towards the memory of the graphical hardware. However, this is more desirable than to overload the actual graphics processor. This type of billboard will allow the quality to be directly translated from your creation software to your virtual environment, and it does not break the billboard into several smaller pieces as the other methods, here the effect stays as one unit. In a controlled environment where the movement of the camera is predetermined or very limited, this type of billboard is close to optimal.

7 Test Methodology

7.1 Definitions

Before the test process is commencing, it is important to clarify the terms that the final design is supposed to be tested up against. Often even the simplest of terms can have several interpretational meanings, and can thus in different contexts mean very different things, or give different results.

7.1.1 Quality

In this project, the term in which the created effects will be tested up against is quality. Quality can be very subjective, but also be an extremely stiff and objective term. So which is going to be used here, and how would it be described?

It is difficult to precisely describe quality; the reason for this is that the definition of quality changes according to whether it is the user or the creator who is viewing, and how much information they have been provided.

John Locke (1632 – 1704), an English philosopher and physician, created a theory regarding the properties of quality, and the different aspects this single term poses. He believes that quality can be divided into two categories, a primary and a secondary. This is also referred to other places as intrinsic quality and extrinsic quality. A review of his thoughts from Stanford University writes:

“The primary qualities of an object are properties which the object possesses independent of us — such as occupying space, being either in motion or at rest, having solidity and texture. The secondary qualities are powers in bodies to produce ideas in us like colour, taste, smell and so on that are caused by the interaction of our particular perceptual apparatus with the primary qualities of the object. Our ideas of primary qualities resemble the qualities in the object, while our ideas of secondary qualities do not resemble the powers that cause them.” (50)

This means that the primary qualities of an object are the qualities which are present when we are not. This includes static elements such as; shape, mass, material etc. While the secondary qualities are elements which requires us to interpret the object for them to be present, such as; sound and colour. Colour is an interpretation of reflected light in our brains, without us; there is simply light and no colour.

So how does this definition of quality fit into this project? It seems apparent to link the primary qualities to the unchangeable elements of the expected outcome in this project. Two things come to mind when thinking of the effect which is to be produced.

1. **Smoke:** Smoke is smoke, and if the effect is being created with enough resemblance to portray this, smoke is and will be smoke.
2. **Imposter:** The effect is being created through the use of an imposter; this element as well is unchangeable. And no matter who is viewing the effect, the means of how it is portrayed, in this case through an imposter, does not change. This also includes attributes attached to an

imposter, such as having texture. And can also include elements such as being in motion (for an animated imposter).

What is referred to as the secondary quality of the effect, is a much more wide and interpretational matter.

If it is the desire to test on the quality of an effect, by describing it through attributes of John Locke's secondary quality, things might get messed up. The alternative could be to use terms which can be categorized though the secondary quality. Adjectives which can be used to describe the visual appearance of the smoke effect can be used instead.

As John Locke states, an object can be either blue or red, depending on the observer, but an object can also be beautiful or ugly. Of course this is very subjective and can get out of hand if interpreted by people with very extreme opinions about either of these terms. What can be used to eliminate this issue is a set of boundaries in which beautiful and ugly are defined, thus helping people providing the right opinion to the quality of the effect. See section 10.2.1.1 of how this was implemented in the test.

7.1.1.1 Side Note

There is a dilemma to this definition brought to us by *John Locke*, regarding how it is being used in this report. *John Locke* uses it to describe qualities or characteristics of objects, characteristics which can either change or not. In this project the object (effect) will be changed on purpose, and thus it is the goal to test whether or not the observer notices this change. However, the reason that this exact definition of quality is being used is because it helps us understand what parameters can be changed and tampered with, and which should be left alone to maintain the natural understanding of the effect which will give a more clear insight in object values and subjective interpretations. Especially interpretations caused by physical appearances and subconscious reactions, and not those caused by beliefs, opinions, attitudes or standpoints.

What this basically boils down to, is *not* a definition that allows us to use quality as a way of describing what is good quality or what is bad quality. It is a definition, which separates the elements of an object into two categories which tells us which qualities or elements of an object that can be perceived different by the viewer, and which ones are static. Of course, this as a whole *is* what defines the actually quality of an object as being good or bad, but this comes from adjectives and alternative attributes which can fit into either of these categories. This is important to keep in mind.

7.1.1.2 Subjective Feelings or Objective Facts

Another way of referring to this intrinsic and extrinsic definition of quality is to use the phrase 'subjective feelings or objective facts', which is what it really is in its core essence. This is particularly interesting to the thoughts of this project. The goal is to change a set of parameters of an effect which would yield a different outcome. This is an objective fact; nobody can deny that these factors have been changed. However, when the viewer is not informed about this, he or she will have to determine the factor, in this case describe the quality, by using his or her own subjective feelings about this particular effect. With the right set of guidelines, restrictions and information, this should be a good method for testing the quality of this projects product. But it has to be understood, that the quality is perceived differently by the user and the creator.

7.2 Subjective Test

According to the British Council for teaching English, a subjective test is:

“A subjective test is evaluated by giving an opinion.” (51)

This means that each question which is asked can be interpreted and formulated as the user wishes. This provides a big amount of freedom of expression, and gives a very personal and insightful answer, tied directly to the mind-set of that person. This also means that each question encourages the user to evaluate and question it more than an objective test would. However, when the necessity to solve a question through subjective opinions, eventually leading up to the desire to conclude with a *yes* or *no* answer, subjective opinions are not completely fulfilling, and can be hard to conclude static fact upon.

An example tied directly to this project could be:

“Question 1: describe the quality of effect number one, and effect number two - individually”

This means the user subjectively, through observation will describe their opinion tied to the quality of effect number one and effect number two. However, in the end the goal is to figure out if effect one is actually better than effect two. That is a simple *yes* or *no* question, but it is not as simple as that, the goal is also to figure out if their reasons for believing that effect one is better than effect two, are congruent with the actual differences between the given effects, and *why* the user thinks so, which are questions that can only be answered through subjective evaluations. The user might have an entirely different explanation of why one effect might be better than the other, and if that explanation does not agree with the hypothesis, then it does not validate any of the points which is trying to be proved in this project

7.3 Objective Test

According to The Free Dictionary, an objective test is:

“A test, such as one using multiple-choice questions, in which the feelings or opinions of the person marking it cannot affect the marks given.” (52)

This type of testing is significantly different, and a lot less prone to bias and unwanted influences compared to the subjective test. It is naturally to question the existence of a completely objective test, however changing the specifics of a test can make the method lean towards either end of the scale between subjective and objective. Objective tests deals with static and constructed questions that only allow the user to answer through a list of pre-determined options.

Having to test something that provides you with a subjective impression seem impossible with an objective type of test. However, there are still a multitude of situations where objective testing can be used, especially when combined with a subjective method.

8 Design

8.1 Reference Material

When creating an effect from the bottom up, the challenge lies in how to make it look exactly the way it is supposed to. But what is the reference for success? When creating smoke, there are a large number of ways to portray it, and a large amount of variables which makes the smoke look exactly as it does. Just to name a few - we have; swirl, buoyance, transparency, colour, size and so on. Even though the idea not is to create photorealistic smoke, the factors that determine whether or not the smoke is registered as smoke to us, are almost all of the time taken from real-world memories of smoke, or in few cases created by other virtual productions such as blockbuster movies, which then again are created from real-world memories.

In this section a few images and video sequences have been found and will function as reference material for the creation of the effect that is being designed.

8.1.1 Reference One

The first reference is from a number of oil barrels exploding; the fire here is very volatile and rises far up into the smoke cloud. However it is a great reference for seeing how the fire behaves, and how fast the smoke moves in a very volatile situation, and also to gain an idea of the ratio between the amounts of fire versus smoke in these types of situations. Also, the actual movement and billowing is easy to notice here.



Figure 14 – Screenshot from a movie showing an oil fire. Movie found on www.youtube.com. (53)

8.1.2 Reference Two

The second reference is from an oil and gas fire from the Williston Oil Basin in North Dakota, USA. This smoke is very black and thick, and it contains a little less fire than reference one, but still a great example of an oil fire.



Figure 15 – Screenshot showing massive oil fires in North Dakota. Movie found on www.youtube.com. (54)

8.1.3 Reference Three

The third reference here is from a movie called *Fires in Kuwait*, and is portraying a lot of oil fires from the cold war there. The movie is filled with good references; two of the best looking ones have been pointed out and is shown below. These fires and smoke clouds are very large, and the proportions between fire and smoke appear to be better, especially when the focus is on smoke and not on fire.



Figure 16 – Screenshot showing massive oil fires in Kuwait. (55)



Figure 17 – Screenshot showing massive oil fires in Kuwait. (55)

There is a prominent similarity between these four reference shots, and they make up a great baseline as to guide a smoke and fire creation towards something which behaves and looks like smoke and fire would in the real world.

8.2 Smoke

When having presented the above references, it would be easy to just claim; the smoke should look like that. However there are some requirements to the smoke which naturally needs to be explained even though they are viewable in the references, but there also exist some requirements which cannot be seen.

1. **The smoke should have a point of origin:** This means that the smoke should (like the references) have a fire source or something which can be seen as the natural cause of the smoke. It would look odd if the smoke just appeared out of the blue, especially if the point of origin is not concealed in any way.
2. **The smoke should have a direction:** When changing viewing angle and manoeuvring around a smoke, there has to be some kind of significant variation. Having the smoke rise through some kind of wind or draft and thus rise in a specific direction, solves this perfectly, and gives the smoke some characteristics that makes the movement result in a more visible change within the smoke.
3. **The fire/smoke ratio should be balanced:** The focus of this assignment is smoke, so there should be a much larger amount of smoke than fire inside the animation. The fire should only be there to fulfil point no. 1, to generate a realistic point of origin. The fire/smoke ratio appears to be visually balanced in the reference images I examined; those will act as the guideline for achieving a correct mixture of smoke and fire.
4. **The colour should be reflecting the type of smoke:** An oil fire is characterized by having a very dark and thick smoke; this should also be the case in the creation of this effect. The smoke should be coloured dark grey borderline black. The fire should be bright, and have a distinctive white/yellow/orange colour scale.
5. **Buoyance and swirl/turbulence should reflect the type of source which generates the smoke:** Oil fires burn hard and are intense and hot, thus the smoke should reflect that. This means that the buoyance of the smoke should be high, making the smoke rise rather fast as a result of the heat. At the same time, the turbulence and randomness should also be of a high value as (as explained in no. 2), as it would cause a lot of turbulence and billowing within the smoke.
6. **Should resemble real smoke:** As explained in section 8.1, the smoke should attempt to mimic the characteristics of a real world smoke. It is not the idea to make the smoke look like real world smoke, or to strive for realism, but to mimic the characteristics that makes people recognize that it *is* smoke, even though the elements which contribute to either of them might be the same.

8.3 Surroundings

To be able to determine that there is an actual movement around the smoke effect, there has to be some kind of surrounding or context in which the smoke effect exists. And, just like the smoke effect there has to be some requirements to this surrounding. However, as the surrounding has nothing to do with the actual goal, it has to be kept at a simple level, where it delivers the movement reference that it needs to, but steals as little attention away from the smoke as possible.

1. **Simplicity:** As briefly explained, it is important that the scene is kept simple, the focus here is solely on the presented effect, and the surroundings are only here to aid the user in noticing that there is a movement around the effect. The scene could in theory be totally blank, and whether or not it might pose a small difference. However, as the effect in itself is of a quite complex structure, it makes it harder to notice the movement, and thus the surroundings are there to help the user.
2. **Objects:** The easiest and simplest way to construct a reference for the user, from which he or she can determine the amount of movement around the desired effect, is to create a few simple objects. These objects will function as guidelines, and will not be limited in amount of possible viewing angles as the effect will. Objects like a simple box are suitable for this task, they are simple, and provide clear parallel lines which can be used for perspective reference. More aesthetically pleasing objects can also be used along with textures; the main goal is to create a clear reference through the objects, hence the comfort of using boxes for their parallel lines.
3. **Skybox:** Another element which could contribute to the wholeness of the scene is a skybox. It is simply a box which portrays a sky scenario. This would help put the effect into a somewhat believable environment, and give it some more depth, rather than just being a smoke effect floating in space.
4. **Colours:** Sharp colours tend to steal a lot of attention, thus the various elements of the scene; ground, objects/boxes and skybox should be created with a minimal amount of colours, as well as being kept in a more subtle colour scale.

8.4 Camera

There is not much design to the actual camera function itself, but there are still a few points towards the design of the camera movement which is important.

1. **Symmetric:** The movement of the camera should be symmetric, meaning it should move in a curve around the effect which does not increase or decrease the distance or height at any point. The camera should move in a circular fashion, having the centre of rotation at the centre of the smoke origin.
2. **Smooth:** The movement of the camera should be smooth, there should be no acceleration or deceleration when initiating and stopping camera motion. Additionally, the camera should move with the same speed throughout the whole movement. This is to make sure that the desired angles are reached at a fitting speed, reducing any possible confusion around the desired look of the effect.

8.5 Rendering

Rendering is not of important to the functionality, movement or perception of the scene directly. The only requirements to render options, is that it should be high enough settings to ensure that it does not compromise the functionality of the other visual elements.

1. **Quality:** The render settings should be as high as possible, but still able to produce the product within the given timeframe. The settings contain a few important variables:
 - i. **Raytracing:** Shadows play a big role in portraying the orientation of elements, and thus it is essential that shadows are rendered properly. This is easily done with the raytracing shadow option, thus this is a requirement.
 - ii. **Anti-Aliasing:** Portraying a smooth image that doesn't jitter along the edges of objects or when moving is essential for eliminating elements that can steal attention and remove focus from where it should be. A medium setting of anti-aliasing should be enabled during rendering.
2. **Resolution:** Is it important that the size of the video and effect being produced fits the screen resolution is going to be used for testing. This means that the resolution should be of minimum 720p, and if possible, 1080p.

8.6 Summary

Having established these more or less superficial but important requirements; the scene structure, visual appearance and the settings requirements are in place. This allows the next step to happen, which is actually trying to construct this scene according to the requirements established here.

9 Implementation

This section will explain how the product came to be, from initial design to finished product. It will explain what methods have been used, what elements have fulfilled the design, what elements have exceeded the requirements, and which ones have not been able to live up the requirements. All of the implementations were completed on my own home computer which is considered to be fairly new as of the release date of this report. It contains:

- Intel i5-2500k 3,30GHZ quad-core processor
- 8GB 1600MHZ RAM
- 500GB 7200RPM hard drive

These are the most essential parts of the computer when doing this type of work, combining these information's with some of the timelines provided in the following section should give a better understanding of the amount of hardware stress it takes to create these types of effects.

9.1 Smoke

The smoke in this project was created in *Autodesk Maya* through one of the program features named *Maya Dynamics*. It is created through a relatively low amount of elements, with a large amount of settings and tweaking's.

9.1.1 Emission, Colour and Opacity.

The first thing to do to create your fluid is to create a fluid emitter and a fluid container in which the fluid is being controlled. In this production, two emitters were connected to a polygon sphere which was then acting as the emitter. Figure 18 shows the emitter.

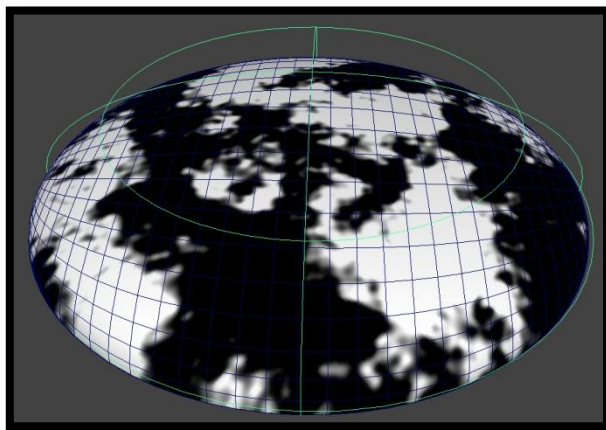


Figure 18 - Screenshot showing the polygon sphere and the attached emitter. Image by the author.

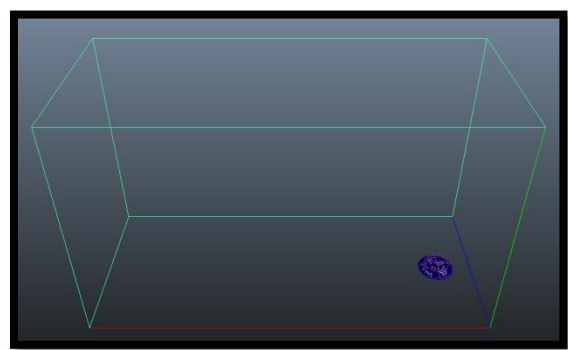


Figure 19 – Screenshot showing the fluid grid box. Image by the author.

Both the emitter properties are tied to the fluid grid box which is shown in Figure 19. All settings which are not directly tied to emission of the various emission types are controlled through the settings in the fluid grid box properties.

The next thing to do after having created the grid box and the emitters, is to define the emission types.

The two emitters were named as:

1. *OilEmitter*
2. *Heat_FuelEmitter*

The *OilEmitter* is emitting the basic density of the fluid voxels which is creating the actual smoke, and is putting out 100000 voxels each second. The *Heat_FuelEmitter* is emitting 20000 heat voxels per second and 15000 fuel voxels each second. While the density determines the actual volume of the smoke, the heat and fuel values added to the volume, but are acting different, both in movement and in colour.

The basic colour of the smoke is set to a constant value, which means the basic density voxels are coloured this way. However the incandescence values are coloured after the temperature values given to each voxel inside the grid. The temperature is determined by the heat released, and the fuel which is burned. Fuel burns when above a certain temperature, after which it slowly burns to a zero value. Heat and fuel thus provides temperature, and when the temperature above a certain range, it colours the smoke through the incandescence ramp. Heat adds a steady temperature increase at the bottom of the smoke, which ensure that the fuel is burned and the incandescence ramp colours the smoke. Once the smoke rises further up and the fuel is used up, the smoke turns a different colour. How long the reaction of the fuel lingers is determined in the fuel options through the *heat released* option, and the *reaction speed* option.

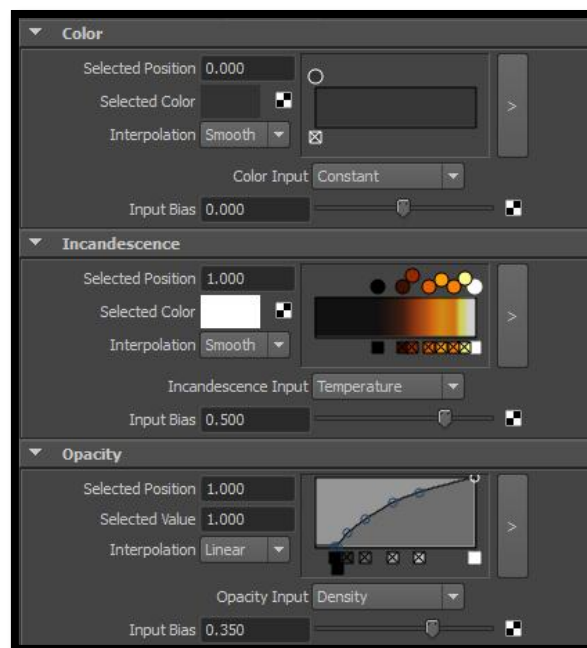


Figure 20 – This screenshot shows the colour ramp, the incandescence ramp and the opacity ramp which are the parameters determining the colour and transparency of the smoke. Image by the author.

As seen in Figure 20, the incandescence ramp gives the colour white to the voxels which have the highest temperature, and the colour black to the voxels which has reached the lowest temperature, with a number of in between values given to voxels with temperatures that lies between the minimum

and maximum temperature. The opacity of the smoke is determined by the density, a high concentration of voxels had opacity of 0, while a low concentration of voxels could reach 100% opacity.

9.1.2 Movement

Having created the actual emission of voxels and provided the reaction between density, fuel and constraints and colouring. The next step is to shape the smoke and allow it to move according to the desired design.

What differs most from the standard settings and the production created here, is the velocity, texturing and custom animation of the effect. The content details are a section in *Maya* which controls these elements, here most of the minor tweaks have taken place, and this includes: density, velocity, turbulence, temperature and fuel. I will not go directly in detail with every little setting as there are around 25 different settings, which all in some minor way affect the look of the smoke.

There are however a few settings which I want to highlight, the first being buoyancy, which is the force that determines the upwards force in fluid on an object, or the weight of the fluid that would otherwise occupy that space. What this basically determines is the up drift of the smoke, and the scale of which the smoke rises. This was set to a value of 8.4, which is a rather high value compared to the standard value of 3.0. Notice buoyancy is both found under density settings, and under temperature settings, this however is the buoyancy of the temperature. The density buoyancy is unchanged at 1.0.

Another element is the dissipation of the density and temperature. This determines the speed of how fast the values dissipate. These have a standard value of 0.0, but are in this production set to 0.4 and 0.5, which makes sure that the container is not being too crowded with the released smoke voxels.

Another element to notice in the movement section of the product creation is that the density emission is animated. In order to get this *burst* effect at the start of the animation, the density emission has 4 keyed frames, at frame 1 the emission is at 100000, at frame 10 it has dropped to 50000 which lasts until frame 15, after that it climbs back to 100000 at frame 20. This can be seen in Figure 21

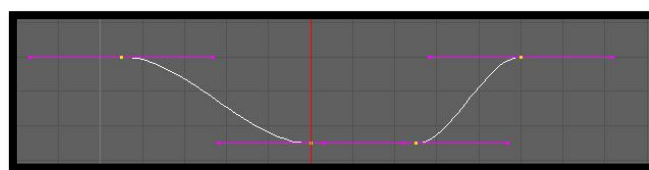


Figure 21 - Screenshot showing the animated emission curve for density. Image by the author.

This is done so that the smoke initially gives of a *burst* of smoke, which scales slightly down during the next frames, when it scales down the initial release of smoke appears bigger and seems to represent the initial ignition of the smoke/fire. After this effect occurs the release of density is scales back to a steady level.

Finally, an emission map has been attached to the density emission of the smoke; this was done though an animated noise texture attached to the emission container. What this does it make sure that the actual parts of the polygon sphere which were emitting voxels, were not always emitting from an equally squared place, but only emitting from the white parts of the black and white noise pattern.

When this is animated it changes texture position every frame. In the end it makes sure that the release of density voxels are more randomly distributed though the initial part of the smoke, giving a more randomized and less static look. This was done by adding a string to the colour of the texture:

$$\text{noise1.time} = \text{time} * .02;$$

This made the value of the texture time change with +0.2 every second. The texture of the polygon emitter sphere can be seen in Figure 22.

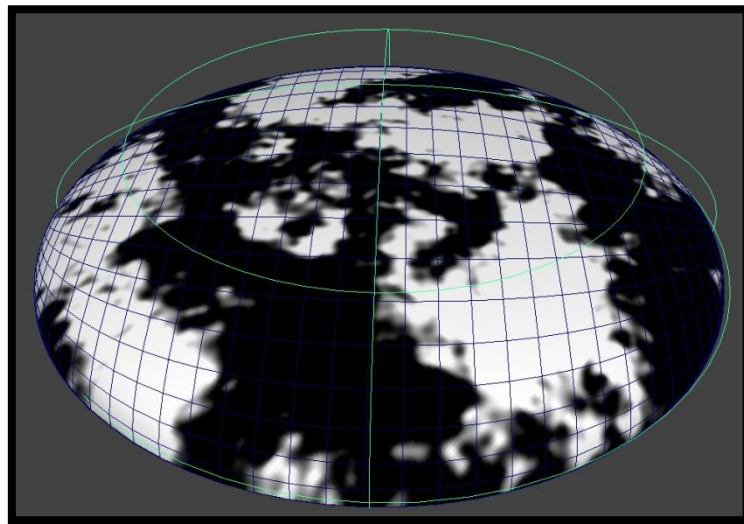


Figure 22 - Screenshot of the polygon emitter sphere. Image by the author.

9.1.3 Wind

The last element which contributed to the movement of the smoke was the fan. It is a simple feature in *Maya*. A fan is simple created through the menu called fields, and then by selecting air. This creates a fan that emits a force which pushes the smoke voxels in a specific direction. I set the direction to blow the smoke directly along the y axis, and then adjust the magnitude to make it fit the size of the smoke; the magnitude got a value of 10.0.

9.1.4 Smoke Detail

9.1.4.1 Resolution

One of the elements which contributes most towards the level of detail and the visual quality of the smoke, is the grid resolution, this setting is as standard set very low, as even small increases can increase render and simulation times by large amounts. However, when raising the grid resolution, not only does the level of detail increase, the smoke moves also different. It does not only increase the visual aspect of the level of detail, but also the level of detail in movement internally in the smoke voxels. This setting is desired to be as high as your computer can manage to solve within the given timeframe. This production was created with a resolution of 350. The resolution are not equally divided though, if the length, height or depth of the grid box are uneven, the resolution will be scaled according to how high the values are for each side. In the initial design for this production, the grid box had a size of 22x13x10 through the x,y,z direction. The final grid solution for production was then 350x206x159.

Another setting which adds to detail is the swirl option under the velocity settings. What swirl does is to add small scale vortices which can add detail to the smoke, this is especially useful for smokes which has an tendency to develop unnatural dampening, or smoke which are not using the high detail solver. The exact details of swirl are hard to understand, and the *Autodesk Maya* help section does not offer much insight. They write:

"Swirl generates small scale vortices and eddies in the fluid. It is useful for adding detail to simulations that are not using the High Detail Solve method. In some cases, high Swirl values can cause artefacts as well as instability in the fluid." (56)

However, I found a value of 25.00 to give the smoke a good amount of extra details, especially for this kind of billowing smoke that resembles the reference shots presented in section 8.1. This is a rather high value, but no artefacts were encountered as warned by the help files.

9.1.4.2 Texture

The last, but important part which were created to add detail to the animation were to add texture to the incandescence part of the smoke. Adding the texture is simple, and is merely done by clicking a checkbox under the texture pane of the grid box, perlin noise were used for the texture pattern, as I found that to be the one closes resembling smoke in this case. Also, I adjusted the gain a little bit to make the texture appear more easily, making the value to be 1.2 instead of 1.0. The texture appears very static though, throughout the animation when not adjusted additionally. This is fixed by first animating the texture time, so that it is given a texture that changes appearance as time passes. This is done by adding a script string called:

```
oilSmoke.textureTime = time *.5;
```

This gives the texture a value of +0.5 every time one second (or 24 frames) passes. That did not appear to be enough though, as the texture changed, it was still too obvious to notice that the texture was only given to the bottom part of the smoke where the incandescence values were being used. To accommodate this, the position of the texture were also animated, so that it looking like it was moving in the same direction as the smoke was moving. The texture would then slowly fade away as the incandescence levels faded away. This was done by animating the x and y positions of the texture with two strings:

```
oilSmoke.textureOriginX = time * -.15;
```

```
oilSmoke.textureOriginY = time * -.15;
```

What these does is to make the texture move -0.15 along the x and y axis each second. This seemed to be a fitting amount to match the movement of the actual smoke voxels. The result can be seen in Figure 23. Notice how the lowest part of the smoke has a lot of texture; this is where the incandescence levels are high. The highest part of the smoke which does not contain incandescence is not being supplied with texture. Keep in mind though, that is a low quality render where the texture is easier to notice and distinguish, and not a render from the final product.



Figure 23 - Screenshot which shows the texture gain of the smoke. Image by the author.

9.1.4.3 Complications

It has to be said that every aspect of *Autodesk Maya* is as great as it sounds. Maya 2011 introduced a new feature which allowed the grid box to auto resize itself, in order to save memory for cache files and to make faster renders. However, as I experienced there are still a few known errors with new features. This particular feature does not work properly with the cache function in *Maya*, the cache feature is a feature which saves the simulation (movement) of the smoke, so that it does not have to recalculate again for every playback and render. This meant that I had to settle with the old technique and work without the auto resize function, as it would cause a voxel stop – this meant that voxels would somewhere in the middle of the simulation stop moving, only in intervals of 10 frames, while the texture kept moving. The result of this was that without this feature enabled, I would have to lower the grid resolution to keep simulation and render times reasonable. I do believe though, that with this error corrected sometime in the future, it would be easy to get a higher resolution grid without also getting a higher simulation and render time.

9.2 Environment

Creating the environment consisted of only a few elements. Creating the actual polygon surroundings in which the effect were to be placed, and creating lights that would lighten up the scene and create shadows.

9.2.1 Surroundings

The surroundings were initially thought to be a squared box, in which a number of polygon models would be strategically placed, so that it would be noticeable when the camera were rotating around the effect, thus creating a point of reference. I did create a squared box in which I placed the effect, and added 8 polygon pylons around the side and edges of the box for reference points. However, because of unforeseen errors in my render output, I was forced to create a sphere environment in which I had to centre the lights, so that there would be no noticeable differences in the environment as the camera rotated around. This error is explained in section 9.3.1.

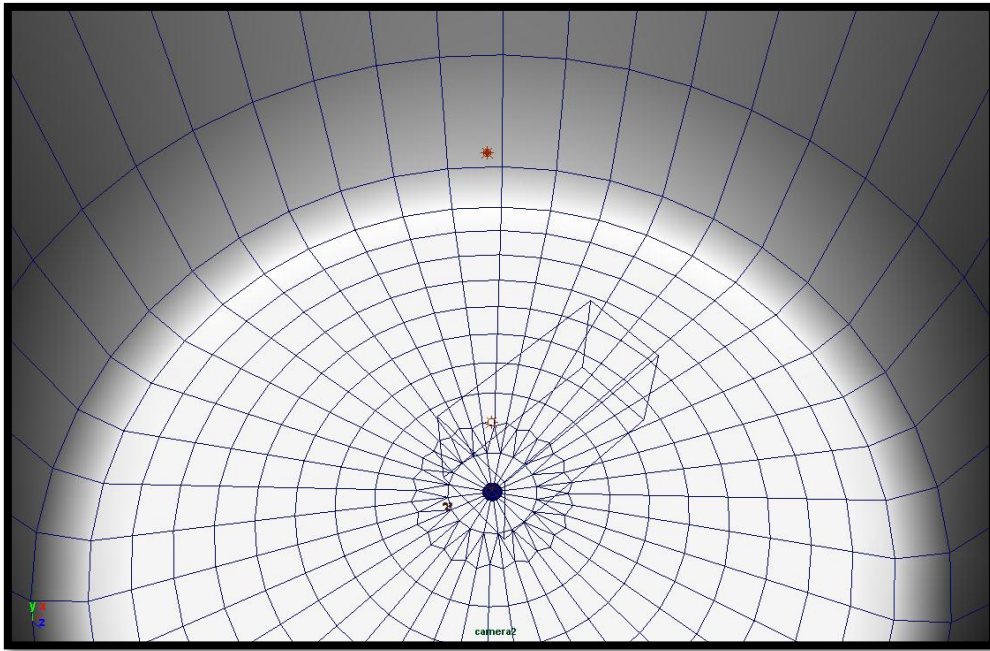


Figure 24 - Screenshot showing the surrounding polygon-dome, the two light sources, the fan, the fluid emitter and the fluid grid box. Image by the author.

Figure 24 shows a screenshot of the final environment used in this production. The environment box is created by modifying a polygon sphere through a series of vertex manipulations. This was done to keep the rounded edges but to attain a flat floor for the effect to be placed on. The nuance light grey were chosen to keep the level of distraction to a minimum, as I felt a bland and colourless nuance would gain less attention than many of the alternatives.

9.2.2 Lights

Two light sources were chosen for the environment, a point light and an ambient light. Both light sources were placed directly above the centre of the fluid emitters. Ambient light was only there to provide a subtle fill light for the environment, and had shadows disabled. The point light was chosen to light up the fluid and to provide a shadow below the fluid. It had ray traced shadows turned on, with the default settings. These two lights together provided sufficient light to the scene, and gave a shadow which was enough for the design requirements.

9.2.3 Camera

Creating the camera and setting it up correctly were crucial for the outcome of the videos to be testable. If the camera positions were not adjusted correctly towards the requirements, the frames would not fit the amount of degrees they were supposed to skip, and the test results would be biased.

The smooth camera was created and adjusted so that it would provide as much of the smoke as possible, as well as a good view of the emission spot, while not showing any of the grid edges to be visible, as they would break the illusion as the smoke would start clipping and disappear out the edges. This was done by adjusting the position of the camera. It was moved directly away from the emitter, moved slightly upwards, and then moved to view in an angle of 10 degrees downwards.

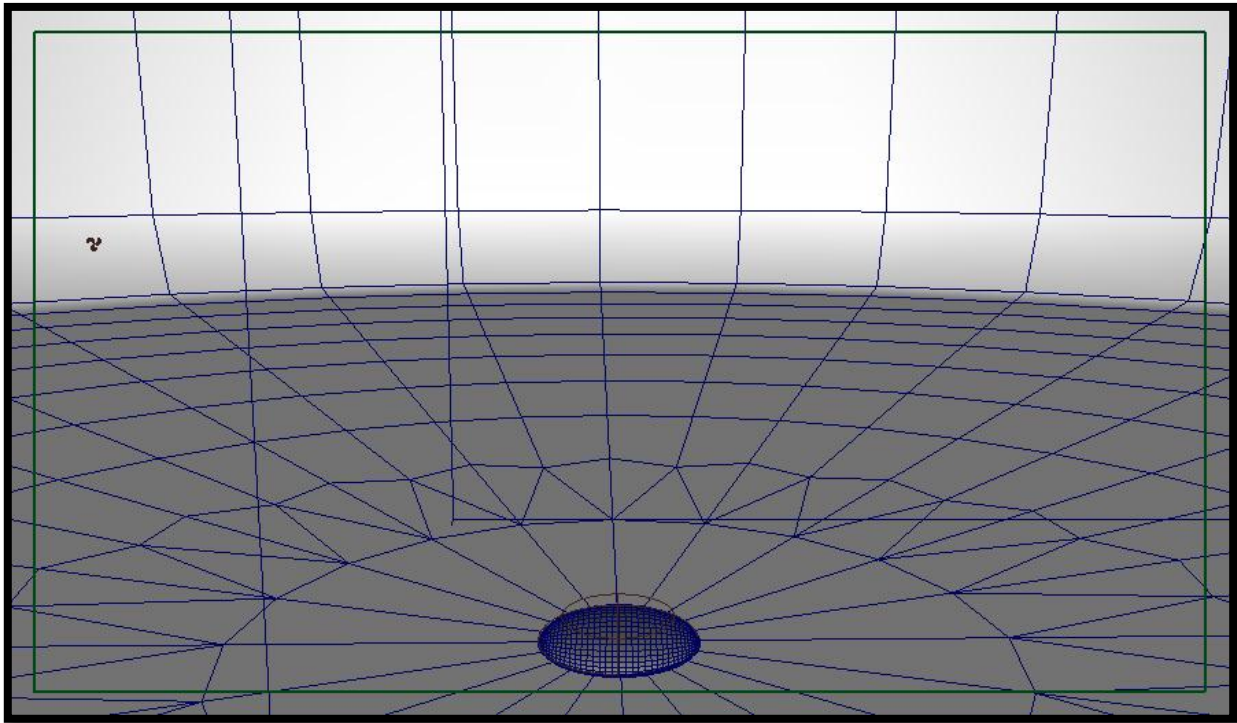


Figure 25 - Screenshot showing the camera view, the green edge shows where the camera view starts. Image by the author.

Figure 25 shows the camera view inside *Maya*, the green edge shown at the outside of the image is the actual camera resolution gate which shows how much of the view inside *Maya* which the camera is actually recording.

After having adjusted the position of the camera, the only thing left were to create the movement for each of the cameras recording each of the effect versions. The camera for the first version were simple, it were keyed at frame 1 at degree 1, and at frame 360 at degree 360. This meant that the camera would move around steadily throughout the animation, ending up having exactly one frame at every angle. To make sure the camera did not smoothen out the movement in each end of the animation, the tangents for each key were set to linear so that they would end without smoothing the movement out.

The camera for version two and three were created in a similar fashion. A single step were created as it was with the camera for version one, however here the time step would only reflect one of the jumps between angles, and were then set to be repeated infinitely. For version two, the camera would then be keyed to record two frames at angle 2, and then move on and record two frames at angle 4.

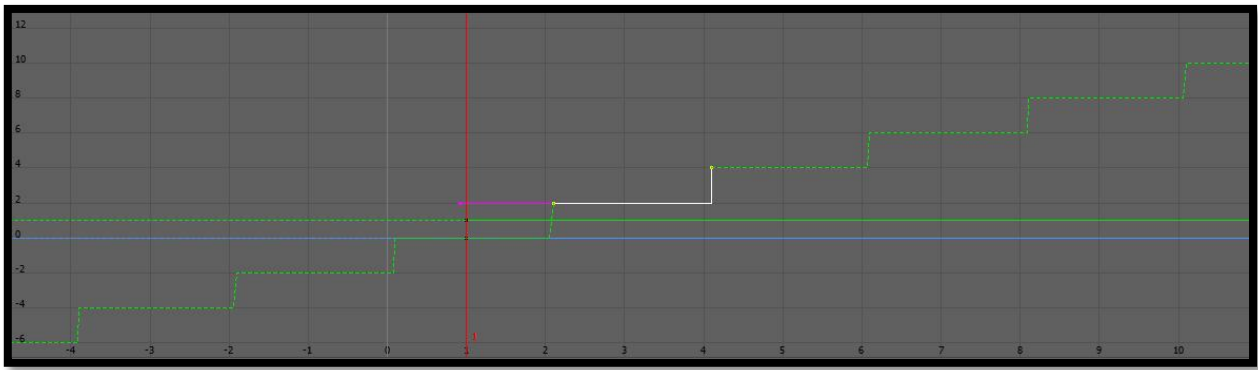


Figure 26 - Screenshot showing the time steps from the camera recording effect version 2. Image by the author.

Figure 26 shows a screenshot from *Maya* where the time steps from the camera recording effect version two are keyed. As you can see it moves in steps just after every second frame, and then jumps two degrees instead of one.

The last camera for effect version three were created in the very same way, only here the camera would record five frames, and then skip five degrees in the rotation before recording another five frames.

9.3 Rendering

The render options for this effect were not changed much. The default quality setting called production was chosen for quality, and the resolution was set to be HD 720. This was the requirement in the design. HD 1080 was desired, but with limited qualities of monitors available for testing, it did feel necessary considering the additional render time. The output was set to be uncompressed PNG files, thus there would be minimal data loss in the images when rendering them out. Mental ray was chosen as render engine as it is faster when working with ray traced shadows. It was also chosen partly because I was tampering around with specific mental ray shaders that would only render through mental ray; however those were not chosen in the end as they were felt unnecessary.

9.3.1 Complications

The reason for not choosing a squared environment with a set of objects as references were to be found in an unexplainable error I encountered in my render results. When I rendered the environment and the smoke effect together, everything would look as it should, but when I had to separate the environment and the effect in order for the effect to appear as it was skipping degrees (which is the whole point of the experiment) I got an error regarding the transparency of the smoke. It appeared to inherit some of the environment colour from the camera into the actual opacity map of the effect. So that if I rendered my smoke effect and its shadow with a black background, when it was composited with the environment, it would get a black aura around the whole effect, and if rendered with white, it would get a white aura. This problem persisted and after a lot of time spend on the web, through guide, and in the help documents I were not able to get rid of the issue, and therefore had to adjust my composition.

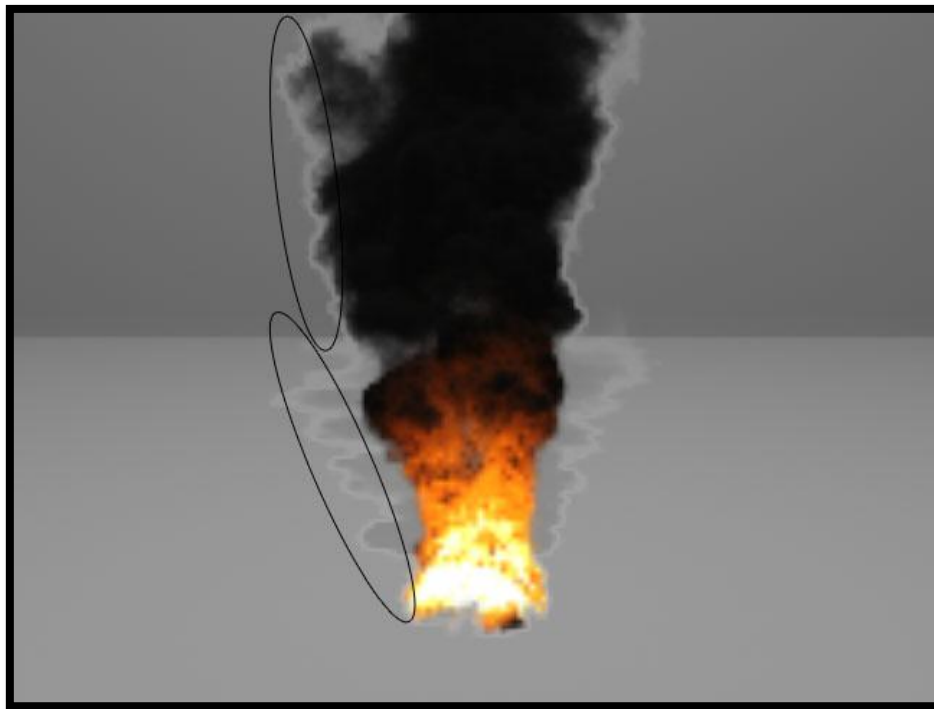


Figure 27 - Rendered image showing the ghosting problem, here with a white aura highlighted by a black circles. Image by the author.

Figure 27 shows the problem on an image which was composited together from different renders. One render with the effect and its shadow, and one render of the environment. This felt as a necessary adjustment to make, as I believe the ghosting aura would cause a lot more confusion and bias in the test results, compared to what I would gain from having a different environment.

9.4 Compositing

Creating the actual video used for testing the production were done in *Adobe Premiere CS5*, (57) which is a compositing and video editing software. The composition was very simple; a few black title screens were created, simply stating the effect versions. The effects were created and spacing between them was added, and an opacity and colour adjust effect was created to adjust fading and colouring of the shoots for a smoother transition.

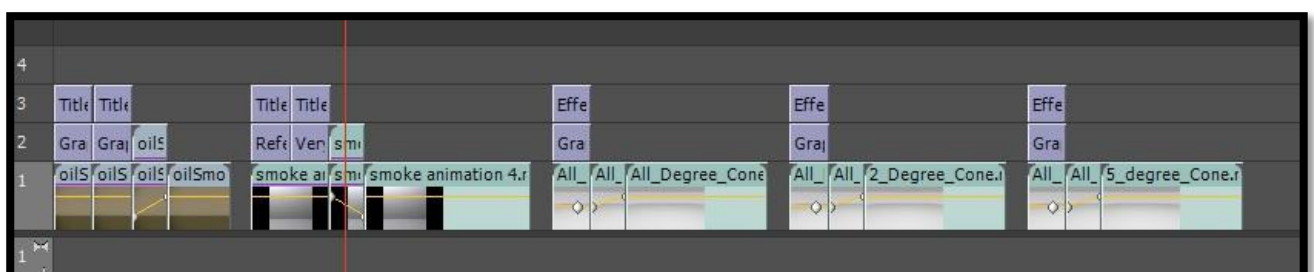


Figure 28 - Screenshot showing the timeline in *Adobe Premiere*. Image by the author.

Figure 28 shows the timeline of the video used for testing. In the timeline you can see the video clips, the title screens and a small piece of graphic (a black bar) which were placed between the video sequence and the title for easier readability. The two small video clips placed in front of the actual effect video are simply copies of the effect videos which are frozen at frame one, at stays that way until the presentation with the title screens have run.

9.5 Time

Producing effects without professional equipment gives a fair amount of insight into the frustrations that can occur, once an effect has been set to simulate or render, and the wait has commenced. This was no exception, the whole process from initial design to fluid caching and final render all took a great amount of time. Initially the fluid has to be shaped and watched closely while settings are fine-tuned; the dilemma here lies in waiting for a full fluid cache to finish seeing the movement, or waiting a medium amount of time to view a fraction of the sequence. What is most frustrating is that it is hard to fine-tune the settings with low quality as quality also affects the movement. Once this is in place, the fluid motion has to be cached; otherwise the software needs to recalculate the motion for every render. Finally, the fluid has to be rendered into the desired amount of versions. To elaborate a bit further, note that this projects final renders, the caching for both versions took around ~8-9 hours each, while each render took around ~5-6 hours. This is excluding error versions, changes and other elements which might have caused the process to be restarted.

9.6 Optimizations

Having gotten the final output video sequences, it was now possible to calculate the amount of memory which could actually be saved by utilizing this project method in a virtual environment. One sequence had the size of 20.8MB, this means that an effect which should have the possibility to be viewed from start to end from every angle (360 angles), ends up on 7488MB, which is a lot. However by using the two degree version, memory usage would already be reduced to 50%, or 3744MB. Using the five degree version would reduce the memory usage by 80% from the original version, an end up only using 1497.5MB, which is a big cut in memory usage. However, these numbers are unusually large as my videos are not compressed in the same manner as they most likely would be, should they be implemented in a game or virtual environment. Yet the savings in percentages would remain the same.

10 Test

10.1 Clarification

This section of the report has a lot of references towards different tests, different effects, and different versions of these effects. Therefore, I will shortly describe and name these tests, effects and versions to minimize any possible confusion. The tests will be referred to as *test 1* – the *fan* test, which was the test with the original effect design, and *test 2* – the *cone* test, which was the test with the alternative effect design.

Test 1 – <i>Fan</i> test	Test 2 – <i>Cone</i> test
<ul style="list-style-type: none">• Version 1 – All degrees (<i>Fan 1</i>)• Version 2 – Two degrees (<i>Fan 2</i>)• Version 3 – Five degrees (<i>Fan 3</i>)	<ul style="list-style-type: none">• Version 1 – All degrees (<i>Cone 1</i>)• Version 2 – Two degrees (<i>Cone 2</i>)• Version 3 – Five degrees (<i>Cone 3</i>)

The text in italics will be the names used for reference in the rest of this section.

The first test were conducted which were utilizing the original design that was intended for the smoke effect. However, as the results happened to be rather surprising, a second test were conducted with an alternative, however similar effect. This was done to examine any alternative uses for the efficiency method explained in this report, as the method did not yield the expected results in the first test.

10.2 Procedure and Design

The design of the actual test was the same for both the *fan* test and the *cone* test. Also, the procedure which the test were conducted with was the same for both.

10.2.1 Design

The test consisted of two parts; a video which were shown to the subjects, and a questionnaire which were to be filled out afterwards.

10.2.1.1 Video

The video consisted of 5 different elements:

1. **Good reference:** This was a reference shot which were shown at the beginning of the video. It was there to establish a visual impression of a smoke effect which was (by me) considered to be of very good quality. The video was taken from the Gnomon Workshop (58) and were designed much in the same way as my own effect, however with higher render settings and a higher grid resolution.
2. **Bad reference:** This reference shot were shown afterwards to establish a visual impression of a smoke effect which was poor quality.
3. **My effect, version 1:** After having established a set of loose guidelines as to what should be considered good and bad visual appeal, the first version of my effect were shown. This was the version which had a full rotation, and did not leave any angles out.
4. **My effect, version 2:** Following that, my effect was shown in its second version. This version only showed the effect from every second angle.

5. **My effect, version 3:** Ending the video sequence, the third version of my effect was shown. This was the effect which was only shown from every fifth angle.

The reference shots were considered good and bad looking based on a large variety of aspects, and not only as a result of render quality. The quality drop in my effects were as a result of fewer viewing angles, and even though that was not same a part of the quality drop that occurred between the reference shots, it felt as a good method for establishing a visual impression of what functions well as an effect, and what functions poorly. It was not meant to be a strict guideline towards how they should rate the effects from this project, but more of a casual reference. As seen in Figure 29 and Figure 30, there was a very distinguishable difference between the reference shots.



Figure 29 - Screenshot from a video sequence portraying a smoke effect of 'very good' visual appeal. (58)



Figure 30 - Screenshot from a video sequence portraying a smoke effect of 'very bad' visual appeal, found on www.youtube.com. (59)

During the test it felt natural for the test subjects to rate the quality of my effects. Additionally, the test subjects were instructed to rate the whole environment (my scene was really not containing more than the smoke and background) as being a part of the effect, so that they also gave thought to elements such as movement, flow, render settings, colour and so on, including as many elements into the actual the rating of visual appeal. More or less all the test subjects understood that, and rated (if noticed) the jagged camera movement as a part of the visual appeal of the effect. So the reference shots in the video appeared to function as intended, and did not confuse or provide the test subjects with any skew guidelines for grading the visual appeal of my effects.

10.2.1.2 Questionnaire

The questionnaire was handed out to test subject after they had finished watching the video. It consisted of 5 major questions directed towards the video, and 4 minor questions directed towards their personal information, as well as their experience in creating visual effects.

- **4 minor questions:** Age, sex, education and experience with visual effects creation. These questions were asked merely as a backup. If a test subject had answered something unusual towards my effects, it could be as a result of the person being educated in something specific, being a professional visual effects artist or simple because the person had an age outside the norm of the test subjects. Therefore these questions were added.
- **5 major questions:** These questions were divided into two types, the first three inquired about the visual appeal of my effects, version 1, 2 and 3. The last two questions inquired the test subject if he or she noticed any difference between effect version 1 and 2, and version 1 and 3. Each of these questions was followed up with a side note that allowed the subjects to specify the reason as to why they had answered why they did.

Both the questionnaire and all the specific data which were gathered during the test can be found either in the appendix in section 17.1 or on the attached CD.

10.2.2 Test 1 – The *Fan* Test

The first test, also internally known as the *fan* test, was conducted first. 28 test subjects went through the process; saw the video and answered the questionnaire. It was referred to as the *fan* test because it was using the video version, where the smoke is affected by a virtual fan inside Maya; this gives the smoke an angle and makes it blow in a specific direction, very noticeable in Figure 31.



Figure 31 - Screenshot from the *fan* version of the smoke effect - Image by author

10.2.3 Test 2 – The *Cone* Test

The second test, the *cone* test, were conducted with the exact same procedure, however it were using a different set effect. The effect had no fan, and was instead rising straight up in a cone like shape, see Figure 32 . 24 test subjects tested this version.



Figure 32 - Screenshot from the *cone* version of the smoke effect - Image by author

10.3 Observations and Results

This section will explain the conclusive elements which were discovered during the two tests. However, as there are a lot of different ways each small element can be interpreted, only those found most conclusive, interesting and obvious will be presented here. For a complete insight into the specific test results, I refer to the attached CD.

10.3.1 Test 1 – The *Fan* Test

The *fan* test had 28 subjects. This was the test where the subjects were watching the effect designed after the original intentions. However, as you will see, this test did not confirm the hypothesis. The test results showed that the majority of subjects' noticed the difference in viewing angles, even in the *fan 2* version which was at only 2 degrees.

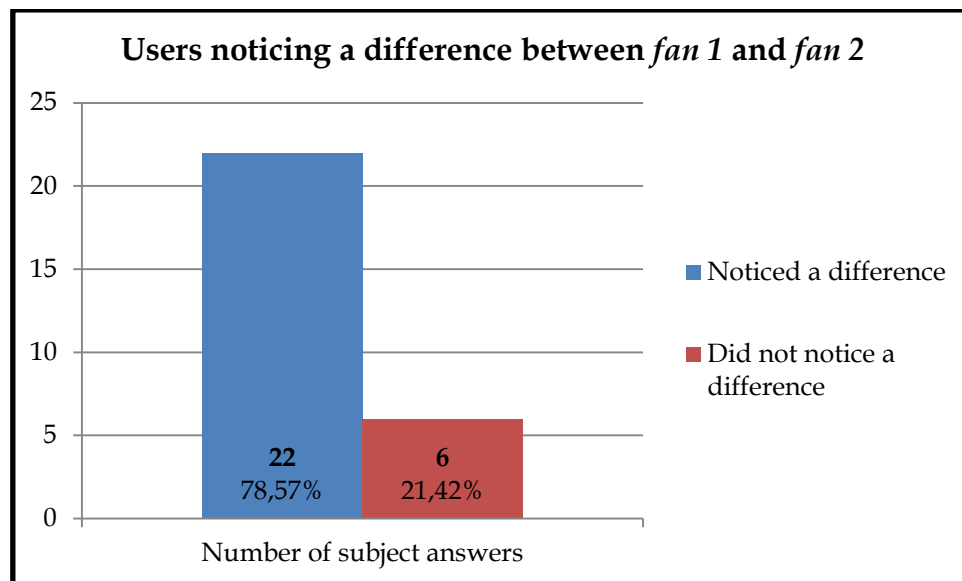


Figure 33 - Figure showing the amount of test subjects who noticed a difference between *fan 1* and *fan 2*.

As seen in Figure 33, the amount of test subjects who noticed a difference between *fan 1* and *fan 2* were a total of 22, which is 78.5% of all the test subjects. The comments towards these answers were also analysed, and it shows that *all* 22 subjects who noticed the difference, actually noticed the jagged movement as the difference. In the remaining 6 subjects, 5 of those actually gave *fan 2* a lower score than *fan 1*. So even though they answered *no* to noticing a difference, they still rated *fan 1* lower than *fan 2*, meaning that they somehow managed to perceive *fan 2* as worse. That leaves only one subject who did not rate *fan 2* lower, or noticed a difference between *fan 1* and *fan 2*.

These results are clear, *fan 1* had an arithmetic mean of 4.5, and *fan 2* had 3.00, so the difference is obvious there. *Fan 3* scored as low as 2.42, and it is not a result of a large spread in answers, as the medians helps indicate; *fan 1* scored 4.5, *fan 2* scored 3.0 and *fan 3* scored 2.0.

If we look at the same question between *fan 1* and *fan 3*, the answers are very clear. 25 out of 28 noticed a difference between the effects, in the remaining three, all of them still rated *fan 3* as having less visual appeal, and one of the remaining three even commented on the jagged movement. This means that all 28 rated *fan 3* lower, or noticed the difference.

10.3.2 Test 2 – The Cone Test

The *cone* test was conducted with a different set of effects, here to smoke rose directly up into the air. It was my belief, that the extreme angle of the smoke in the *fan* version, made it too easy to notice the difference between the angles. Therefore this second test was created, to examine if it would be any different with a more centre oriented effect.

Results in the *cone* test are different from the first test. The most peculiar result is; that *cone 2* was given a higher rating than *cone 1*, meaning that even though the effect is the same just with lower viewing angles, it scored a higher average than *cone 1*. *Cone 1* got an arithmetic mean of 4.08, *cone 2* got a score of 4.20, and *cone 3* got a score of 2.79. The scores of *cone 2* and 3 are both higher than the values in the *fan* test, even though the score of *cone 1* has a lower value than *fan 1*. Medians are also showing different numbers, rating *cone 1* with 4.00, *cone 2* with 4.00 and *cone 3* with 2.50.

What is also very interesting in the *cone* test is that the results taken from the question which inquires about whether or not the subjects noticed a difference between *cone 1* and *cone 2*. Unlike the *fan* test, here a slight majority of the subjects did *not* notice a difference. This is illustrated in Figure 34.

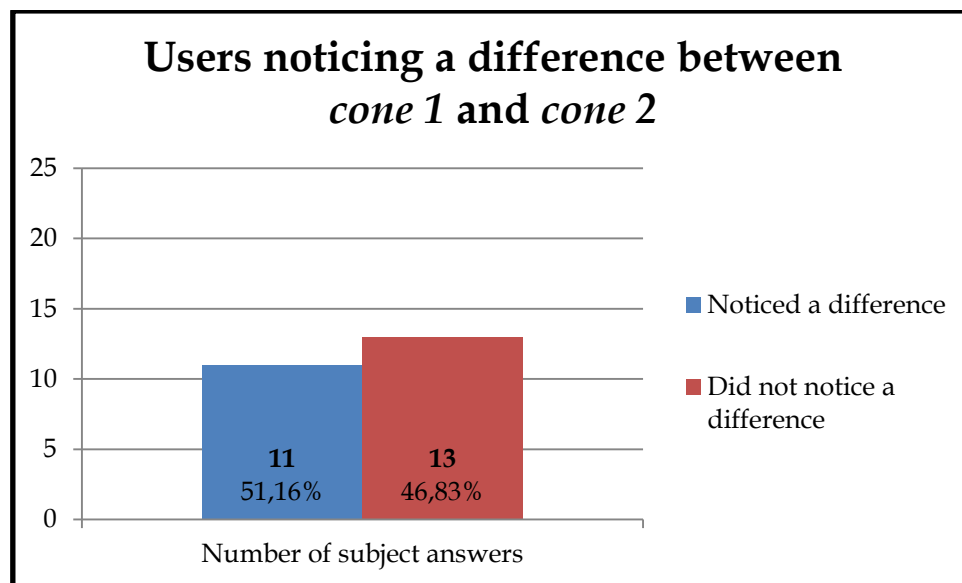


Figure 34 - Figure showing the amount of test subjects who noticed a difference between *cone 1* and *cone 2*.

Every single answer was followed up by a comment as well, and examining those closer, the vast majority of the reasons for noticing a difference are not related to the fewer viewing angles, but are referring to changes in the smoke which are actually not happening, but elements which the subjects subconsciously has made up. In Figure 35, a breakdown of who actually noticed the lower viewing angles (or *jagged movement*, as a lot calls it), is illustrated. Almost 92% of the test subjects did not notice the *jagged movement* as the difference.

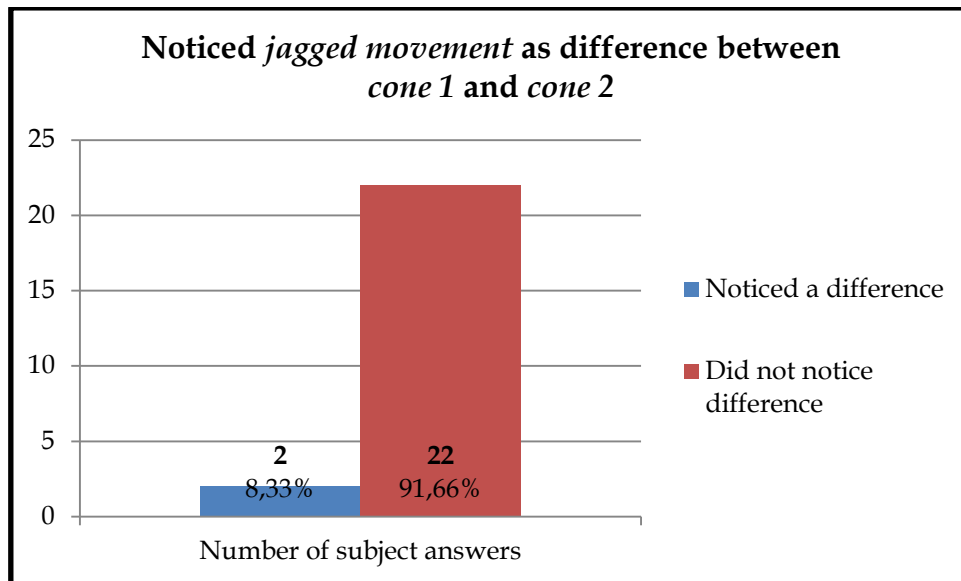


Figure 35 - Figure showing the amount of test subjects who noticed *jagged movement* as a difference between *cone 1* and *cone 2*.

Moving on to the difference between *cone 1* and *cone 3*, the results again start to contradict the hypothesis. 18 test persons noticed the actual difference between the versions, which is exactly $\frac{3}{4}$ of the test subjects in the *cone* test.

10.3.3 Comparison

These results between *fan 1* and *fan 2* in the *fan* test, and *cone 1* and *cone 2* in *cone* test are almost opposite. Combining values and comments; only one person in the *fan* test did *not* rate *fan 2* lower or noticed a difference. In the *cone* test, only two persons *did* notice the actual difference between the *cone 1* and *cone 2*. Also it is important to notice the actual values which were given to them. The arithmetic mean from *fan 1* to *fan 2* in the *fan* test dropped around 1.5, while it in the *cone* test actually rose 0.2, making *cone 2* the one with the best rated visual appeal in the *cone* test.

The difference between version 1 and 3 appears more equal between the two tests. In the *fan* test, the values continue their drop and ends up on 2.42, which is the lowest score of both tests. In the *cone* test, *cone 3* gets a drop as well, and ends up on 2.79. Figure 36 shows a graphical breakdown of the arithmetic means of both tests.

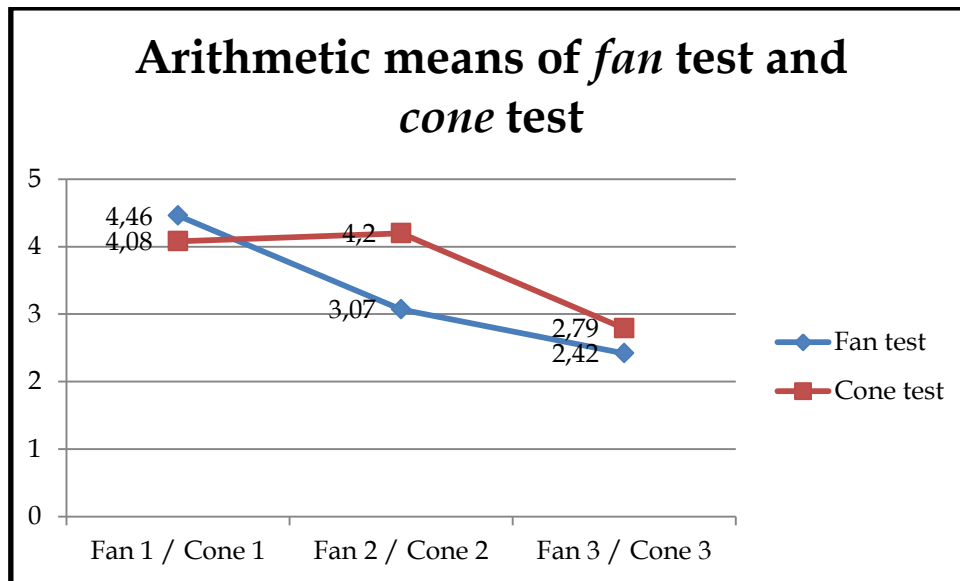


Figure 36 - Figure showing the arithmetic means of the *fan* test and the *cone* test.

These were the most significant results from both the *fan* test and the *cone* test. There are other minor differences and results which can be highlighted, but I do not consider them to be of any influence towards the major results.

10.3.4 Discussion

There are of course a lot of speculations as to why the results ended up as they did, and why the fan resulted in a disproval of the stated hypothesis. There are three main areas which i believe contributed to this: context, focus and design.

10.3.4.1 Context and Focus

These two elements go hand-in-hand. When talking about context, it is with referral to the actual environment and situational context which the effect is placed within. In this test the centre of attention is the effect, there are no alternative context or environment which can steal or alter the perception of the effect. This means that all the subjects focus is placed upon the effect. This can in theory be why the subjects has such an easy time (particularly in the *fan* test) noticing the reduced amount of viewing angles. However this has to be verified with additional testing and theory.

10.3.4.2 Design

This means the design of the very effect itself. Here the *fan* inside Maya is what does most of the difference. It angles the smoke, so that the reference points both in the smoke and in the shadow are easier to notice. This is particularly true at the points in the video where the smoke faces towards the camera and away from the camera; this happens around frame 90 and frame 270.

11 Conclusion

This project was initiated with the simple question:

"To what degree can the creation of virtual effects such as smoke or fire be simplified while still maintaining the same graphical satisfaction of the viewer?"

This was based on a personal concern that producers of virtual environments and digital games were leaping forward in a tempo, which could not always be matched by the consumer in terms of hardware performance. A research was initiated to explore methods which could maintain the graphical quality, while keeping the hardware stress within an affordable margin. Once the various solutions were narrowed down, a hypothesis were stated based on personal experiences, it sounded:

"...it is the belief that even with minimal elements in a scene in addition to the effect under examination, there has to be significant changes in the quality of the effect for the user to notice."

Attempting to prove that hypothesis right, a range of effect with various qualities were produced, and screened individually towards an audience. The audience were asked in a multitude of ways to rate and explain the qualities of these effects.

It was proved that the effects produced in regard to the original design, were not able to disguise the drop in quality at all. Almost the entire audience experienced the quality drop, both in the effect containing a larger drop in quality, and in the one containing a small drop in quality. This disproved the hypothesis and showed the exact opposite of what was initially thought. A second test was conducted on a series of effects with a similar, but alternative design. This time around the results were different. The results showed that with an effect containing a small drop in quality, almost the entire audience did *not* notice the drop in quality; some even perceived an imaginary raise in quality. However, once a certain threshold were exceeded, the audience again perceived a drop in quality, thus the boundaries of how much the quality can be adjusted appeared to lie somewhere in between the two versions (two degrees and five degrees), at least for this setup.

Conclusively, this proves that a technique of reducing the amount of possible viewing angles of a billboard effect can be effectively used without the user perceiving a drop in quality. Still, the limitations of the method must be acknowledged.

12 Discussion

Despite the fact, that valid conclusions can be drawn and new information discovered from the test results in this report, there still exist elements of this product, test and conclusion which are less than ideal, and do not provide completely unbiased results towards any actual usage or benefit that this project might provide. The initial thoughts revolving around this are already presented shortly in section 10.3.4, it is clear that the specific scenario created during this project yielded a list of somewhat useful results, one way or the other, but if the concept and benefits should be used in a professional production, then the nature of this project has to be changed, or at least developed and tested further, in order to explore more alternatives and additional ways of testing this technique in an even more specific and relevant scenario and context.

There are two main areas in which I believe that this test could be improved and developed further in; context and diversity.

Context means the actual environment in which the effect is tested, digitally, not physically. This would mean that the effect were to be tested in a vast amount of scenarios, and not just inside a round sphere. This would put the focus element into the equation, how would the effects be rated when there are alternative virtual elements included in the environment? How would the effects be rated if there are other users, other effect or perhaps if the environment itself changed during the experience? It is a completely different study to examine how attention and focus is distributed during a user's interaction with a game or virtual environment, yet one easily imagines that this could contribute in both a positive and negative way of how effects are perceived.

Diversity refers directly to the selection of effects, and not the environment in which they are put. The plan here would be, to not only use one or two different types of effects, but to provide a larger amount of effects and a list of effects with an even bigger difference in behaviour and appearance. This would aid in specifying the limits of what the presented technique can't be used, but also broaden the view of where the technique *can* be used. There are other effects than smoke or fire which can benefit from these solutions, but even within the range of smoke and fire effects, there still exist a large amount of different types of effects which are still unexplored.

These two areas are both span wide, and to accommodate for the interest in both, it would take an excessive amount of work to integrate them into a suitable solution. Nonetheless, I am without doubt sure, that the overall usefulness of the technique presented in this project could be heightened, and the results provided through the test could be strengthened additionally.

I would not be as bold to say that these are in any way flaws in this project method, however they are important factors when considering the actual usefulness in professional production, should this technique at some point make it that far. The issues stretch far, and it would be an absolute dream scenario to reach it all.

13 Perspective

There are a number of people and companies which might contribute from results such as those presented in this report. As already written in section 5.8, the target group were either producers of virtual environments for home use, with high standards for quality, as well as producers of virtual environments where hardware is limited. Especially the latter, where hardware capabilities are fixed and un-upgradable, it would be ideal to be able to mix and match between various display methods such as the ones presented here, to create the ideal graphics settings for the given device.

Once tested fully through additional elements such as those presented in section 12, this research could be useful, and be a helpful tool for developers wishing to create good looking visual effects with minimum hardware stress. This research will continue to be relevant, and methods like this are, and will continue to be used until algorithms and hardware solutions are reliant and fast enough to provide effects like this real-time. Small hints of this are currently happening, however in the foreseeable future, I expect that techniques like this will still be used in a majority of virtual environments.

14 Future Development

Continuing the work which has been conducted in this project, I would initially accommodate for some of the thoughts presented in section 12, and attempt to create a different scenario in which these effects could be presented. Using an actual game or simply using a video sequence with a more dynamic camera movement would be an ideal scenario to proceed for additional testing. It is still my belief that the boundaries and limitations of this technique can be pushed even further under more natural and more dynamic conditions. Additionally, considering the significant change in test results between *fan 1 & 2* and *cone 1 & 2*, it would also be interesting to continue to search for the exact reason for this, and to attempt to integrate it further into a new test, to attempt to push the limits as to how much the effects can be reduced in quality without the user noticing.

15 List of Figures

Figure 1 – A screenshot from <i>Doom 2</i> by <i>id Software</i> , copyright © 1994, showing an early usage of image-based rendering, which is portraying a character and two fire effects. (4)	7
Figure 2 – Image showing the example of an environment map for a skybox. (7)	8
Figure 3 - Screenshot taken from the game <i>Command & Conquer: Red Alert 2</i> by <i>Westwood Studios</i> , Copyright © 2000, showing a great example of how sprites can be used to mimic 3D objects. (9).....	9
Figure 4 – Screenshot from <i>Counter-Strike</i> by <i>Valve</i> , Copyright © 1999.	11
Figure 5 – Image showing the difference between view plane and viewpoint billboards. (14)	12
Figure 6 – Screenshot from <i>World of Warcraft</i> by <i>Blizzard Entertainment</i> , Copyright © 2004, portraying various grass and tree billboards.	13
Figure 7 – Image showing the differences between a real 3D object and its corresponding imposter. (16).....	14
Figure 8 - Image showing a 3D football player and its corresponding imposter. (18)	15
Figure 9 - Image from their report showing how their use of imposters gradually replaces more of the actual model. (21)	17
Figure 10 - Image showing the effects on hardware by changing method for portraying effects. Image by the author.	18
Figure 11 - Screenshot from <i>Counter-Strike</i> by <i>Valve</i> , Copyright © 1999, showing how shadows are presented. (25).....	18
Figure 12 - The <i>Navier-Stokes</i> formula. (40).....	25
Figure 13 - Screenshot from <i>Call of Duty: Black Ops</i> by <i>Activision</i> , Copyright © 2010, showing a fire and smoke effect which appears to be created through a particle billboard system. (49)	28
Figure 14 – Screenshot from a movie showing an oil fire. Movie found on www.youtube.com . (53)	33
Figure 15 – Screenshot showing massive oil fires in North Dakota. Movie found on www.youtube.com . (54).....	34
Figure 16 – Screenshot showing massive oil fires in Kuwait. (55).....	35
Figure 17 – Screenshot showing massive oil fires in Kuwait. (55).....	35
Figure 18 - Screenshot showing the polygon sphere and the attached emitter. Image by the author.	39
Figure 19 – Screenshot showing the fluid grid box. Image by the author.	39
Figure 20 – This screenshot shows the colour ramp, the incandescence ramp and the opacity ramp which are the parameters determining the colour and transparency of the smoke. Image by the author.	40
Figure 21 - Screenshot showing the animated emission curve for density. Image by the author.....	41
Figure 22 - Screenshot of the polygon emitter sphere. Image by the author.....	42
Figure 23 - Screenshot which shows the texture gain of the smoke. Image by the author.	44
Figure 24 - Screenshot showing the surrounding polygon-dome, the two light sources, the fan, the fluid emitter and the fluid grid box. Image by the author.....	45
Figure 25 - Screenshot showing the camera view, the green edge shows where the camera view starts. Image by the author.	46
Figure 26 - Screenshot showing the time steps from the camera recording effect version 2. Image by the author.	47
Figure 27 - Rendered image showing the ghosting problem, here with a white aura highlighted by a black circles. Image by the author.....	48

Figure 28 - Screenshot showing the timeline in <i>Adobe Premiere</i> . Image by the author.	48
Figure 29 - Screenshot from a video sequence portraying a smoke effect of 'very good' visual appeal. (58).....	51
Figure 30 - Screenshot from a video sequence portraying a smoke effect of 'very bad' visual appeal, found on www.youtube.com . (59)	51
Figure 31 - Screenshot from the <i>fan</i> version of the smoke effect - Image by author	53
Figure 32 - Screenshot from the <i>cone</i> version of the smoke effect - Image by author	53
Figure 33 - Figure showing the amount of test subjects who noticed a difference between <i>fan 1</i> and <i>fan</i> <i>2</i>	54
Figure 34 - Figure showing the amount of test subjects who noticed a difference between <i>cone 1</i> and <i>cone 2</i>	55
Figure 35 - Figure showing the amount of test subjects who noticed <i>jagged movement</i> as a difference between <i>cone 1</i> and <i>cone 2</i>	56
Figure 36 - Figure showing the arithmetic means of the <i>fan</i> test and the <i>cone</i> test.	57

16 References

16.1 Bibliography

1. **Jackson, Peter.** *Lord of The Rings Trilogy*. New Line Cinema, 2011.
2. **Akenine-Möller, Tomas, Haines, Eric and Hoffman, Naty.** *Real-Time Rendering*. s.l. : AK Peters, 2008.
3. **Heung-Yeung, Shum and Kang, Sing Bing.** *A Review of Image-based Rendering Techniques*. s.l. : Microsoft.
4. **The Doom Wiki.** The Doom Wiki - Arch-Vile . *The Doom Wiki Web site*. [Online] 2011.
<http://doom.wikia.com/wiki/Arch-vile>.
5. **id Software.** id Software - Doom 2. *id Software Web site*. [Online] 2011.
<http://www.idsoftware.com/games/doom/doom2/>.
6. **Farlex, Inc.** The Free Dictionary - parallax. *The Free Dictionary Web site*. [Online] 2011.
<http://www.thefreedictionary.com/parallax>.
7. **Carsten Fuchs Software.** Cafu Engine - Skybox example. *Cafu Engine Web site*. [Online] 2011.
http://www.cafu.de/wiki/_media/textures:skytut_1s.gif.
8. **Geczy, George.** *2D Programming in a 3D World*. s.l. : UMB Techweb, 2001.
9. **Playrage.** Playrage Command & Conquer review. *Playrage web site*. [Online] 2011.
<http://www.playrage.com/reviews/yurisrevenge.shtml>.
10. **Westwood Studios.** Command & Conquer front page. *Command & Conquer web site*. [Online] 2011.
<http://www.commandandconquer.com/>.
11. **Valve Corporation.** Steam Counter-Strike. *Steam Web site*. [Online] 2011.
<http://store.steampowered.com/app/10/>.
12. **Fernandes, António Ramires.** Lighthouse3D - Billboarding tutorial. *Lighthouse 3D web site*. [Online] 2011. <http://www.lighthouse3d.com/opengl/billboarding/>.
13. **Autodesk, Inc.** Softimage Wiki - Billboard orientation. *Softimage Web Site*. [Online] 2011.
http://softimage.wiki.softimage.com/xsidocs/iceref_Billboard_Orientation.htm.
14. **Flipcode.** Flipcode - Billboard example. *Flipcode Web site*. [Online] 2011.
<http://www.flipcode.com/archives/billaln.jpg>.
15. **Blizzard Entertainment Inc.** World of Warcraft front page. *World of Warcraft Web site*. [Online] 2011. <http://eu.battle.net/wow/en/>.
16. **Schaufler, Gernot and Stürzlinger, Wolfgang.** *A Three Dimensional Image Cache for Virtual Reality*. Linz : s.n., 1995.

17. **Christiansen, Kenneth Rohde.** *The use of Imposters in Interactive 3D Graphics Systems.* Groningen : s.n.
18. **Aubel, Amaury, Boulic, Ronan and Thalmann, Daniel.** *Animated Impostors for Real-time Display of.* Lausanne : s.n.
19. **Harris, Mark J. and Lastra, Anselmo.** *Real-Time Cloud Rendering.* Chapel Hill : EUROGRAPHICS, 2001.
20. **100fps.** 100fps - frames of the human eye. *100fps web site.* [Online] 2011.
http://www.100fps.com/how_many_frames_can_humans_see.htm.
21. **Remolar, I., et al., et al.** *Real-Time Tree Rendering.* Castellón : Springer-Verlag, 2004.
22. **Umenhoffer, Tamás, Szirmay-Kalos, László and Szijártó, Gábor.** *Spherical Billboards and their Application to Rendering Explosions.* Budapest : s.n.
23. **Rousseau, Pierre, Jolivet, Vincent and Ghazanfarpour, Djamchid.** *Realistic Real-Time Rain Rendering.* s.l. : Computers & Graphics, 2006.
24. **Eric, Risser.** *True Imposters.* s.l. : University of Central Florida.
25. **Southimpact.** Southimpact front page. *Southimpact Web site.* [Online] 2011.
<http://www.southimpact.com/>.
26. **nVidia Corporation.** nVidia PhysX. *nVidia Web site.* [Online] 2011.
http://www.nvidia.com/object/physx_new.html.
27. **Electronic Arts Inc.** Electronic Arts Crysis 2 front page. *Electronic Arts Web site.* [Online] 2011.
<http://www.ea.com/crysis-2>.
28. **Stam, Jos.** *Stable Fluids.* s.l. : Siggraph, 1999.
29. **Autodesk Inc.** Autodesk Maya. *Autodesk Web site.* [Online] 2011. <http://usa.autodesk.com/maya/>.
30. **Stam, Jos, Fedkiw, Ronald and Jensen, Henrik Wann.** *Visual Simulation of Smoke.* 2001.
31. **Princeton University.** Princeton University - Viscosity. *Princeton University Web site.* [Online] 2011. http://www.princeton.edu/~gasdyn/Research/T-C_Research_Folder/Viscosity_def.html.
32. **Nguyen, Duc, Enright, Doug and Fedkiw, Ron.** *Simulation and Animation of Fire and Other Natural Phenomena in the Visual Effects Industry.* Stanford : Stanford University, 2003.
33. **IMDB.** IMDB - Terminator 3: Rise of The Machines. *IMDB web site.* [Online]
<http://www.imdb.com/title/tt0181852/>.
34. **Cohen, Jonathan M., Tariq, Sarah and Green, Simon.** *Interactive Fluid-Particle Simulation using Translating Eulerian Grids.* s.l. : Nvidia, 2010.
35. **Foster, Nick and Metaxas, Dimitris.** *Modeling the Motion of a Hot, Turbulent Gas.* Philadelphia : s.n., 1997.

36. **Multiple.** RealFlow Forum - Solvers & Methods used in the softwares available in the market. *RealFlow Forum web site*. [Online] 2011. http://www.realflowforum.com/view_topic.php?pid=40815.
37. **Wikipedia Foundation Inc.** Wikipedia - Fluid Dynamics. *Wikipedia Web site*. [Online] 2011. http://en.wikipedia.org/wiki/Fluid_dynamics.
38. —. Wikipedia - Claude-Louis Navier. *Wikipedia Web site*. [Online] 2011. http://en.wikipedia.org/wiki/Claude-Louis_Navier.
39. —. Wikipedia - Sir George Stokes. *Wikipedia Web site*. [Online] 2011. http://en.wikipedia.org/wiki/Sir_George_Stokes,_1st_Baronet.
40. **Bidson, Robert.** *Fluid Simulation for Computer Graphics*. Welleslay : A K Peters, Ltd., 2008. ISBN: 978-1-56881-326-4.
41. **Washington University.** Washington University - Department of Chemistry. *Washington University*. [Online] 2011. http://depts.washington.edu/chemcrs/bulkdisk/chem520A_aut05/notes_Week_05_Lecture_07.pdf.
42. **Fefferman, Charles L.** *Existence & Smoothness of The Navier-Stokes Equation*. Princeton : Princeton University, Department of Mathematics, 2000.
43. **NASA.** NASA Navier-Stokes equation. *NASA web site*. [Online] <http://www.grc.nasa.gov/WWW/k-12/airplane/nseqs.html>.
44. **Wikipedia Foundation Inc.** Wikipedia - Leonhard Euler. *Wikipedia Web site*. [Online] 2011. http://en.wikipedia.org/wiki/Leonhard_Euler.
45. —. Wikipedia - Euler Equations. *Wikipedia Web site*. [Online] 2011. [http://en.wikipedia.org/wiki/Euler_equations_\(fluid_dynamics\)](http://en.wikipedia.org/wiki/Euler_equations_(fluid_dynamics)).
46. —. Wikipedia - Lagrangian Equations. *Wikipedia Web site*. [Online] 2011. http://en.wikipedia.org/wiki/Lagrange_equations.
47. **The GAP Group.** GAP - Lagrange. *GAP Web site*. [Online] 2011. <http://www.gap-system.org/~history/Biographies/Lagrange.html>.
48. CGSociety - Why Fluid Dynamics? *CGSociety Forum Web site*. [Online] 2011. <http://forums.cgsociety.org/showthread.php?f=86&s=1acfdcb99fed634c0a01135d12cefe8&t=838205&page=1&pp=15>.
49. **Activision Publishing, Inc.** Call Of Duty front page. *Call of Duty Web site*. [Online] Activision, 2011. <http://www.callofduty.com/>.
50. John Locke: Stanford University. *Stanford University Web Site*. [Online] 2001. <http://plato.stanford.edu/entries/locke/>.
51. **British Council - BBC.** British Council - BBC - Subjective Test. *British Council - BBC Website*. [Online] 2011. <http://www.teachingenglish.org.uk/knowledge-database/subjective-test>.

52. The Free Dictionary - Objective Test. *The Free Dictionary*. [Online] 2011.
<http://www.thefreedictionary.com/objective+test>.
53. **smokinblackdmax**. *Oil fire*. 2006.
54. *Williston North Dakota Oil Fire*. NorthDakotaOilandGas, 2011.
55. *Fires of Kuwait*. IMAX, 1992.
56. **Autodesk**. *Autodesk Maya 2011 Help Index*. [Computer files provided with the Autodesk Maya 2011 package] s.l. : Autodesk, 2011.
57. **Adobe Systems Incorporated**. Adobe Premiere. *Adobe Web Site*. [Online] 2011.
<http://www.adobe.com/products/premiere.html>.
58. **Gnomon Workshop**. *Introduction to Maya Fluid Effects*. [Movie Tutorial] s.l. : Gnomon Workshop.
59. **Sargesso**. *amoke animation 4*. 2007.

17 Appendix

17.1 Appendix 1: Questionnaire

Age:____ Sex:____ Education:_____

Do you have any experience in creating computer effects, such as smoke or fire?

☐ Professional ☐ Experienced ☐ Intermediate ☐ Beginner ☐ No experience

‘The visual appeal should be compared to the two reference shots which were shown at the beginning of the test.’

*Please rate the visual appeal of the **first** effect:*

☐ Very bad ☐ Bad ☐ Neutral ☐ Good ☐ Very good

*Was there any particular element which made you give this exact rating for the **first** effect?:*

*Please rate the visual appeal of the **second** effect:*

☐ Very bad ☐ Bad ☐ Neutral ☐ Good ☐ Very good

*Was there any particular element which made you give this exact rating for the **second** effect?:*

*Please rate the visual appeal of the **third** effect:*

☐ Very bad ☐ Bad ☐ Neutral ☐ Good ☐ Very good

*Was there any particular element which made you give this exact rating for the **third** effect?:*

Did you notice any differences between the **first** and the **second** effect?:

☐ Yes ☐ No

If **yes**, please describe what the differences between the effects were, as detailed as possible:

Did you notice any differences between the **first** and the **third** effect?:

☐ Yes ☐ No

If **yes**, please describe what the differences between the effects were, as detailed as possible:

Filled out by test controller

Effect:

☐ Fan ☐ Cone