Tiago Luís Oliveira Leite

Web-based Virtual Lab for Computer Generated Holography

Outubro de 2015

Universidade do Minho
Escola de Engenharia
Departamento de Informática

Tiago Luís Oliveira Leite

**Web-based Virtual Lab for
Computer Generated Holography**

Dissertação de Mestrado
Mestrado em Engenharia Informática

Trabalho realizado sob orientação de

**Professor Luís Paulo Santos
Waldir Pimenta**

Outubro de 2015

Acknowledgements

I want to thank both of my supervisors, Prof. Luís Paulo Santos and Waldir Pimenta. Their knowledge, dedication, experience and patience helped me through the development of the thesis.

I also want to thank my parents and my family. Without their help and support I wouldn't be able to have the opportunity to do this project.

To my friends at Infobiz @ Uminho, that put up with me every day and helped me when I needed. Thank you.

And finally, I want to thank everybody that believed in me and helped me reach this moment of my life.

Abstract

Optical holograms are captured using specialized equipment. Current computing capabilities allow for the simulation of this process with a computer, creating Computer Generated Holograms. Since CGH first appeared, several researchers have been trying to find ways to accelerate this process, proposing several alternative approaches. To create a CGH it is necessary to use the wave model of light transportation so that is possible to calculate interference between waves. This is a computationally very demanding process, requiring the design of more effective ways to generate CGH without sacrificing quality. Understanding the wave model of light transportation and how holograms are created are no easy tasks, since quite a bit of knowledge in physics is required. One of the goals of this project is to help people understand these phenomena by means of a web-based interactive virtual laboratory that makes use of a 3D environment to illustrate waves emitted from various point sources and their interference. This system will be parametrizable through a graphical interface, allowing the user to visualize and investigate the effect of various parameters.

Resumo

Hologramas ópticos são criados utilizando equipamento especializado. As capacidades de computação atuais permitem a simulação deste processo usando computadores, criando Hologramas Gerados por Computador. Desde a primeira aparição dos HGCs, vários investigadores têm tentado encontrar maneiras de acelerar este processo, propondo diversas abordagens alternativas. Para criar um HGC é necessário utilizar o modelo de ondas do transporte de luz, para que seja possível calcular a interferência entre as ondas. Este processo é muito exigente computacionalmente, sendo necessário a criação de formas mais eficazes para gerar HGCs sem sacrificar a qualidade. Entender o modelo de ondas do transporte de luz e como hologramas são criados não são tarefas fáceis, uma vez que algum conhecimento em física é necessário. Um dos objetivos deste projeto é ajudar as pessoas a entender estes fenómenos usando um laboratório virtual interativo na web com um ambiente 3D para ilustrar ondas emitidas a partir de vários pontos de luz e sua interferência. Este projecto será parametrizável por meio de uma interface gráfica que permitirá visualizar e estudar o efeito de diferentes parâmetros.

Contents

Acronyms

BSD Beam Splitter Direction. 36, 47

BSP Beam Splitter Position. 37

BSR Beam Splitter Rotation. 37

CcLD1 Concave Lens Direction 1. 36

CcLD2 Concave Lens Direction 2. 36

CcLP1 Concave Lens Position 1. 37, 44

CcLP2 Concave Lens Position 2. 37

CcLR1 Concave Lens Rotation 1. 37, 44

CcLR2 Concave Lens Rotation 2. 37

CGH Computer Generated Hologram. iv, 2, 5, 21–23, 68

CvLD Convex Lens Direction. 36

CvLP Convex Lens Position. 37

CvLR Convex Lens Rotation. 37

FPS Frames per Second. 65, 66

GPU Graphics Precessing Unit. 32, 51, 66

LSD Laser Source Direction. 36

LSP Laser Source Position. 37

LSR Laser Source Rotation. 37

MD Mirror Direction. 35, 47

MP Mirror Position. 37

MR Mirror Rotation. 37, 47

OD Object Direction. 36

OP Object Position. 37

OR Object Rotation. 37

PP Plate Position. 35

PR Plate Rotation. 35

RWA Reference Wave Angle. 35, 47, 50, 52–54

List of Figures

List of Tables

Listings

Chapter 1

Introduction

1.1 Context

Vision is a very important sense for humans, and how the most important moments are saved and visualized is something valued.

Since the invention of photography, its evolution has been tremendous. The first photos were black and white and very low resolution. Now its possible to take photos with billions of different colors at very high resolutions. Video was another technique invented in order to visually record the environment, but instead of recording just a moment in time like photography, it records an entire time interval.

Although photography and video are the most used ways of recording the environment, these techniques only produce two-dimensional images. This means that although the photo or video belongs to a 3D object, it does not offer all the depth cues necessary to fully trick the human brain into thinking that what it is seeing is a true 3D object and not a flat image.

Lately, the search for technologies capable of presenting content in three dimensions has been increasing. For example, the movie industry has been trying, in the last few years, to elicit true 3D perception in movies. Also, there have been several researchers in the field of medicine trying to create ways of generating 3D images of the human organs. Although some of the solutions found to solve these problems

are capable of satisfying the requirements, they are still far from achieving the 3D perception that holography offers.

Holography is a technology capable of creating images that reproduce a three-dimensional scene faithfully without the need of an ocular device and has the potential to trick the human brain into thinking that a hologram is a real physical scene. Holography also allows several users observe the same image from different perspectives. If a user is looking at a hologram from a given perspective and then he changes the viewing angle, the perspective of the image will also change, as if he was looking at a real object. Holography may be the best way to create and display three-dimensional images, but to create holographic videos with acceptable frame rate and resolution, much research has yet to be made. Nevertheless, holographic images are already in use, for example, to prevent counterfeit money.

There are two ways of creating holograms. Optical holograms are those captured using specific equipment, while Computer Generated Holograms are created by simulating the optical process with a computer. To this day, researchers have found several ways to create CGHs, but there may still be room for improvements, either creating new methods or improving the existing ones.

1.2 Motivation and objectives

The recording and reconstruction of holograms is based on the physical phenomena of interference and diffraction, respectively, so unlike ray tracing, which uses the geometric model of light transportation to create highly realistic images, computer generated holography requires the wave model of light transportation in order to reproduce those physical phenomena.

The wave model of light transportation, the interference and diffraction phenomena, and how these play on the recording and reconstruction of holograms, are not easy topics to understand. At least, some research into physics and holography needs to be made, and in the end some people may still be confused. This dissertation entails designing, developing and assessing a web-based virtual laboratory, which

allows users to examine how the wave interference phenomenon works, how holograms are recorded from it, and how the diffraction phenomenon contributes to its reconstruction.

Interactivity is very important in this laboratory. The user have control of several parameters that can change properties that affect the quality of the final hologram. Several objects are available for choosing, with different complexity levels, so the user can also assess hologram recording time as a function of scene complexity. Furthermore, the laboratory is based on a three-dimensional scene, so that the illustration of the several components and phenomena involved –which results in the hologram–, and the changeable parameters that affect this processes, can be more evident for the user.

As this laboratory has the main goal of being a didactic tool for those who know nothing or very little about holography, another goal for this project is to assess whether the laboratory actually helps people grasp the concepts involved in the recording and reconstruction of holograms. In order to test this, the tool was evaluated with the help of students who can potentially use this tool in the future. This evaluation will provide feedback that will be used to improve the tool at the user experience level.

Is also expected that the laboratory can be used by researchers in the field of computer generated holography to help them on their research. The goal with this is to allow researchers to use the tool to quickly draft their experiments, and save some time by easily changing the parameters to see an approximation of the result for these experiments. This will make the conclusions about those parameters values to be reach more quickly.

This project is also expected to draw some conclusions about the use of web technologies in the creation of computer generated holograms. More precisely, explore the performance limits during the simulation of the physical phenomena involved on the creation of a hologram and explore the performance limits of rendering all the geometry needed to illustrate the process. How that will limit the use of the laboratory both by normal users, who just want to know more about the subject,

and more experienced users is something that will be discussed as well.

1.3   Structure of the dissertation

This dissertation is organized as follows:

Chapter 2 - Theoretical foundations - addresses several topics that are required to understand holography and the phenomena involved. An introduction to computer generated holography is made and one of the main algorithms is presented.

Chapter 3 - Planning of the virtual lab - the plans for the laboratory are shown here. Mockups of the laboratory and the technologies that will be used are presented on this chapter.

Chapter 4 - Implementation of the virtual lab - the implementation of the laboratory is presented here. How the scene is constructed, how the interference patterns are generated and the creation of the user interface are addressed on this chapter.

Chapter 5 - User experience on the virtual lab - this chapter serves as a guide for the user. Its objective is to show the users their possibilities with the laboratory.

Chapter 6 - Virtual lab evaluation - an evaluation on the performance of the laboratory and its limitations is done here. This chapter also addresses how external feedback influenced the outcome of the laboratory.

Chapter 7 - Conclusion - concludes this dissertation with comments on the work done, results, and future work.

Chapter 2

Theoretical foundations

This chapter addresses several important topics with the objective of helping understand holography and the phenomena behind it. First, the physical phenomena which make holography possible are explained. Second, an introduction to holography is made by explaining in a brief way how holograms are generated using the optical method. Next, computer generated holography is addressed by explaining the differences to optical holography and the way it works. The different families of methods to generate CGH are referenced. Only the physically-based approach is explained in more detail since it is the one adopted by the project.

2.1 Wave propagation, interference and diffraction

During the history of mankind, several theories about what light is and its behavior were put forward. However, only since the 17th century theories with solid scientific bases started to arise. The first of these was proposed by Isaac Newton, he claimed that light was composed by particles [Klu]. He affirmed that light was emitted from a light source and was composed by particles that eventually ended up hitting the human eye. In the same century, the physicist and mathematician Christiaan Huygens affirmed that Newton was wrong and that the only way light can travel

at very high speed would be if it were composed by waves [Klu]. Thus, light would have a wave behavior and not a corpuscular one. This theory was not very well accepted until the 19th century.

In the beginning on the 19th century, Thomas Young and Augustin Fresnel developed the ideas of Huygens. Young did an experiment to prove that the wave theory was correct [You02]. The experiment consisted in proving that light suffers from interference, a phenomenon distinctive of all waves. Although the wave theory of light was accepted after Young's experiment, in the end of the 19th century the theory was questioned for failing to explain all phenomena of light. Albert Einstein tried to respond these questions affirming that light displays both wave and corpuscular behavior [Ein09]. This way, light is composed by both corpuscles (called photons) and waves. This theory responds to the problems posed at the time and it is the most accepted theory nowadays.

Today it is accepted that light is a form of energy that leaves some kind of source and reaches the human eye as electromagnetic waves. The human eye is able to perceive electromagnetic waves within a particular interval of frequencies. This interval is called visible light. Other intervals have been named, such as the infrared or the ultraviolet, as can be seen in the figure 2.1.
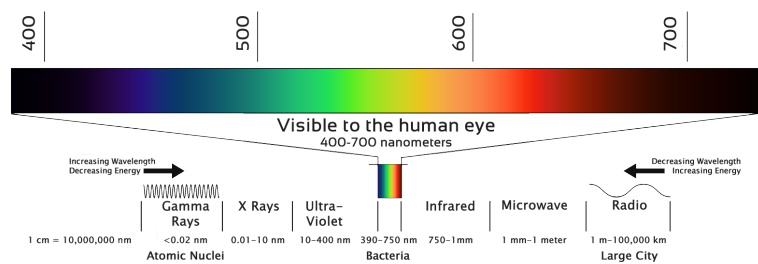
Figure 2.1: *Electromagnetic spectrum. Image from SST Software website*

2.1.1 Wave propagation

Electromagnetic waves corresponds to spatially propagating oscillations. This behavior allows their representation as a sinusoid like the one on figure 2.2.
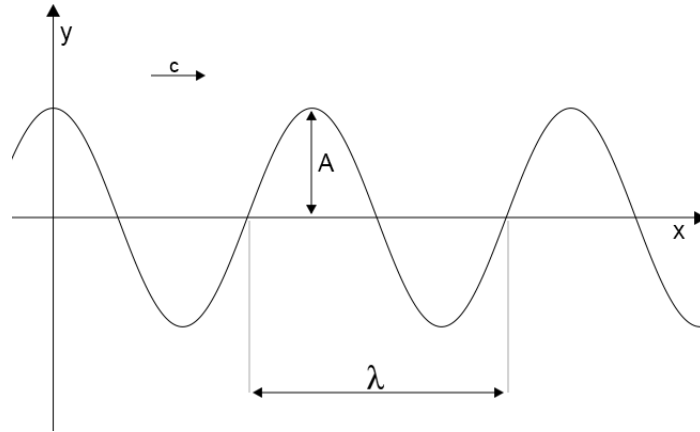
Figure 2.2: *Electromagnetic wave representation*

These waves, like any other waves, have properties like amplitude (A), which is the difference value between the rest position and the crest position of the wave. The distance between one point of a wave oscillation and the same point of the next oscillation is called wavelength (λ). Electromagnetic waves travel at a constant speed, c. In vacuum this speed is approximately $3 \times 10^8 m/s$. The wave period is the time it takes for a wave to travel the wavelength distance (λ) at constant speed c. So, the wave period can be calculated by:

$$\tau = \frac{\lambda}{c} \tag{2.1}$$

The wave frequency is the number of oscillations that it makes per second, so if the period is the time it takes to make one oscillation, the frequency can be given by:

$$f = \frac{1}{\tau} \tag{2.2}$$

With that said, considering a given instant t in time and initial phase φ, which is the number of radians that the wave is displaced initially, a electromagnetic wave can be described mathematically with the following equation:

$$y(t) = A\cos(\omega t + \varphi) \tag{2.3}$$

Where A is the amplitude of the wave. If $A = 1$ and $y(0) = 1$, then φ can have the value 2π, for example. ω is the angular frequency, which is given by:

$$\omega = 2\pi f \tag{2.4}$$

The angular frequency represents how many radians the wave oscillated in one second. So, for example, if a wave has frequency $f = 3Hz$ (completes three oscillations in one second), it means that the wave oscillated 6π radians in one second.

Considering that the wave described by 2.3 is moving in the x-direction, if a point of the wave, at a distance x of the beginning the wave is considered, the phase shift corresponding to that point is proportional to the time it takes the wave to travel the distance x with constant speed c, which is given by $t_0 = x/c$. So, taking this into account its possible to change to the equation 2.3, obtaining:

$$y(t) = A\cos(\omega(t - t_0) + \varphi) \tag{2.5}$$

So, the wave in figure 2.2 can represent a snapshot of a electromagnetic wave propagating in the x-direction and can be mathematically described by the equation 2.5. If this equation is developed, the following relation can be found:

$$
\begin{aligned}
y(t) &= A\cos(\omega(t - t_0) + \varphi)\\
y(t, x) &= A\cos(\omega(t - \frac{x}{c}) + \varphi)\\
&= A\cos(\omega(t - \frac{x}{\frac{\lambda}{\tau}}) + \varphi)\\
&= A\cos(\omega(t - \frac{x\tau}{\lambda}) + \varphi)\\
&= A\cos(\omega(t - \frac{x\frac{1}{f}}{\lambda}) + \varphi)\\
&= A\cos(\omega(t - \frac{\frac{x}{f}}{\lambda}) + \varphi) \tag{2.6}\\
&= A\cos(\omega(t - \frac{x}{f\lambda}) + \varphi)\\
&= A\cos(2\pi f(t - \frac{x}{f\lambda}) + \varphi)\\
&= A\cos(2\pi ft - \frac{2\pi fx}{f\lambda} + \varphi)\\
&= A\cos(\omega t - \frac{2\pi x}{\lambda} + \varphi)\\
&= A\cos(\omega t - kx + \varphi)
\end{aligned}
$$

In the equation 2.6 there is now a time component and space component. For each moment t is possible to calculate the amplitude of the wave for each value of x. On

8

this equation, k is the wave number and has the expression: $k = 2\pi/\lambda$. The wave number represents the number of radians per unit distance the wave travels and its purpose is to convert distances into angles (radians). So, assuming that $\lambda = \pi$ ($k = 2$) and that x is at a distance of 2π from the beginning of the wave ($x = 2\pi$), this distance corresponds to 4π radians in the wave.

Using the Euler law (Eq. 2.7) its possible to avoid the use of trigonometric functions and transform them into exponential functions using complex numbers [AE07, Chapter 2]. The reason behind this transformation is to simplify the calculations used in the interference process. With exponential functions a specific set of mathematical rules can be applied to these functions. This way is more easy to achieve the necessary formulas to understand that process.

$$e^{ix} = \cos x + i \sin x \tag{2.7}$$

From equation 2.7, Euler also deducted an interpretation of the cosine and sine functions as weighted sums of the exponential function:

$$\cos(x) = \frac{1}{2}(e^{ix} + e^{-ix}) \tag{2.8}$$

$$\sin(x) = \frac{1}{2i}(e^{ix} - e^{-ix}) \tag{2.9}$$

Using the relation obtained with the equation 2.8, it is possible to transform the previous equation 2.6 into:

$$y(t, x) = A\frac{1}{2}(e^{i(\omega t - kx + \varphi)}) + A\frac{1}{2}(e^{-i(\omega t - kx + \varphi)}) \tag{2.10}$$

The equation 2.10 describes the sum between a complex number and his conjugate. So, representing these numbers in a coordinate system will result in the two imaginary parts cancel each other and the real parts adding up to result in the cosine wave. Therefore, its possible to define the wave just taking one of the parts of the equation and multiplying it by 2.

$$y(t, x) = Ae^{-i(wt - kx + \varphi)} \tag{2.11}$$

2.1.2 Interference

Interference is a physical phenomenon resulting from the superposition of two or more waves, creating a new wave with a new amplitude for each of its points, that can be greater or lower than the amplitude of the original waves. A light point source produces a circular wave when two dimensions are being considered (spherical wave for three dimensions). This means that when the light of two or more light points overlaps, the interference pattern will map the way the phase difference between the two waves varies in space. This pattern will be dependent of the wavelength of the waves and distance between the several points. Usually, interference refers to the interaction of coherent waves (a requirement when using techniques like holography), that is, all the waves have the same wavelength. However, in order to superposition of waves occur the coherence the waves is not a requirement. When two or more waves overlap the principle of superposition always happens.

The interference phenomenon uses the superposition principle to obtain the amplitude of the resultant wave. The amplitude of a point of the wave can be positive or negative depending if a crest or a trough is being considered. So, there are two types of interference, constructive and destructive.

Constructive interference happens when the waves have their crests and troughs aligned, that is, the phase difference between them is a multiple of 2π, which corresponds to a full cycle. The resulting amplitude is the sum of the amplitudes of the waves involved in the interference, so the magnitude of the resulting amplitude will be greater than the magnitude of the amplitude of each wave involved.

Destructive interference happens when a crest of a wave is aligned with a trough of another, that is, the phase difference between the waves is an odd multiple of π, which corresponds to a half cycle. The resulting amplitude is also the sum of the amplitudes of the waves involved, but in this case the amplitudes haves different signs, so the magnitude of the resulting amplitude may be lower than the magnitude of the amplitude of each wave involved, or even zero.

The two cases presented above are the extremes. The waves can have a phase

difference between $N2\pi + \pi, N \in \mathbb{N}$ (maximum destruction or minimum amplitude value) and $N2\pi, N \in \mathbb{N}$ (maximum construction or maximum amplitude value). There are also middle cases where the value of the resulting amplitude lies between the minimum and maximum values.

Figure 2.3 illustrates the interference phenomenon in one dimension, while figure 2.4 illustrates the same phenomenon in two-dimensions.
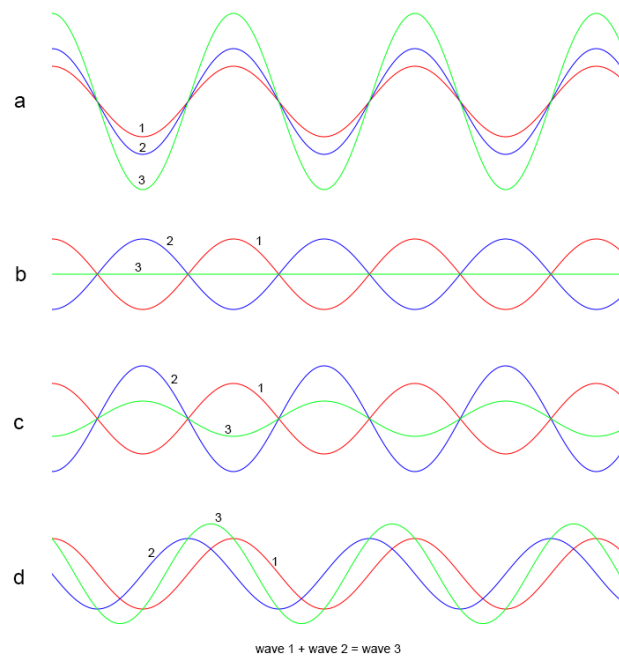
Figure 2.3: *Wave interference. a) Maximum constructive interference b,c) Maximum destructive interference d) Intermediate case*
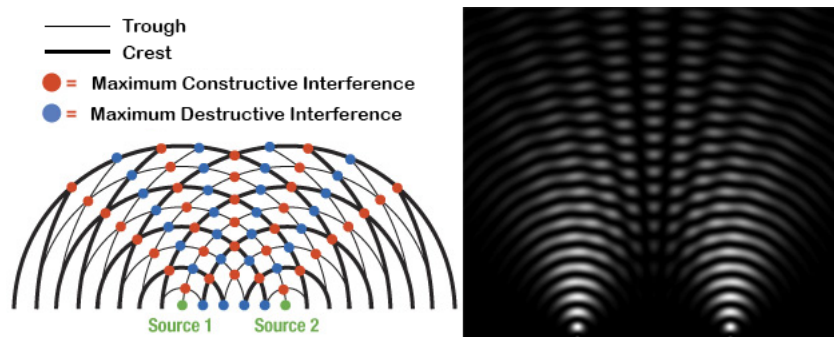
Figure 2.4: *Wave interference in two-dimensions. Images from PhysicsForums website and William J. Prest article on HubPages website*

The concepts involved in interference can be described mathematically [AE07, Chap-

ter 2] [Har02, Appendix A] [dSS12, Chapter 2]. To do that, two waves O and R must be considered.

$$O(t, x_o) = A_o e^{-i\Phi(t, x_o)} \tag{2.12}$$

$$R(t, x_r) = A_r e^{-i\Psi(t, x_r)} \tag{2.13}$$

Where, $\Phi(t, x_o) = wt - kx_o + \varphi$ and $\Psi(t, x_r) = wt - kx_r + \psi$. The interference can be observed through the intensity of an interference pattern. The intensity of the resultant wave in a particular point is the square of the sum of the two waves (O and R) in the same point [AE07, Chapter 2]:

$$\begin{aligned} I &= |O + R|^2 \\ &= OO^* + OR^* + RO^* + RR^* \end{aligned} \tag{2.14}$$

The star($*$) represents the complex conjugate. The four summands on equation 2.14 can be calculated as:

$$\begin{aligned} OO^* &= A_o^2 \\ RR^* &= A_r^2 \\ OR^* &= A_o A_r e^{-i\Phi} e^{i\Psi} \\ RO^* &= A_r A_o e^{-i\Psi} e^{i\Phi} \end{aligned} \tag{2.15}$$

Now, using the Euler law in the equation 2.8, it is possible to obtain:

$$I = A_o^2 + A_r^2 + 2A_o A_r (\cos(\Phi - \Psi)) \tag{2.16}$$

From the equation 2.16 it is possible to deduce that the resulting wave will have its maximum intensity when $\cos(\Phi - \Psi) = 1$ and its minimum intensity when $\cos(\Phi - \Psi) = -1$. So, the respective maximum and minimum intensities are described by the equations:

$$I_{max} = A_o^2 + A_r^2 + 2A_o A_r \tag{2.17}$$

$$I_{min} = A_o^2 + A_r^2 - 2A_o A_r \tag{2.18}$$

So, to discover when the intensity is maximum the following calculation can by made:

$$
\begin{aligned}
\cos(\Phi - \Psi) &= 1 \\
\cos((wt - kx_o + \varphi) - (wt - kx_r + \psi)) &= 1 \\
\cos(wt - kx_o + \varphi - wt + kx_r - \psi) &= 1 \\
\cos(-kx_o + \varphi + kx_r - \psi) &= 1 \\
\cos(k(x_r - x_o) + \varphi - \psi) &= 1 \\
k(x_r - x_o) + \varphi - \psi &= N2\pi, N \in \mathbb{N} \\
\frac{2\pi}{\lambda}(x_r - x_o) + \varphi - \psi &= N2\pi, N \in \mathbb{N} \\
\frac{2\pi}{\lambda}(x_r - x_o) &= N2\pi - \varphi + \psi, N \in \mathbb{N} \\
(x_r - x_o) &= \frac{\lambda(N2\pi - \varphi + \psi)}{2\pi}, N \in \mathbb{N} \\
(x_r - x_o) &= \lambda(N - \frac{\varphi}{2\pi} + \frac{\psi}{2\pi}), N \in \mathbb{N}
\end{aligned}
\tag{2.19}
$$

Assuming that the waves involved in the interference start with same initial phase, it is possible to assume that $-\frac{\varphi}{2\pi} + \frac{\psi}{2\pi} = 0$. So, the equation 2.19 can be simplified:

$$
(x_r - x_o) = \lambda N, N \in \mathbb{N}
\tag{2.20}
$$

Using the same strategy used to calculate when the intensity is maximum, it is possible to obtain when the intensity is minimum:

$$
(x_r - x_o) = \lambda(N + \frac{1}{2} - \frac{\varphi}{2\pi} + \frac{\psi}{2\pi}), N \in \mathbb{N}
\tag{2.21}
$$

And again, assuming that the waves involved in the interference start with same initial phase, then:

$$
(x_r - x_o) = \lambda(N + \frac{1}{2}), N \in \mathbb{N}
\tag{2.22}
$$

From the equations 2.20 and 2.22 it is possible to deduce that the intensity of the resulting wave will have maximum intensity when the difference between the distances x_r and x_o is $\lambda N, N \in \mathbb{N}$ and minimum intensity when the difference is $\lambda(N + \frac{1}{2}), N \in \mathbb{N}$, as represented if the figure 2.5.
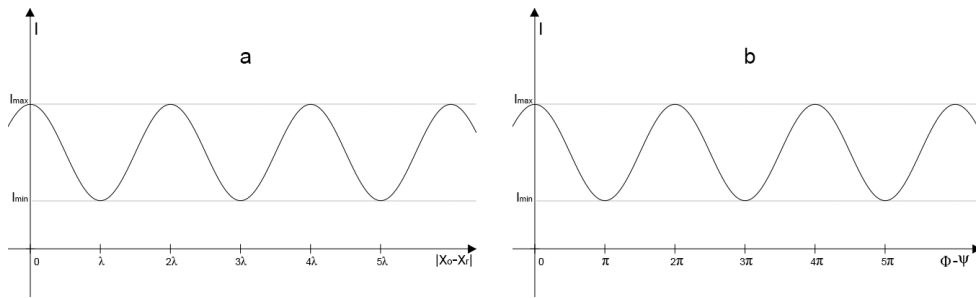
Figure 2.5: *Maximum and minimum intensities. a) depending on the difference between the distances x_r and x_o from equations 2.20 and 2.22 b) depending on difference between the phases Ψ and Φ from equation 2.14*

2.1.3 Diffraction

Interference and diffraction are actually similar phenomena. It is common to use the term "interference" in a more general sense, where the interfering waves may not have the same source, or where a beam was split and is interfering with itself after being recombined. Diffraction is more commonly used when talking about the way a single wave evolves as it propagates.

The diffraction phenomenon is easily visible when a plane with two slits is put in front of a coherent monochromatic light. The slits must have the approximated size of the wavelength of the waves emitted by the light source. The only way to the light go through the plane is through the slits. And those slits will start to behave like two new light points, as illustrated on figure 2.6.
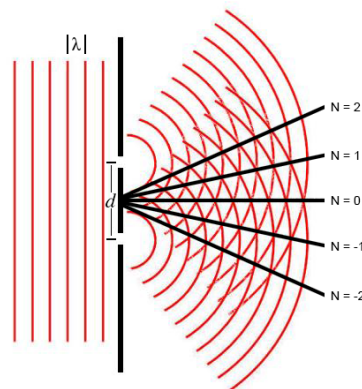
Figure 2.6: *Double-slit diffraction example*

It is possible to know mathematically the deviation that a wave will suffer when going through a slit [BBJ08, Chapter 5] [dSS12, Chapter 2]. The equation 2.20 shows when a wave has maximum intensity. Assuming a point after the the plane with the slits, one of the waves may have to travel a bigger distance to arrive at that point. The figure 2.7 illustrates one of these cases. So, the difference of distances can be calculated as:

$$(x_r - x_o) = d \sin \theta_{out} \tag{2.23}$$
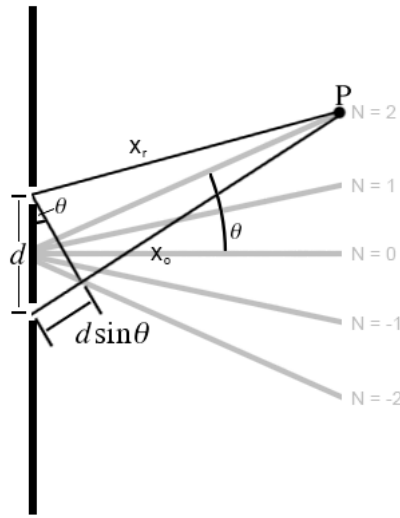
Figure 2.7: *Calculation of the difference between the distances x_r and x_o to an arbitrary point P*

That way, it is possible to know the angle of the diffracted wave to a given value of N. This N, can be called *order*. For each order, the wave is diffracted by an angle θ and an angle $-\theta$. The "first order" waves being those closest to the straight-through or "zero order" wave. The next set, if those waves exist at all, are the "second order" waves, and so forth [BBJ08, Chapter 5].

$$d \sin \theta_{out,N} = \lambda N, N \in \mathbb{N}$$
$$\sin \theta_{out,N} = \frac{\lambda N}{d}, N \in \mathbb{N} \tag{2.24}$$

The equation 2.24 is known as "On-axis grating equation", because the waves arrive at the slits with an angle of 90°. However, the wave may arrive at the slits in an angle that is not perpendicular. This situations are described by the "Off-axis

grating equation", where the angle of entrance in the split (θ_{in}) is considered [BBJ08, Chapter 5].

$$\sin\theta_{out,N} - \sin\theta_{in} = \frac{\lambda N}{d}, N \in \mathbb{N}$$
$$\sin\theta_{out,N} = \frac{\lambda N}{d} + \sin\theta_{in}, N \in \mathbb{N} \tag{2.25}$$

This equations (2.24 e 2.25) can be also used in cases where there are more than two slits. With more slits there is more interference between waves to consider, but the diffraction angles will be the same.

2.2 Optical holography

Optical Holography (or Classic Holography) is a technique used to create three-dimensional images that store waves reflected by one or more objects, with all their properties, on a photosensitive plate. The process of recording the hologram uses the interference phenomenon, while the process of reconstruction uses the diffraction phenomenon discussed in the previous section.

2.2.1 Recording of a hologram

In order to record a hologram, the wave emitted by the object (object wave) needs to interfere with a reference wave on the photosensitive plate. To do that, the two waves must be coherent. This means that both waves must have the same wavelength before they hit the plate.

To be sure that both waves have the same properties, a beam splitter is used. A beam splitter allows a beam to the divided in two with the same properties but with different propagation directions. This way, both waves have the same wavelength.

Next, one of the beams will illuminate the object and the wave created by interaction with it (object wave) will spread out outwards, part of it reaching the photosensitive plate. The other wave (reference wave) will be directed to the plate without any changes. On the plate, interference between these two waves will occur and the

intensity of the resulting wave will be recorded on the plate. Where constructive interference occurs, the plate will become opaque (the interference pattern gets recorded), and where destructive interference occurs, the plate remains transparent because the intensity of the resulting wave on those areas is zero, due to the interfered waves having the same amplitude. Figure 2.8 illustrates this process.
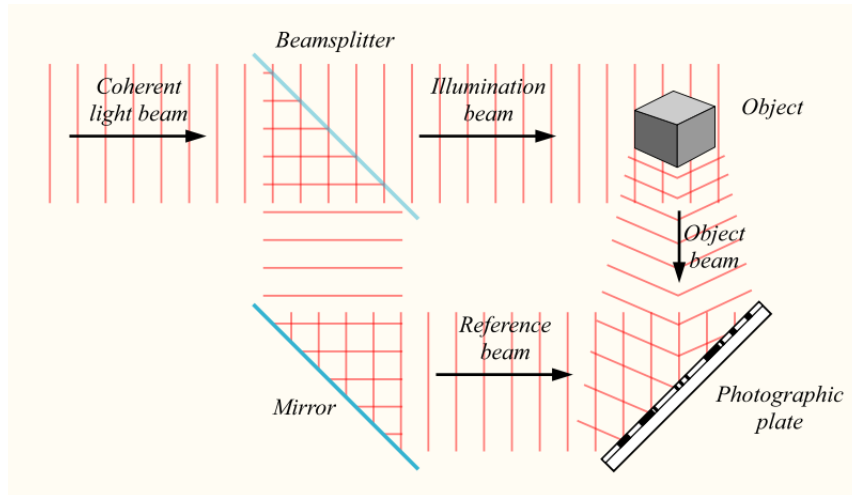
Figure 2.8: *Generic hologram recording. Image from the "Holography" Wikipedia article.*

On the photosensitive plate, fringes will appear resulting from the alternation of constructive interference areas and destructive interference areas. The proximity of these fringes depends on the angle of interference and the phase of the waves. The information of the object wave is completely contained in the resulting interference pattern. During the recording of holograms, the object and reference waves can be spatially superposed in different ways by changing the position of the reference light source or the object. The photosensitive plate can be also placed in different positions within of the field of interference. Those changes will result in several different holographic methods [AE07, Chapter 3] [BBJ08], like the in-line holograms or Gabor holograms, the off-axis holograms or Leith-Upatnieks holograms, the Fourier holograms and the reflection holograms, among others.

In order to demonstrate the mathematical calculations behind this process in a general way, lets assume two waves O and R, the same way as in the previous

section, where O represents the object wave and R the reference wave:

$$O(x, y) = A_o e^{-i\Phi(x,y)} \tag{2.26}$$

$$R(x, y) = A_r e^{-i\Psi(x,y)} \tag{2.27}$$

However, the reference wave can be simplified, since it can be assumed that it is a plane wave [AE07, Chapter 2] [Har02, Chapter 1]. The reason behind this choice lies on the fact that simplifies the calculations and helps on the reconstruction of the hologram, since it does not become dependent on the distance of the reference wave source to the hologram. So, the phase of the reference wave depends on the angle of incidence on the photosensitive plate, θ, and can be calculated by $\Psi = -2\pi x\sigma$, where $\sigma = \sin\theta/\lambda$. The reference wave is then represented by:

$$R(x, y) = A_r e^{i2\pi x\sigma} \tag{2.28}$$

It was already seen in the previous section that the intensity of the interference between two waves is given by equation 2.16:

$$I(x, y) = A_o^2 + A_r^2 + 2A_o A_r (\cos(\Phi - \Psi))$$

Optical holograms, like chemically developed photos, need exposure time to be recorded. The fraction of incident light that passes through the photosensitive plate is called transmission and is given by [AE07, Chapter 2]:

$$t = t_0 + \beta v I = t_0 + \beta E \tag{2.29}$$

The transmission decreases proportionally to the exposure intensity I and exposure time v because of the parameter β, which is negative and is related to the sensitivity of the photosensitive plate. t_0 is the time without exposure. $E = vI$ describes the energy density of the light, also known as exposure. So, it is possible to develop the equation 2.29 obtaining:

$$
\begin{aligned}
t(x, y) &= t_0 + \beta E \\
&= t_0 + \beta(vI) \\
&= t_0 + \beta v(A_o^2 + A_r^2 + A_o A_r e^{-i\Phi} e^{i\Psi} + A_o A_r e^{i\Phi} e^{-i\Psi}) \\
&= t_0 + \beta v A_o^2 + \beta v A_r^2 + \beta v A_o A_r e^{-i\Phi} e^{i\Psi} + \beta v A_o A_r e^{i\Phi} e^{-i\Psi})
\end{aligned} \tag{2.30}
$$

Simplifying the phase of the reference wave like in equation 2.28 results in:

$$t(x,y) = t_0 + \beta\upsilon A_o^2 + \beta\upsilon A_r^2 + \beta\upsilon A_o A_r e^{-i\Phi} e^{-i2\pi x\sigma} + \beta\upsilon A_o A_r e^{i\Phi} e^{i2\pi x\sigma} \qquad (2.31)$$

Equation 2.31 shows that both amplitude and phase of the object wave are recorded on the photosensitive plate. The amplitude will determine the opacity of the hologram on the plate and the phase will determine the distance between fringes.

2.2.2 Reconstruction of a hologram

The process of reconstruction of a hologram consists in illuminating a photosensitive plate where a interference pattern has been recorded with the same reference wave used in the recording process. This wave, when colliding with the interference pattern will suffer a deviation, that is, will undergo diffraction effects [dSS12, Chapter 2] [FS86, Chapter 12].

Each transparent point of the interference pattern, when illuminated, will create a new light point. The waves emitted by these points will eventually interfere with each other. This interference will produce zones where the light is more intense (constructive interference) and zones with no light (destructive interference). The zones where the light is more intense are called "diffraction zones". The figure 2.9 illustrates this process.

On the diffraction of first order formed in this process (one wave of order 1 and a wave of order -1) is where the three-dimensional image of the object used on the recording process will appear. The image formed by the order 1 of diffraction will present a orthoscopic behaviour, that is, it will produce correct sense of depth and it will appear on the same spot where the physical object was when the hologram was recorded. This image also has the name "virtual object" because the light does not converge to the front of the hologram in order to create a real image. It is the same as looking at a window and seeing an object behind that window. The image formed by the order -1 of diffraction will present a pseudoscopic behaviour, that is, the observer will experience an inverted sense of depth. This image has the
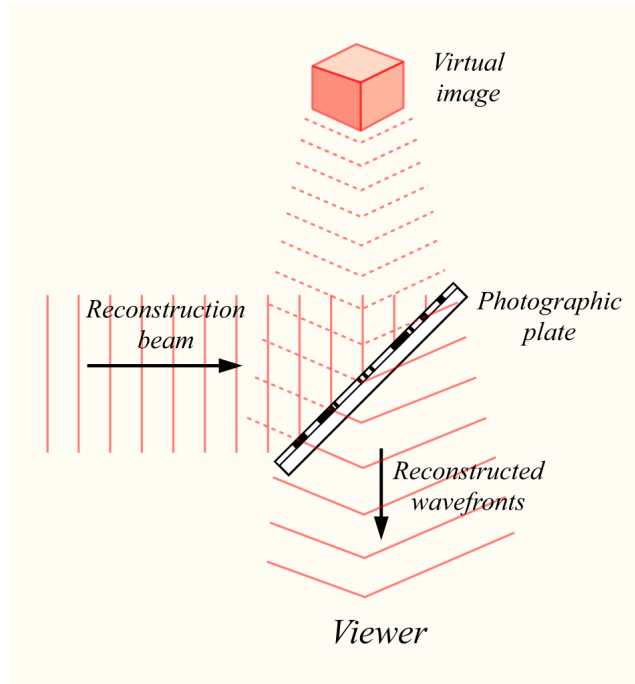
Figure 2.9: *Generic hologram reconstruction. Image from the holography Wikipedia web page*

name "real image" and appears in the front of the hologram, which makes it more difficult to see. It is the same as looking at a window and see an object before that window [Har02, Chapter 2] [dSS12, Chapter 2].

A unique feature of holography is that even if the hologram has been shattered in half, it is still possible to reconstruct the image contained in the hologram. However the image will present reduced quality.

Mathematically, the process of reconstruction can be described as the product of the reference wave R from equation 2.27 by the hologram $t(x, y)$ from equation 2.31. The hologram $t(x, y)$ will act as a filter for the reference wave. The resultant wave is described as $u(x, y)$ [AE07, Chapter 2] [Har02, Chapter 2] [dSS12, Chapter 2].

$$u(x, y) = R \times t(x, y) \tag{2.32}$$

Using the equations 2.27 and 2.31 it is possible to obtain a expression that describes

the effect of a hologram on a light wave during the reconstruction process.

$$
\begin{aligned}
u(x,y) = {} & (t0 + \beta\upsilon A_r^2)R \\
& + \beta\upsilon A_o^2 R \\
& + \beta\upsilon A_r^2 A_o e^{-i\Phi} \\
& + \beta\upsilon A_r^2 A_o e^{i\Phi} e^{i4\pi x\sigma}
\end{aligned}
\tag{2.33}
$$

The first two terms of the equation 2.33 correspond to the zeroth diffraction order. The first term refers to a intensity reduction of the reference wave by the factor $(t0 + \beta\upsilon A_r^2)$ during the reconstruction. The second term is responsible for the appearance of a halo around the reconstructed wave due to the fact that this term contains spatial variation. The small diffraction angles cause this effect. The third term corresponds to the diffraction order 1. This is the term responsible for the reconstruction of the object wave, which corresponds to the image recorded on the hologram. The virtual image created will appear at the position of the original object. The fourth term corresponds to the conjugate complex of the object wave and represents the diffraction order -1. This term will create a conjugated real image and will have a pseudoscopic behaviour [AE07, Chapter 2] [Har02, Chapter 2].

2.3 Computer Generated Holography

The development of Computer Generated Hologram (CGH) was a great advance in the field of holography. When comparing computer-generated holography with optical holography, several advantages in the use of CGH can be found [SCS05]. For instance, this alternative way of creating holograms does not rely on the specialized holographic equipment and controlled environments necessary to record a hologram. Another advantage is that with a computer, the propagation of light waves can be simulated, allowing, for example, three-dimensional images of nonexistent objects to be generated. CGH also gives total control of the light waves, making them easier to store, manipulate, transmit and replicate.

The concept of Computer Generated Hologram was first developed by Adolf Lohmann and Byron Brown in 1966 [BL66]. These two scientists used the Fourier transform

to create holographic patterns that could be reconstructed later. After that, other techniques with the same basis were proposed [BL69] [Lee70] [Bur70] [Wal68]. One of the techniques, proposed by Walters [Wal66] in 1966, said that a hologram could be generated with an approach where an object could be decomposed in several light points. However, this approach has complex structures with a lot of information, and so, there was not enough computational power to generate holograms with good quality without having to wait a considerable amount of time.

Other techniques arose over the years, trying to reduce the computational power needed. A different type of holograms was suggested by Toyohiko Yatagai [Yat76], having the work of Nicholas George and J. T. McCrickerd [GM69], who created the first stereographic hologram, as basis. These holograms involve the capture of images from several different perspectives of an object. So, each human eye receives a slightly different perspective of the object. This technique uses the stereoscopic effect as a way to create the idea of depth. However, the quality of these holograms is not as good and the transition between images can be noticed if not enough images are used. The positive side is that holograms of this type are generated much faster than the previous ones.

In the 90s great technological advancements were made. Computers kept growing more powerful and digital holography evolved because of that. Tasks that before lasted for hours now could be made in minutes or seconds. These technological advancements allowed holograms to be created faster and with better quality. So, interest in creating holographic video started to appear. Mark Lucente was the first to attempt to produce holograms in real time. Lucente created a method called "diffraction-specific fringe computation" [Luc], where is used spatial and spectral discretization, and also precomputed primitives, called *basis fringes*, to generate the patterns rather than using mathematically intensive interference computations. This technique achieves good results in terms of speed, being one of the techniques used to generate holographic video.

Its possible to divide CGH into two big classes: Fourier holograms and Fresnel holograms. The main difference between these two types of holograms is the distance

between the object and the hologram.

In the Fourier holograms, it is assumed that the object is located at a large distance
from the hologram. In this case, the object wave on the hologram plane can be
calculated using the Fourier transform of the wavefront at the object plane [BL69]
[Wal68]. Fourier holograms are a good starting point for computing holograms
where the object is far way, but the images are not optimal for display holography.
The images present low quality and lack of 3D can be easily detected [BBJ08,
Chapter 19] [Poo06, Chapter 8].

On the other hand, in the Fresnel holograms the object is near the hologram, which
offers a greater capacity to create realistic images for display. It can offer higher
resolutions and can reconstruct deep three-dimensional scenes.

The difference in computational complexity between Fourier and Fresnel holograms
is substantial. Because Fourier holograms use the Fourier transform, a solution
can be quickly obtained using fast algorithms or optimized hardware. In contrast,
computing Fresnel holograms has no simple analytic approach. Most of the work
done in this area is research on more efficient numerical techniques for the generation
of Fresnel holograms [BBJ08, Chapter 19].

Fresnel holograms usually fall into one of three categories [BBJ08, Chapter 19]:
physically based techniques, holographic stereogram techniques and hybrids of the
previous two. On this work stereographic holograms will be left aside, since they
are a family of methods whose purpose is to avoid the time-consuming calculation
of wave interference. Only the physically based technique will be considered, since
it focuses on the interference modeling of waves, producing patterns similar to the
ones produced with optical holography.

2.3.1 Physically-Based CGHs

The recording of a CGH consists in creating a interference pattern that later can
be used to diffract light and create a three-dimensional virtual object when a user
looks at it. This can be achieved with the calculation of the resulting intensity of

the complex amplitude of the object wave interfered with a reference wave in the hologram plane. Thus, a digital hologram is a data structure composed by points disposed in a bidirectional plane. And for each point the respective intensity needs to be calculated.

The input of the algorithms to record a hologram can be real or virtual. Real data is given through photos. However, this presents a problem, since photos don't have all the information about the three-dimensional objects. To solve this problem, all the information about the objects and their three-dimensional models is provided to the algorithm. In the case of virtual data, 3D modeling programs are used.

Physically based techniques create interference patterns similar to the ones created using classic holography. To that effect, the object is decomposed into light points [BBJ08, Chapter 19]. These light points are usually located at the surfaces of the objects. Each one of these points is independent from the others and its interference with the reference wave needs to be calculated.

Assuming that there is no time dependence and propagation amplitude attenuation, simplified versions of the reference and object waves can be achieved.

Considering the reference wave as a plane wave (as discussed before) traveling on the xz-plane with an amplitude A_r, wavelength λ, initial phase φ_0 and angle of incidence with the plane where the hologram will be recorded θ_{ref}, the reference wave can be expressed by [BBJ08, Chapter 19] [Poo06, Chapter 8]:

$$R(x, z) = A_r e^{-i(k(x \sin \theta_{ref} + z \cos \theta_{ref}) + \varphi_0)} \tag{2.34}$$

Considering one of the light points of an object, with coordinates (x_0, y_0, z_0) at a distance d of the plane, where $d = \sqrt{(x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2}$. The wave emitted by the light point, with an amplitude A_p, wavelength λ and initial phase ψ_0 can be expressed by [BBJ08, Chapter 19] [Poo06, Chapter 8]:

$$P(x, z) = A_p e^{-i(kd + \psi_0)} \tag{2.35}$$

On both equations, k represents the wave number and is given by $k = 2\pi/\lambda$.

So, the object wave is composed by the waves of N light points. Summing the N light waves, the object wave is obtained.

$$O(x, z) = \sum_{i=0}^{N} P_i(x, z) \tag{2.36}$$

This way, the intensity on the hologram plane can be calculated as:

$$
\begin{aligned}
I(x, z) &= |O(x, z) + R(x, z)|^2 \\
&= |O(x, z)|^2 + |R(x, z)|^2 + 2Re\{O(x, z)R^*(x, z)\}
\end{aligned}
\tag{2.37}
$$

2.3.2 Bipolar Intensity method

In 1992, Mark Lucente [Luc92] analyzed equation 2.37 and discovered that each of the terms of the equation does something different in the generation of the hologram.

The term $|O(x, z)|^2$ represents the interference of the object light points with each other, also known as *object self-interference*. In computational and optical holography this term often introduces unwanted image artifacts. The term $|R(x, z)|^2$, called *reference bias*, represents a slight increase of the hologram's general intensity. The final term $2Re\{O(x, z)R^*(, x, z)\}$ is responsible for the creation of the interference fringes between the object light waves and the reference wave, which when illuminated produce the holographic image.

In optical holography there are ways of decreasing the appearance of unwanted image artifacts and of the reference bias. However, in computational holography its possible to just ignore these two terms. This will result in the creation of holograms with more quality and at the same time decrease the time needed to generate them.

So Lucente simplified the equation 2.37, creating the "bipolar intensity method", which is expressed by:

$$I(x, z) \cong 2Re\{O(x, z)R^*(x, z)\} \tag{2.38}$$

The equation 2.38 can then be developed using the equations 2.34 and 2.36 [Luc92] [BBJ08, Chapter 19]. Assuming that the hologram plane is positioned in the xy-

plane and $z = 0$, the following expression is obtained:

$$I(x, z = 0) \cong 2Re\{\sum_{j=0}^{N} A_{pj}e^{-i(kd_j+\psi_{0j})} \times A_r e^{i(kx\sin\theta_{ref}+\varphi_0)}\}$$

$$\cong 2A_r \times \sum_{j=0}^{N} A_{pj}\cos(k(x\sin\theta_{ref} - d_j) + \phi_{0j}) \qquad (2.39)$$

Where $d_j = \sqrt{(x - x_{0j})^2 + (y - y_{0j})^2 + z_{0j}^2}$ and ϕ_0 represents the initial phases of the reference and i^{th} object waves together. Additionally, it is possible to remove A_r in order to eliminate any uniform bias. The final intensity pattern in the hologram plane is then given by:

$$I(x, z = 0) \cong \sum_{j=0}^{N} A_{pj}\cos(k(x\sin\theta_{ref} - d_j) + \phi_{0j}) \qquad (2.40)$$

Chapter 3

Planning of the virtual lab

3.1 The laboratory

The main idea behind the construction of the laboratory for computer generated holography is for the laboratory to be an interactive web site, which takes the form of a tree-dimensional scene. In order to do that, the concepts discussed on the previous chapter are used, and the physically-based "bipolar intensity method" is implemented so the holograms can be recorded using an algorithm that simulates the phenomenon of interference of waves. The use of this method also allows the holograms to be recorded in an acceptable amount of time. Although only one method of generating holograms is implemented, an effort was made to implement the laboratory using a structure that allows new methods to be easily added by others who want to contribute to the project.

3.1.1 The Scene

The laboratory has a scene, which is the core of the project, illustrating the recording and reconstruction of a digital hologram, and is composed by illustrations of the equipment required to record and reconstruct a hologram (such as the light source, the object, a beam splitter, mirror(s), various lens, and the holographic plate), as well a representation of the light as beams or wavefronts. This scene represents

the overview of what is happening and it is be possible to see the final interference pattern on the representation of the holographic plate. Figure 3.1 illustrates what was planned.
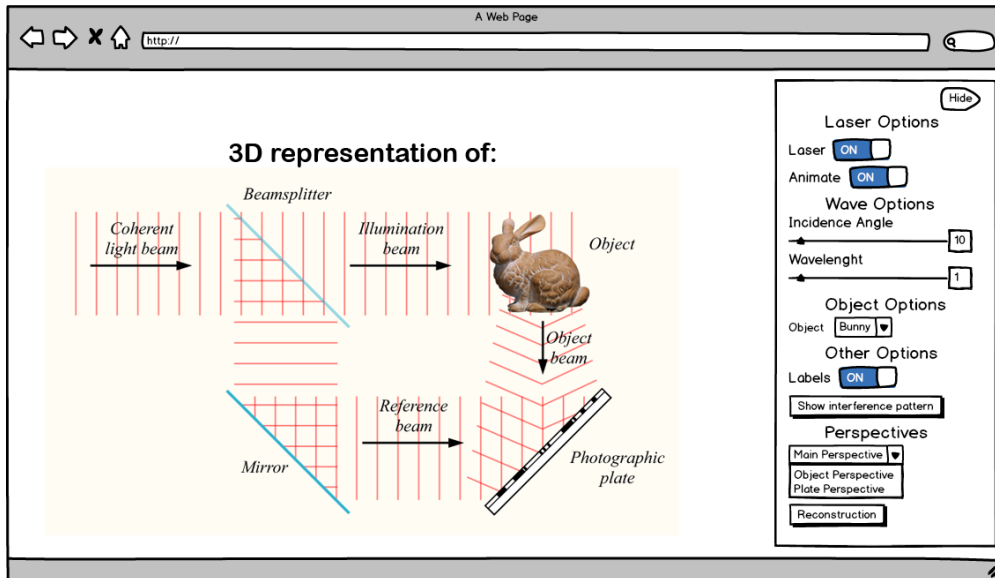
Figure 3.1: *Mockup of the main perspective of the scene, showing the most important parameters. Setup image taken from wikipedia and bunny image taken from waldyrious.net/learning-holography website, called "Learning Holography"*

The user can choose to be taken to different perspectives of the scene. Each of these perspectives corresponds to a zoom-in on a particular component. On these perspectives the user is able to see with more detail what is happening with that particular component and how it contributes to the hologram. For example, selecting the representation of the object takes the user to a perspective where it is shown that the object is decomposed into light points, that each point emits an individual wave and that the object wave is the sum of all these individual waves, as illustrated in Figure 3.2. Selecting the representation of the holographic plate takes the user to a perspective where they can have a closer look on the interference pattern. Figure 3.3 illustrates what was planned.

It is also possible to swap between the recording process, shown above, and the reconstruction process, as long as an interference pattern has been generated. This illustration is treated as a second *mode* of the scene. A mode is a setup of the scene.

Figure 3.2: *Mockup of the object perspective of the scene, showing the most important parameters. Bunny image taken from a page in the website "Learning Holography"*

Different modes have different setups, equipment and serve different purposes. So, the scene has a recording mode where three perspectives are used and a reconstruction mode where only the main perspective is used. On the reconstruction mode the layout of the equipment changes to match the requirements of the reconstruction of holograms. This mode is used only for didactic purposes. That is, no physically-based formula or algorithm is used for the reconstruction. In this case the goal is simply to illustrate the process, in order to show that holography is a two step process and that the reconstruction step is the one that creates the virtual image. Figure 3.4 illustrates what was planned.

3.1.2 Parameterization and user interface

For the parameterization, the main plan was to have a side panel that can be shown or hidden by the user, where the parameters can be changed. The user can play with the most basic and some of the core parameters that influence the creation of holograms. Depending of which perspective the user is seeing, the settings change their positions, making it easier to find the settings that influence that perspective the most. Settings that do not influence what is happening on a particular perspective
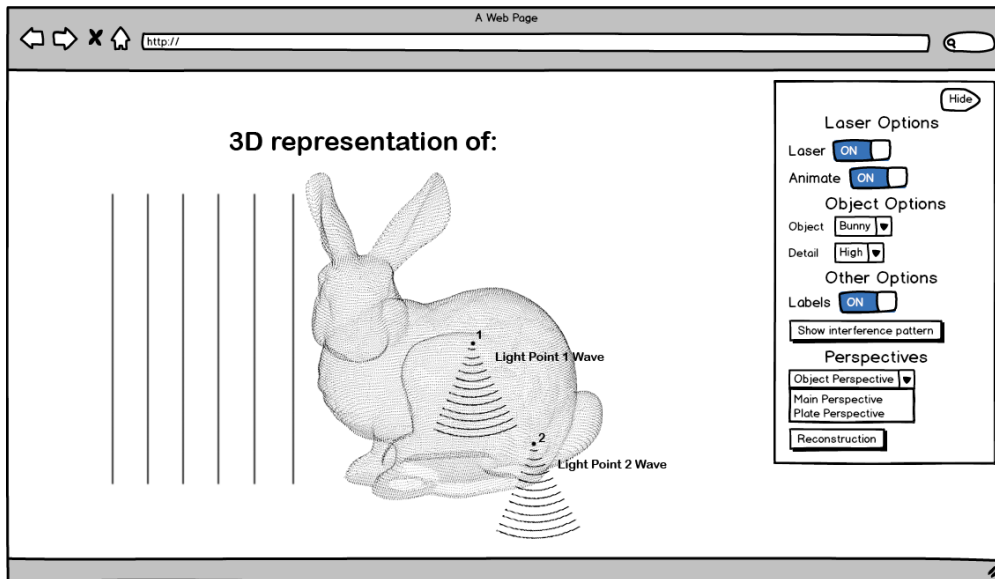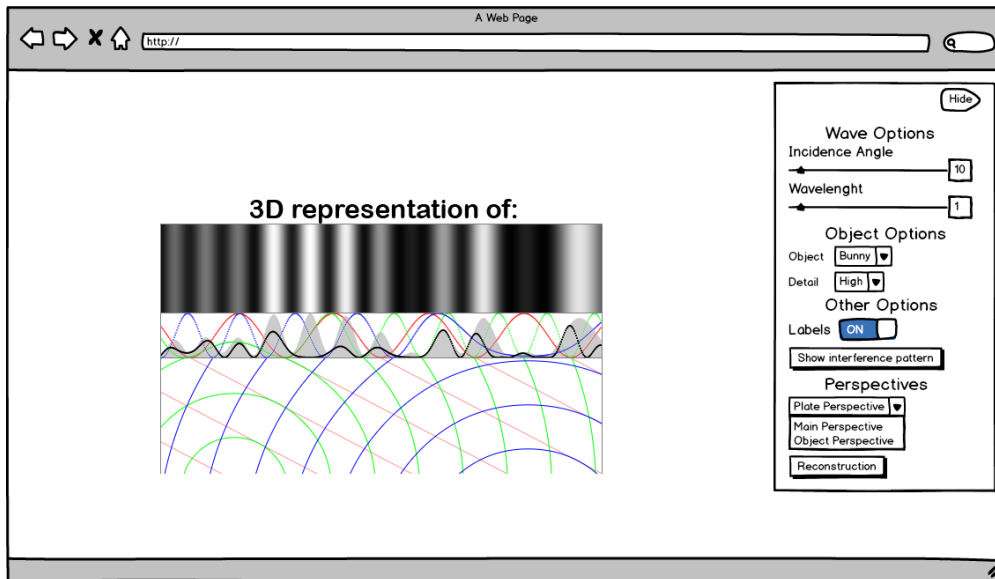
Figure 3.3: *Mockup of plate perspective of the scene, showing the most important parameters. Interference image taken from a page in the website "Learning Holography"*

are hidden.

So, for example, on the main perspective, parameters like the wavelength and angle of incidence of the reference wave are highlighted, while on the object perspective the most important settings are the number of light points in which the object is divided or the ability of swapping the object itself for another model. Settings related to the reference wave or the plate where the hologram is recorded are hidden on the object perspective, because they do not influence what is happening with this particular component. On the plate perspective the user is able to change every parameter that influences the interference pattern. Figures 3.1, 3.2 and 3.3 illustrate what was planned.

The core parameters for each perspective and possible modes can be seen on Table 3.1.

All the changes made in one perspective have an impact on the scene and by consequence on the other perspectives. As the scene is three-dimensional, the camera can be controlled by the user to spin around the scene and zoom in and out to some degree. The geometry is properly labeled on all perspectives so that new users

3D representation of:

Virtual
image

Reconstruction
beam

Photographic
plate

Reconstructed
wavefronts

Viewer

Hide

Laser Options

Laser ON

Animate ON

Other Options

Labels ON
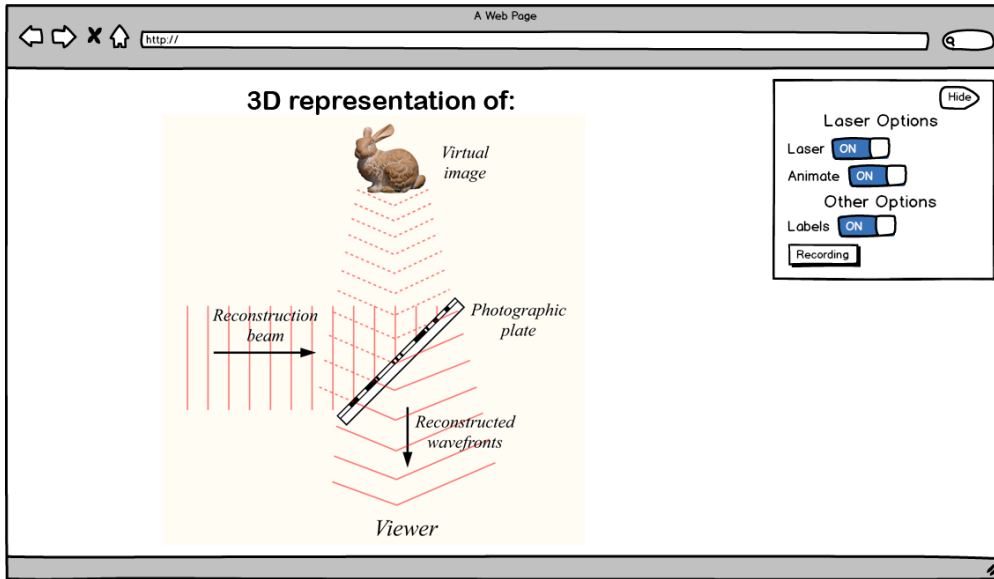
Recording

Figure 3.4: *Mockup of the reconstruction mode of the scene, showing the most important parameters. Setup image taken from wikipedia and bunny image taken from the website "Learning Holography"*

Mode			
Recording			**Reconstruction**
Main Perspective	Object Perspective	Plate Perspective	Main Perspective
Wavelength (1) Angle (2) Geometric Figure	Geometric Figure Object Detail	Wavelength (1) Angle (2) Geometric Figure Object Detail	

Table 3.1: *Core parameters for each perspective and possible modes. (1) Reference wave wavelength; (2) Reference wave angle*

know the components that are being represented. These labels can be turned off at any time. Each perspective also has a pop-up information box, that appears when the user presses the "more information" button on the page. This information box shows more information about what is happening on the components of that perspective and what the user can do with the interface of that perspective.

3.2 Technological tools

In order to make the laboratory web-based and interactive, the use of javascript is necessary. Javascript is a scripting language used to provide interactivity to web sites. To construct the three-dimensional environment required for the laboratory, the web version of the OpenGL library, called WebGL, is used.

WebGL is a javascript API that implements a subset of functions of OpenGL in order to create interactive three-dimensional scenes in the web browser without requiring the user to install anything. The use of javascript allows the laboratory to be client-based. This means that all calculations use the users' computer. The user's GPU will be used in the rendering of the three-dimensional environment and in the generation of the interference patterns.

Over the years people have been making use of WebGL to construct high-level frameworks that simplify the use of some of the features present in WebGL, and include new ones. These frameworks simplify the use of features like scenes, cameras, lights, shadows, materials, shaders, geometric primitives, data loaders, among others. The main goal of these frameworks is to abstract the user from some of the complexities of the use of WebGL, making it easier to program and not requiring the same amount of code as pure WebGL does. One of such frameworks is *three.js*[1], and it was the library chosen to help create the laboratory.

---

[1]`http://www.threejs.org`

Chapter 4

Implementation of the virtual lab

This chapter is intended to explain how the laboratory is implemented, show the general structure, main features and user interface implementation. This laboratory is built using the *three.js* framework, which simplifies the implementation. Other javascript libraries are used to improve the laboratory experience in certain aspects and will be detailed in the coming sections.

4.1 Laboratory structure

The structure of the laboratory is based on a series of classes, the **MainScene** class being the the center of the application. This class has three perspectives: the main perspective, the object perspective and the plate perspective. The main perspective is centered on the overall scene, and is represented by the **MainPerspective** class; the object perspective is centered on the events related to the object from which the hologram will be created, being represented by the **ObjectPerspective** class; and the plate perspective is centered on the events related to the holographic plate where the interference pattern will be recorded, corresponding to the **PlatePerspective** class. The **MainScene** class also comprises a light wave, represented by the **Wave** class, and an object (used to create the hologram), represented by the **HoloObject** class. The **HoloObject** class has a list of light points, represented by the

LightPoint class. The class diagram on the Figure 4.1 illustrates these relations.
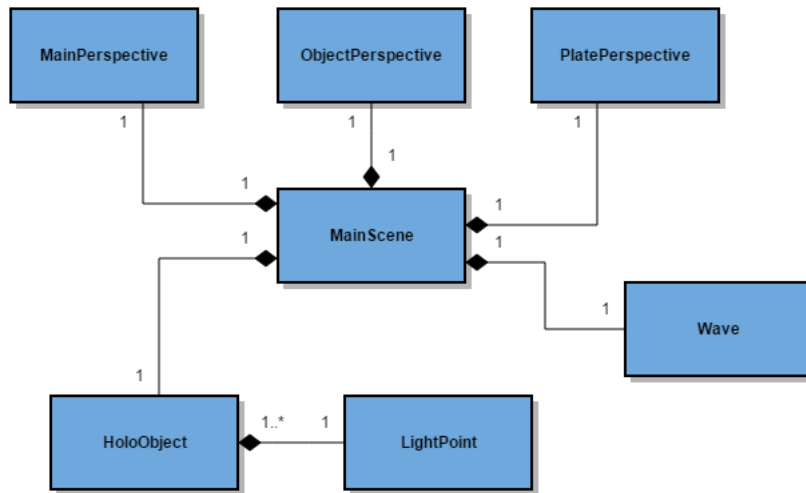
Figure 4.1: *Class diagram of the laboratory*

Being **MainScene** the center of the application, this is where most of the features of the application are implemented, more precisely the ones that affect all the perspectives. More about this will be discussed on section 4.2. The classes of each perspective are where the features that only affect that perspective are implemented. More about this will be discussed on section 4.3. The **Wave** class has the properties of the wave used on the main scene. The **HoloObject** class has all the properties of the object used to create the hologram and the necessary features to manipulate it. Last, the **LightPoint** class has all the properties of each light point that composes the **HoloObject**.

4.2 The Main Scene

As said before, the **MainScene** class is the center of the application. This class is responsible for generating the three-dimensional scene that illustrates all the phenomena involved in the creation of holograms. This scene has two modes: a recording mode, and a reconstruction mode. As said before, when changing modes the setup of the scene changes. The recording mode is responsible for the creation of an interference pattern from a physical object, while the reconstruction mode is responsible

34

for the creation of a virtual object from an interference pattern. The scene is initialized in recording mode, but the modes can be swapped after an interference pattern has been created.

The way the three-dimensional scene is constructed causes the position and rotation of all components involved to be dependent of the position and rotation of the holographic plate. This way, by only defining the Plate Position (PP) and Plate Rotation (PR), all other components will be properly placed on the scene. For this laboratory the position chosen for the plate is $x = 0, y = 80, z = 0$. The value of y is 80, so that all the points of the plate have a positive y value. The rotation of the plate is $0°$. This way the plate will be placed on the xy plane, with $z = 0$.

Hereupon, it is possible to define the directions that other components will have relative to the plate. These components are: the mirror, the object, the laser source, the beam splitter, two concave lens and one convex lens.

The Mirror Direction (MD) is initialized making a $45°$ angle with the normal vector of the plate. This angle is the Reference Wave Angle (RWA), the angle that the reference wave makes with the plate. Figure 4.2 illustrates this setup.
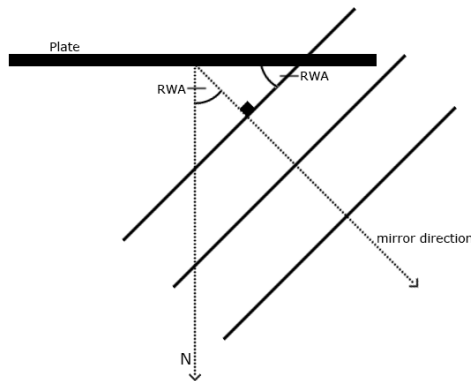

Figure 4.2: *Illustration of how the reference wave angle is related to the angle between the direction of the mirror and the normal of the plate*

So, the direction of the mirror is the vector:

$$MD = (\sin(PR + RWA), 0, \cos(PR + RWA))$$

The Object Direction (OD) also makes a 45°angle with the normal vector of the plate. So, the direction of the object is the vector:

$$OD = (\sin(PR - \tfrac{\pi}{4}), 0, \cos(PR - \tfrac{\pi}{4}))$$

The Laser Source Direction (LSD) makes a 90°angle with the direction of the object. So, the direction of the laser source is the vector:

$$LSD = (\sin(PR + \tfrac{\pi}{4}), 0, \cos(PR + \tfrac{\pi}{4}))$$

The Beam Splitter Direction (BSD) is the same direction as the Object Direction:

$$BSD = (\sin(PR - \tfrac{\pi}{4}), 0, \cos(PR - \tfrac{\pi}{4}))$$

The direction of the concave lens that stands between the beam splitter and the object, Concave Lens Direction 1 (CcLD1), is same as the laser source direction. So, CcLD1 = LSD. The direction of the concave lens that stands between the mirror and the plate, Concave Lens Direction 2 (CcLD2), is the same as the mirror direction. So, CcLD2 = MD.

The direction of the convex lens that also stands between mirror and the plate, Convex Lens Direction (CvLD), is also the same as the mirror direction. So, CvLD = MD.

With the directions defined (all direction vectors are normalized), it is possible to calculate the position of each component in the scene. To do that, distances between components need to be defined. On this laboratory the distances used are:

- Initial distance between the plate and the mirror: 350 units.

- Distance between the plate and the object: 450 units.

- Distance between the laser source and the object: 450 units.

- Initial distance between the mirror and the beam splitter: 450 units.

- Distance between the concave lens 1 and the object: 300 units.

- Distance between the concave lens 2 and the plate: 300 units.

- Distance between the convex lens and the plate: 200 units.

The formula to calculate the position of each one of the components can be found on Table 4.1.

Position	Formula
Mirror Position (MP)	$MP = PP + MD \times 350$
Object Position (OP)	$OP = PP + OD \times 450$
Laser Source Position (LSP)	$LSP = OP + LSD \times 450$
Beam Splitter Position (BSP)	$BSP = MP + BSD \times 450$
Concave Lens Position 1 (CcLP1)	$CcLP1 = OP + LSD \times 300$
Concave Lens Position 2 (CcLP2)	$CcLP2 = PP + MD \times 300$
Convex Lens Position (CvLP)	$CvLP = PP + MD \times 200$

Table 4.1: *Formulas to calculate the position of the components*

In order to the scene makes sense it is also necessary that each one of the components undergoes a rotation, which is dependent of the rotation of the plate. The formulas to calculate the rotation of each one of the components can be found on Table 4.2.

Rotation	Formula
Mirror Rotation (MR)	$MR = PR - \frac{\pi}{2} - ((\frac{\pi}{4} - RWA)/2)$
Object Rotation (OR)	$OR = PR + \frac{\pi}{4}$
Laser Source Rotation (LSR)	$LSR = PR + \frac{\pi}{4}$
Beam Splitter Rotation (BSR)	$BSR = PR + \frac{\pi}{4}$
Concave Lens Rotation 1 (CcLR1)	$CcLR1 = PR + \frac{\pi}{4}$
Concave Lens Rotation 2 (CcLR2)	$CcLR2 = \pi - \arccos(BSD \cdot -MD) - \frac{\pi}{4}$
Convex Lens Rotation (CvLR)	$CvLR = \pi - \arccos(BSD \cdot -MD) - \frac{\pi}{4}$

Table 4.2: *Formulas to calculate the rotation of the components*

The reason for the more complex formulas for MP, CcLR2 and CvLR is that the mirror can change its position, and because of that the rotations of the components involved have to be updated as well.

During the setup of the scene the reference wave is also created. This wave is created using the **Wave** class and it is initialized with a wavelength of 1 unit and amplitude of also 1 unit. The object is also initialized, using the class **HoloObject**. This class

creates a **HoloObject** that has the position and rotation defined for the object but the geometry and light points are not defined yet.

The function that initializes the scene uses these positions and rotation angles to then create the geometry necessary to illustrate the several components and position them correctly in the scene. Some of the geometry is generated with the help of the *csg.js*[1] library, which allows operations of union, subtraction, and intersection on the geometry. In order to this library work properly, a plugin for *three.js* has to be used. So, the *ThreeCSG*[2] library is also used on this laboratory. The geometry of the object is set using the *setObject* function of **HoloObject**. That function, when called, generates the geometry of one **HoloObject** instance according to the geometric figure chosen at the time. There are four possible choices: Cube, Sphere, Cylinder and Torus Knot. The geometry can have several levels of detail, which is the number of light point used to represent the object. At the moment of creation the geometry has always the *low* detail level for performance purposes. After the geometry is defined, a function responsible for generating the list of light points is called. The **LightPoint** class is composed by a vector of four positions. The first three positions store the position of the light point while the last one stores the initial phase of the wave emitted by that light point. Each light point has the position of one vertex of the object and all have initial phase 0 to simplify the calculations. Setting the initial phase can also be useful for the creation of a more realistic version of the lab. For each vertex, a **LightPoint** is created. The function that initializes the scene also fills some other arrays of positions (vertexes of the mirror, plate and beam splitter) necessary for the proper behaviour of the laser. The labels of shown components are also initialized on this function. **MainScene** uses two *three.js* scenes (the class used by the framework to describe a tree-dimensional scene) to construct the overall scene. One of the scenes is for all the components and the other is for labels only. Both *three.js* scenes are rendered in the same camera, so all the geometry in both scenes will appear on the screen. This is so that labels are always visible and never obstructed by the rest of the geometry. The *three.js* scene with

---

[1]http://evanw.github.io/csg.js/

[2]https://github.com/chandlerprall/ThreeCSG

the labels is rendered second so they are always on top. Figure 4.3 illustrates how the scene is setup initially and also shows a screenshot of the laboratory using that setup.
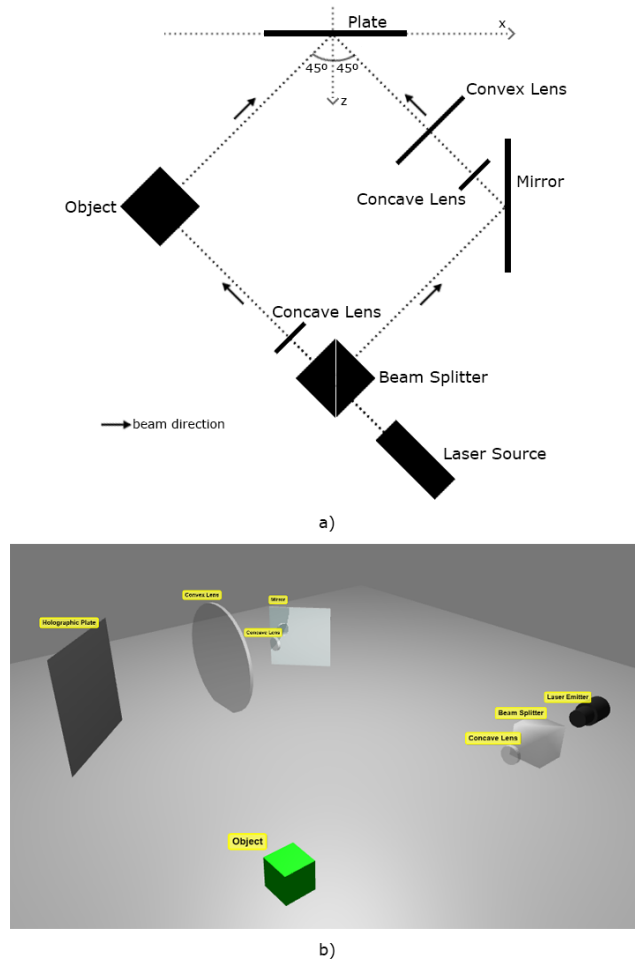
Figure 4.3: *a) Laboratory's initial setup illustration; b) Print-screen of the laboratory using the setup*

4.2.1 Recording mode

When the laser is turned on while in recording mode, a representation of a light beam and respective labels are added to the scene. This representation has two variants. One is the simple laser, which is a static representation of the light beam. The other is the animated laser, which is a version of the light beam where it is possible to see the wavefronts travel through the scene.

The way the simple laser is implemented and behave is very straightforward. This variant uses cylinders, cones and other convex solids to represent the reference and object waves. All the geometry is stored on an array so can be easily managed later. When the laser is turned on all the geometry is placed at the same time and never changes, unless the object is swapped or the mirror is moved. In the case of a swapped object, the object wave needs to be updated. In the case of a moved mirror, the reference wave needs to be updated.

The beam starts at the laser source, being represented by a cylinder, and travels through the beam splitter, where is duplicated, until arrives at the concave lens. Here, the beam is expanded so as to cover the whole object. A cone is used to represent the expanded beam. The object emits the object wave, represented by a convex solid created from the vertexes of the object and holographic plate. The duplicated beam, also represented as a cylinder, travels until arrives at the mirror where is reflected. The reflected beam, represented as a cylinder, travels to the concave lens, where is expanded like in the other concave lens, so a cone is used to represent it. Then, the laser travels to a convex lens so it stops expanding. From there, the beam travels to the plate, being represented by a wider cylinder. Figure 4.4 shows a screenshot of this setup.

Figure 4.4: *Screenshot of the laboratory using the simple laser in "Recording Mode"*

On the other hand, the animated laser has a more complex implementation and behaviour. In this variant the light beam is updated on every frame. When the laser is turned on, an initial wavefront is created and added to a data structure that

stores the wavefronts that travel from the laser source to the object. This structure has four arrays: one array to store the geometry of the wavefronts; one array of booleans which state, for each wavefront, if the next wavefront has already been sent; one array of booleans which state, for each wavefront, if it has already crossed the beam splitter and the duplicate has been created; and a final array of booleans which state, for each wavefront, if it has crossed the object.

```
var laserLight1 = {
    list: [], //where wavefronts are stored
    next: [], //booleans that state if the next wavefront was sent
    beam: [], //booleans that state if each wavefront crossed the beam
    splitter
    object: [] //booleans that state if each wavefront crossed the
    object
};
```

Listing 4.1: *Structure that stores wavefronts that travel from the laser source to the object*

Every time a wavefront is at a distance from the laser source equivalent to 100× the wavelength of the wave, a new wavefront is created, sent and added to the structure from Listing 4.1. The previous wavefront is then marked as already having a "next". The reason to multiply the wavelength by 100 is for the illustration not to become overcrowded with geometry, which could lead to poor performance of the laboratory, as well as its misunderstanding.

Once the wavefronts start to arrive at the object, the wavefronts of the object wave start to be sent. Two things can happen depending on the perspective selected at this moment. If the main perspective is selected, each wavefront that arrives at the object results in the creation of another wavefront (of the object wave) with the form of the object, which grows bigger when closer to holographic plate. Each created wavefront of the object wave is added to a data structure that stores the wavefronts of the object wave that travel from the object to the plate. This structure is composed by one array that stores the geometry of the wavefronts.

```
var objWaveLight = {
```

```
    list: []  //where wavefronts are stored
};
```

Listing 4.2: *Structure that stores wavefronts that travel from the object to the holographic plate on the main perspective*

If the object perspective is selected, when each one of the wavefronts crosses the center of the object a new wavefront is sent from each one of the light points at the same time. In nature, the wavefronts are not all sent at the same time but for the generation of the interference pattern that is irrelevant because there isn't a single "object wave", but a complex multi-wave made up of countless point-source object waves, and it's the phase difference of each pair of object element-wave and reference wave that is important. The waves emitted are represented by spheres, with their center at the position of their light point, that grow until they collide with the plate. However, only part of the spheres is rendered in order to maintain the laboratory simple and with decent performance. The wavefronts created are stored on a data structure. This structure is composed of two arrays. One of them stores the geometry of the wavefronts and the other stores the current scale value of each wavefront.

```
var lightPointWaves = {
    list: [],  //where wavefronts are stored
    scales: []  //where the scale values are stored
};
```

Listing 4.3: *Structure that stores wavefronts that travel from the light points of the object to the holographic plate on the object perspective*

Independently of the perspective, once the wavefronts arrive at the object they are marked as arrived on the structure of Listing 4.1.

The wavefronts that cross the splitter are duplicated and marked as such on the structure of Listing 4.1. The duplicated wavefronts are added to a data structure. This structure is composed by two arrays. One array that stores the geometry of the wavefronts and the other is an array of booleans that mark the wavefronts as arrived at the mirror.

```
var laserLight2 = {
    list: [], //where wavefronts are stored
    mirror: [] //booleans that state if each wavefront arrived at the
    mirror
};
```

Listing 4.4: *Structure that stores wavefronts that travel from the beam splitter to the mirror*

The duplicated waves leave the splitter and travel in the direction of the mirror while the original ones continue to travel in the direction of the object.

Every time a wavefront finds the mirror it is marked as reflected in the structure of Listing 4.4 and a new wavefront is created and saved as a reflection on a new data structure. This new structure is composed by two arrays. One array is used to store the geometry of the wavefronts. The other is an array of booleans to mark the wavefronts as arrived at the plate. This second array is only used in the reconstruction mode. The beam of light that travels from the mirror to the holographic plate is the one called reference wave or reference beam.

```
var laserLight3 = {
    list: [], //where wavefronts are stored
    plate: [] //booleans that state if each wavefront arrived at the
    plate
};
```

Listing 4.5: *Structure that stores wavefronts that travel from the mirror to the holographic plate*

Also, when the wavefronts traveling between the beam splitter and the object encounter the concave lens positioned between these two components, they will start to expand, scaling with the distance to the object. This way the entire object is illuminated by the beam. The same happens with the wavefronts traveling between the mirror and the plate when they encounter the concave lens positioned between them. But in this case the wavefronts only expand until they encounter the convex lens, also positioned between the mirror and plate. This happens so that the reference wave continues to be a plane wave and it can illuminate the entire plate.

43

Figure 4.5 shows screenshots of the use of the animated laser on both main and object perspectives.
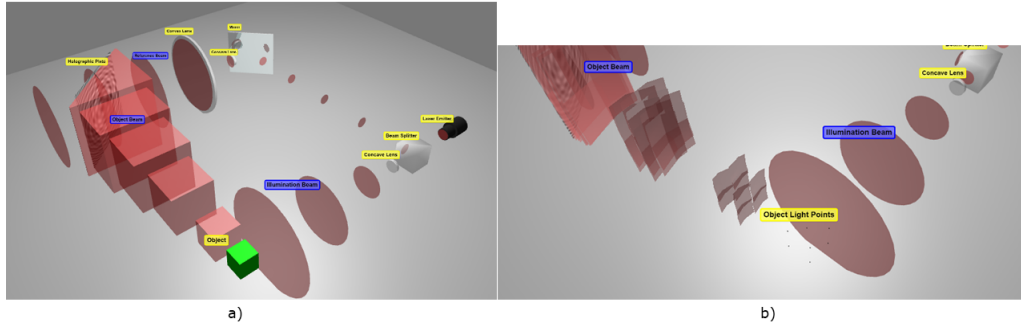

Figure 4.5: *a) Screenshot of the laboratory using the animated laser with main perspective selected; b) Screenshot of the laboratory using the animated laser with object perspective selected*

When both reference and object wave arrive at the plate, a flag for each wave is set to *true*. When both flags are *true* the interference pattern is shown.

All the wavefronts are deleted from the scene and respective structures after arriving at their destinations in order to avoid consumption of computational resources unnecessarily. All the structures are cleaned once the laser is turned off.

4.2.2 Reconstruction mode

As said before, after an interference pattern is recorded on the plate it is possible to change to the reconstruction mode. This mode changes the scene slightly. The biggest changes are the removal of the object and the replacement of the concave lens placed between the beam splitter and the object by an obstacle that prevents the beam from passing through it. This obstacle uses the CcLP1 from Table 4.1 and CcLR1 from Table 4.2 to position itself correctly in the scene. This mode uses only the main perspective, so the scene changes automatically to this perspective if it is not selected. Figure 4.6 illustrates the changes in the setup and also shows a screenshot of the laboratory using that setup.

If the laser is already turned on it will restart. Like in the recording mode, here the
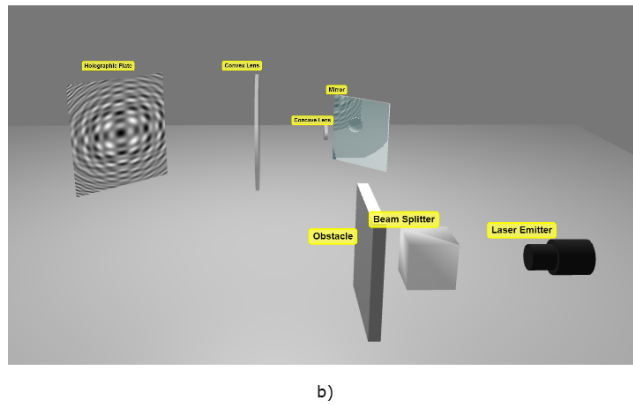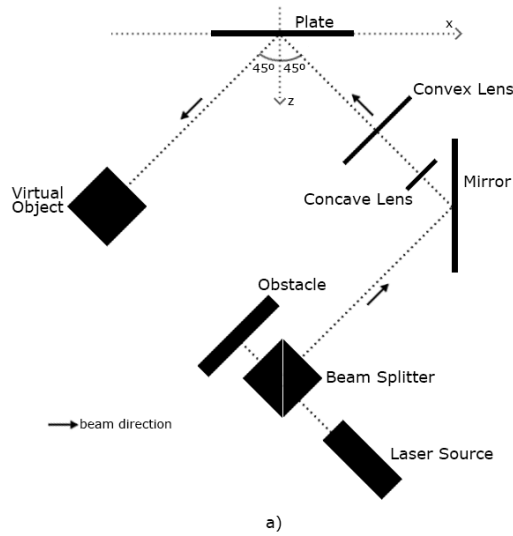
Figure 4.6: *a) Laboratory's reconstruction setup illustration; b) Print-screen of the laboratory using the setup*

same two variants for the representation of the light beam exist, the simple laser and the animated laser.

Both simple and animated lasers are implemented and work in the same way as in the recording mode, for the most part. One of the biggest differences is that the beam that starts at the laser source will stop at the obstacle. The other difference is that when the reference wave reaches the interference pattern on the plate, the reconstructed object wave propagates in the direction of where of the real object was positioned, contrarily to at happens in the recording mode, where the wavefronts travel on the opposite direction. The geometry of the reconstructed wave is the same as the geometry of the object wave of the recording mode. Figures 4.7 and 4.8

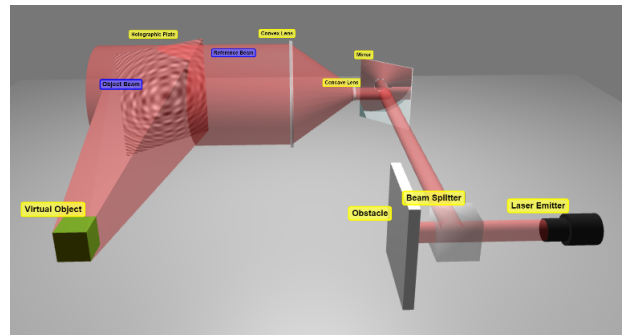show screenshots of these situations.

Figure 4.7: *Screenshot of the laboratory using the simple laser on "Reconstruction Mode"*
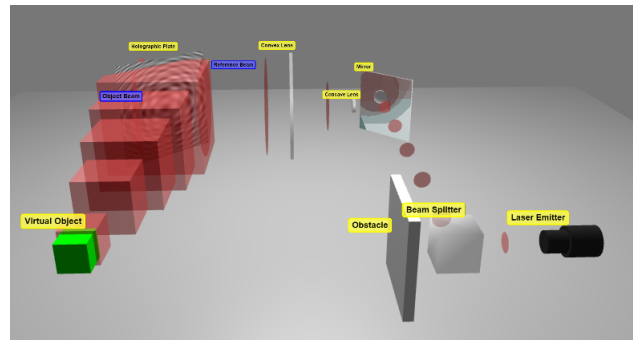

Figure 4.8: *Screenshot of the laboratory using the animated laser on "Reconstruction Mode"*

On both variants, the reconstructed object wave gives the viewer the illusion that an object (virtual object) is in the position where the real object was.

As in the recording mode, all the wavefronts are deleted from the scene and respective structures after arriving at their destinations. The structures are cleaned when laser is turned off and the virtual object disappears.

It is possible at any time return to the recording mode. This action changes the scene to how it was before and the laser starts behaving how it should in that mode. If the laser is on during this transition, it will restart.

4.3 Perspectives

As said before, the scene has three perspectives. While in recording mode it is possible to jump between all these perspectives. When the perspective is changed, the camera is moved and rotated smoothly thanks to the *tween.js*[3] javascript library.

4.3.1 Main Perspective

The core feature from the main perspective is the automatic adjustment of the position and rotation of the mirror when the angle at which the reference wave should arrive at the plate is changed. The position and rotation of the lenses located between the mirror and the plate are also automatically handled.

A new direction for the reference beam is defined, naturally, when the RWA is changed. The new direction of the mirror is defined with the vector MD using the new RWA. A new distance between the plate and the mirror is also defined using the formula:

$$\text{new distance} = \sec(\frac{\pi}{4} - \text{RWA}) \times 350 \qquad (4.1)$$

This way, the Beam Splitter Direction continues the same.

The position of the mirror is then updated, using the formula:

$$\text{MP} = \text{PP} + \text{MD} \times \text{new distance} \qquad (4.2)$$

Figure 4.9 illustrates how the updating of the mirror position works.

The rotation of the mirror is also updated using the formula for MR defined in Table 4.2.

With these changes, the position and rotation of the lenses positioned between the mirror and the plate need to be updated as well. These components need to be kept in the path of the light beam and need to have the same rotation as the wavefronts.

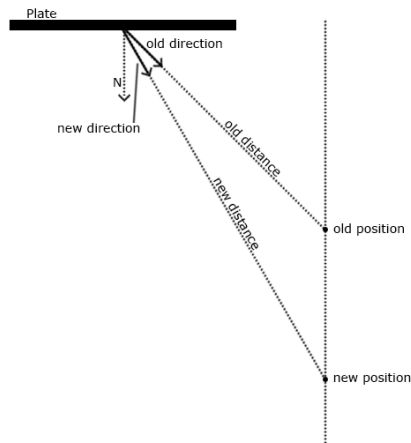[3]https://github.com/tweenjs/tween.js/

47

Figure 4.9: *Illustration of what happens during the update of the position of the mirror due to the change on the reference wave angle*

To do that, the formulas defined for these components in Tables 4.1 and 4.2 are used.

After the components are correctly positioned in the scene, if the labels are turned on, their positions are also updated using the new positions of the components changed. If the laser is on, the position of the label of the reference wave is also updated. Figure 4.10 shows print-screens of the before and after the update of the mirror position and rotation.
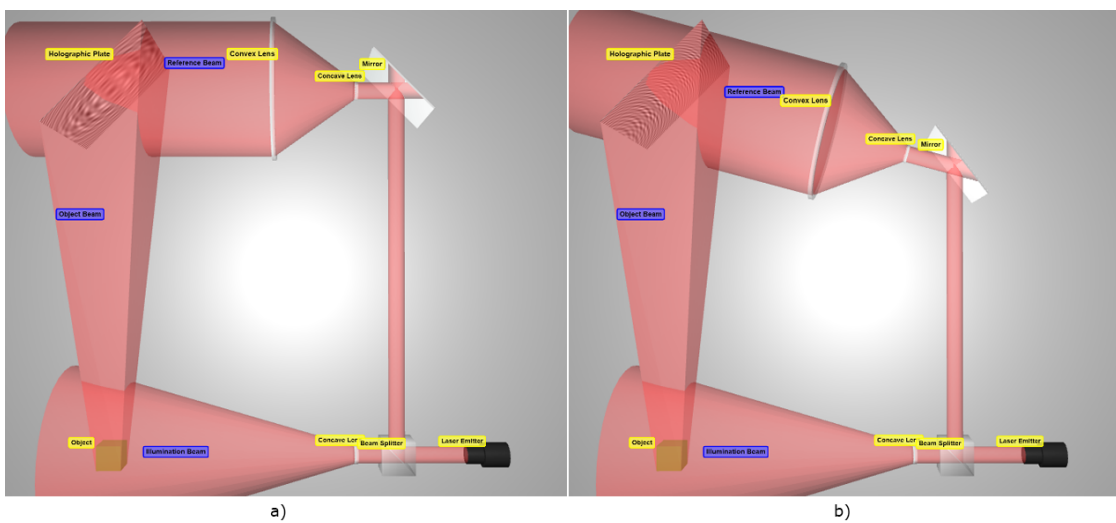

Figure 4.10: *a) Before the update of the position and rotation of the mirror; b) After the update of the position and rotation of the mirror*

When the laser is turned on, it needs to be updated. If the animated laser is being used, the wavefronts that left the mirror until the moment of the update are deleted to avoid inconsistencies. If the simple laser is being used, some of the geometry involved on the illustration needs to be updated. The geometry of the cylinders and cones which represent the light beams traveling from the beam splitter to the plate change, becoming bigger or smaller. Their position and rotation also change, depending on the new position and rotation of the components.

The main perspective also has its own way of changing the object that is being used to create the interference pattern. First, the geometry of the object is changed with the help of a function of the **HoloObject**. The geometry is changed and the light points are updated to match the new geometry. Next, the object wave is updated. If the animated laser is being used, all the wavefronts sent from the object to the plate until the moment of the update are deleted and new ones start to be sent. The interference pattern is also hidden until the new wavefronts arrive at the plate. If the simple laser is being used, the geometry used to represent the object wave is updated. A new convex solid using the vertexes of the plate and the new object is created.

To change the wavelength of the wave emitted by the laser source, a simple function that accesses that variable on the **Wave** class is used.

4.3.2 Object Perspective

The object perspective also has its own features. When the object perspective is chosen, the object in the scene is replaced by its light points, represented by small spheres. The focus of the camera changes to the light points.

When the animated laser is turned on, the class **ObjectPerspective** is responsible for sending the several wavefronts that leave each one of the light points, as was talked in Section 4.2.1.

As in this perspective the object is represented by light points, the function that changes the object is different. First, the previous light points are removed from the

scene. Then, the object is changed using the function from the **HoloObject** class that changes the object. The new light points are then added to the scene using the same function used to replace the object with its light points. If the animated laser is turned on, all the wavefronts sent until the moment of the object change are removed to avoid inconsistencies, and the interference pattern is hidden until the new wavefronts arrive at the plate. If the simple laser is being used, the object wave geometry is updated the same way as in the main perspective.

The object perspective has another feature, one that is not present in the main perspective: the possibility of changing the level of detail that the object has. This feature has a behaviour similar to changing the object. The difference is that in this case, the geometry of the object is replaced by another with a greater or lesser number of light points, however, the geometric figure is still the same. The level of detail is changed by a function from the **HoloObject** class. Figure 4.11 shows screenshots of the object represented by light points using different levels of detail.
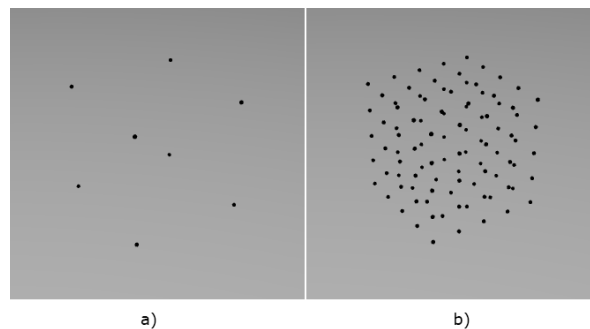

Figure 4.11: *a) Object represented by light points using low detail level; b) Object represented by light points using high detail level*

4.3.3 Plate Perspective

The plate perspective is focused on the holographic plate and on the interference pattern that appears on it. Thus, in this perspective it is not possible to move the camera and the laser is omitted so that the interference pattern can be seen clearly.

The main perspective shares some features with this one. Here it is also possible

to change the RWA and the wavelength of the wave emitted by the laser source, so these features from the main perspective are used in this perspective too. There is no reason to reimplement these features in the plate perspective because the behaviour is exactly the same.

This perspective also has the possibility of changing the object and its detail; however, the functions used are slightly different. Both make use of the functions from **HoloObject**, like the other perspectives, but in this perspective nothing more needs to be made, because the laser is omitted and there is nothing to be updated on that regard.

4.4 Shader Libs

The laboratory makes use of two custom shader libraries. One of them contain the shaders used on the materials of the laser. This custom shader library contains modified versions of the shaders provided by *three.js*. They are modified so that the different stages of the laser beam start and end where they are supposed to. For example, the beam that arrives at the mirror starts to disappear when it hits the plane where the mirror is set. This limit is created using the vertices of the mirror to calculate the plane and then ignoring the fragments that exceed this limit. The same happens to the beam that leaves the mirror. The same method is used to create limits on the holographic plate and beam splitter. A similar approach is used to limit the spheres that represent the wavefronts that leave each one of the light points. In this case, for each light point, the fragments of the spheres that are rendered are limited to the inside of a pyramid created using planes that are generated using the position of the light point and the four vertexes of the holographic plate.

The other library contains the shader used to create the interference patterns. The shader used is a full parallax implementation of the "Bipolar Intensity Method" [Luc92] on the GPU. All light points are used to calculate the intensity in every point of the holographic plate, not checking if some point is occluded by others. The implementation of the algorithm used on the shader is quite simple. The way

51

the intensity of each point of the hologram is calculated is:

$$I = \sum_{j=0}^{N} \cos(\psi_j - \varphi) \tag{4.3}$$

Where ψ_j is the phase of the wave sent by the jth light point when it arrives at the plate, and φ is the phase of the reference wave also when it arrives at the plate.

To find the phase of the reference wave (φ) at the plate, the projection of the wavelength (which is dependent of the RWA) at the plate needs to be calculated. This projection can be calculated using simple trigonometry. Figure 4.12 illustrates this situation.
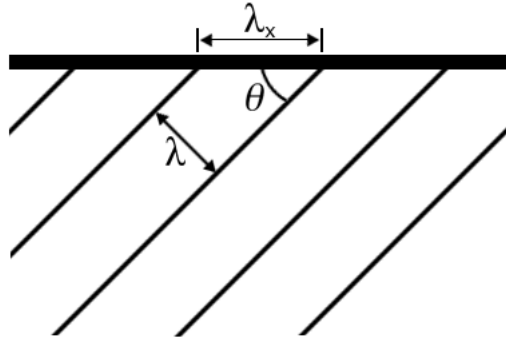
Figure 4.12: *Illustration of what is the wavelength projection and how to calculate it*

Assuming that the wavelength projection is represented by λ_x (the RWA only rotates on the y-axis, so the wavelength is only projected on the x-axis), that the wavelength of the reference wave is λ and that the RWA is represented by θ, the projection is calculated as:

$$\lambda_x = \frac{\lambda}{\sin(\theta)} \tag{4.4}$$

Now, knowing the projected wavelength of the reference wave along the plate, it is possible to calculate the phase of the reference wave for each point of the hologram. All the points with the same x coordinate have the same phase. To obtain the phase at any given point of the hologram, the x coordinate of the point (h_x) is divided by λ_x. The value obtained, which represents the number of cycles completed as h_x grows, is then converted to radians by multiplying it by 2π. The reference wave

phase is then calculated as:

$$\varphi = 2\pi \times \frac{h_x}{\lambda_x} \tag{4.5}$$

To find the phase of the wave emitted by one light point (ψ_j) at the plate, the distance (d) from the light point j to the hologram point needs to be considered. This distance is calculated using the Euclidean distance formula. After that, the initial phase of the wave emitted by the light point (ψ_{j0}) is subtracted from the distance. In this case, the initial phase of all waves emitted is zero, as said in section 4.2. The resultant value of the subtraction is then multiplied by k (how many cycles, in radians, the wave completes per unit of distance). So, the phase of the wave emitted by the light point j at the plate is calculated as:

$$\psi_j = (d - \psi_{j0}) \times k \tag{4.6}$$

4.5 User Interface

The laboratory makes use of the orbit controls available on *three.js* to give mouse control to the user. Thus, the user can use the mouse to rotate the camera around the scene and also zoom-in or zoom-out.

The menus available on the several perspectives and modes are made using the *dat.GUI*[4] javascript library.

The laboratory has a menu with options on it. These options can change to match the needs of the perspective currently selected. Not all options are needed in every perspective. In some perspectives some options make sense and in others not.

On recording mode, if the main perspective is selected, the menu present the options to turn on/off the laser and select the laser type in first place. These are important options, and showing them first makes the user feel the same. Then, options to change the RWA, wavelength of the wave emitted by the laser source and the option to change the object are presented. The wavelength can vary between 0.1 and 3.5 (this value is the chosen maximum because above this number the space between

---

[4]`https://github.com/dataarts/dat.gui`

wavefronts start to be so large that when the animated laser is turned on there are big intervals of time where there are no wavefronts traveling on the scene). The RWA can vary between 1 and 89 degrees. To change the object the user choose one option from a select box with several geometric figures. After that, some generic options are given to the user, like the option to turn on/off the labels and the option to show the interference pattern right way without turning on the laser. This last option exists so the users can see the interference pattern without the need of waiting for the laser, or just don't want to use the laser at all, because they already know how it works or because of any other reason. The last options available are the options to change the perspective and to swap between recording and reconstruction mode. Figure 4.13 shows a screenshot of the interface for the main perspective.
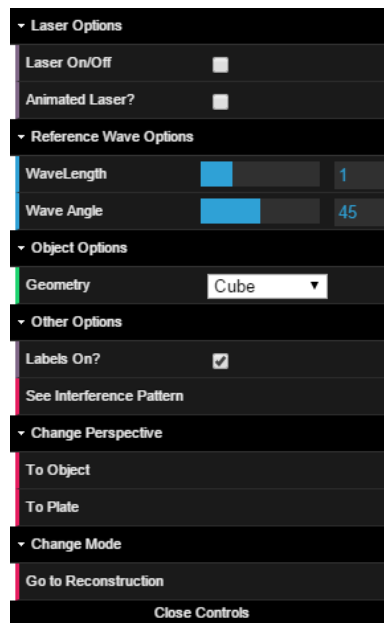
Figure 4.13: *Interface of the main perspective on recording mode*

If the object perspective is selected, the menu presents the options to turn the laser on/off and select the laser type in first place, just like in the main perspective, for the same reasons. Next, the core options of this perspective are presented. As the object perspective is selected, the options are centered on the object. So, the options available are to change the object and the detail of the object. Both these options are select boxes. In the case of changing the object, the select box contains several geometric figures, while in the case of changing the detail, the select box contains

several levels of detail. After that, the same generic options of the main perspective are presented here. And, as in the main perspective, the last options are the ones to change the perspective and mode. Figure 4.14 shows a screenshot of the interface for the object perspective.
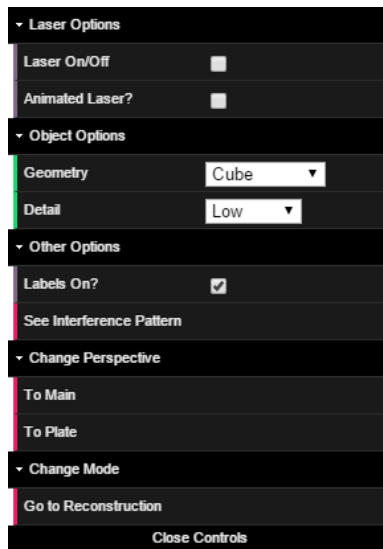

Figure 4.14: *Interface of the object perspective*

If the plate perspective is selected, the menu presents all the core options first. These options are all the core options of the other perspectives combined–that is, all the wave and object options. On this perspective there are no options for the laser because it is omitted. The option to show the interference pattern right away is also presented here, as are the options to change the perspective and mode. Figure 4.15 shows a screenshot of the interface for the object perspective.

On reconstruction mode, the main perspective is selected by default and it is the only perspective available. On this mode, the only options presented to the user are the option of to turn the laser on/off, select the laser type, turn the labels on/off and the option to go back to the recording mode. All the other options are hidden because the interference pattern is already generated and what is being shown is the reconstruction of that interference pattern. Figure 4.16 shows a screenshot of the interface for the reconstruction mode.
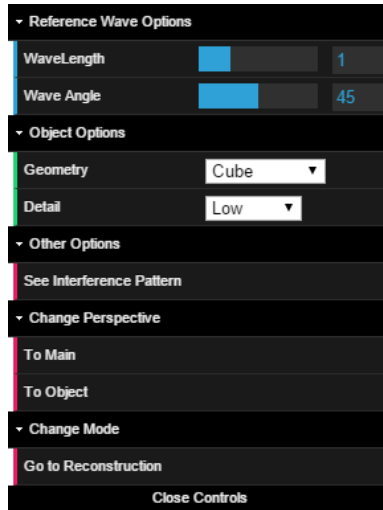
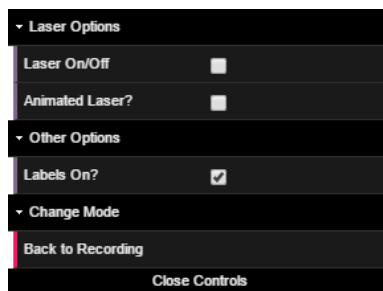Figure 4.15: *Interface of the plate perspective*

Figure 4.16: *Interface of the main perspective on reconstruction mode*

Chapter 5

User experience on the virtual lab

When an user enters, the laboratory is presented as a simple holography setup for the recording of interference patterns. The laboratory is initiated in the main perspective with the laser turned off and with all components labeled, as shown on Figure 5.1.
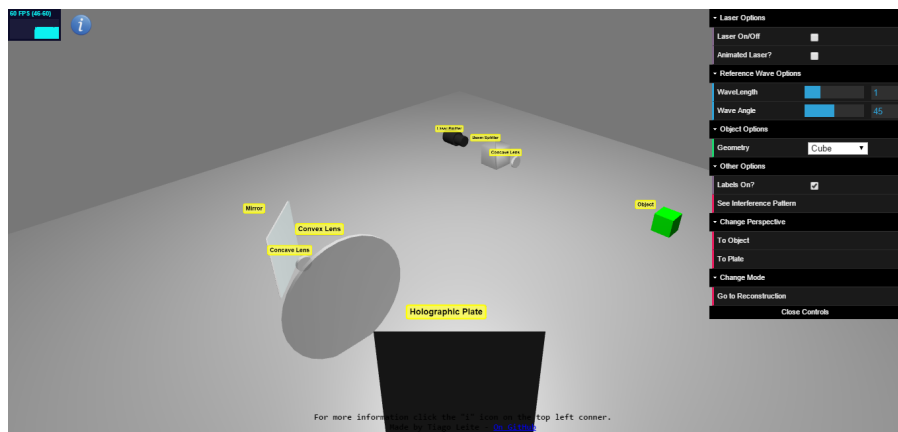

Figure 5.1: *Main perspective*

From here, the user can, for example, turn on the laser. By default the laser is turned on as *simple laser*, as shown on the Figure 5.2.

However, with a simple click, the animated laser can be activated, as shown on Figure 5.3.

While on the main perspective the user can also change parameters related to the
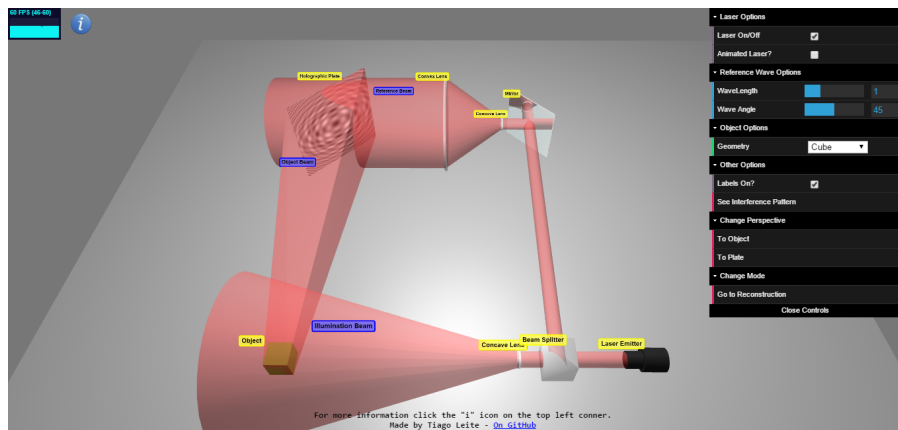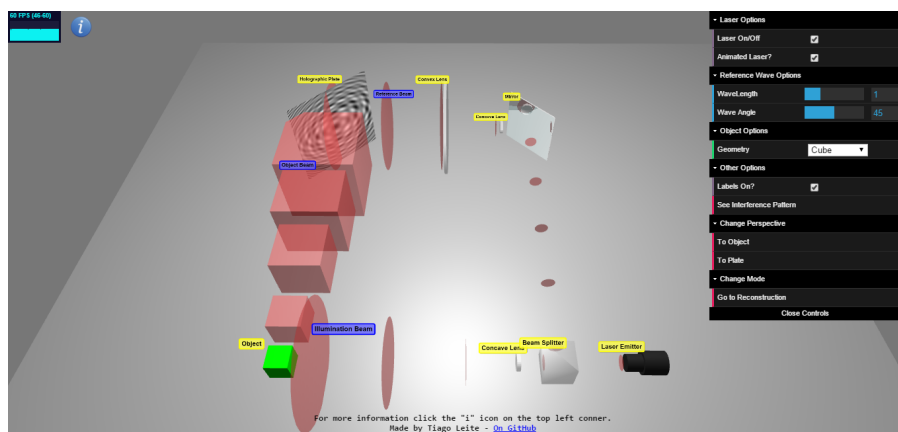
Figure 5.2: *Main perspective with simple laser*

Figure 5.3: *Main perspective with animated laser*

wave, namely, the wavelength and the reference wave angle. These changes influence the position of the mirror and lenses, and the spacing between wavefronts, as shown on Figure 5.4

Another option available to the user is the chance to change the object used to create the interference pattern. There are several options on a drop-down menu that can be chosen. Figure 5.5 shows the swapping of objects.

Other general options are available to the user. The user can see the interference pattern without turning the laser on if they want. Also, the labels can be turned off, as shown on the Figure 5.6.

If the user wants to take a closer look at the object and its light points, the object perspective is available, as shown on Figure 5.7.
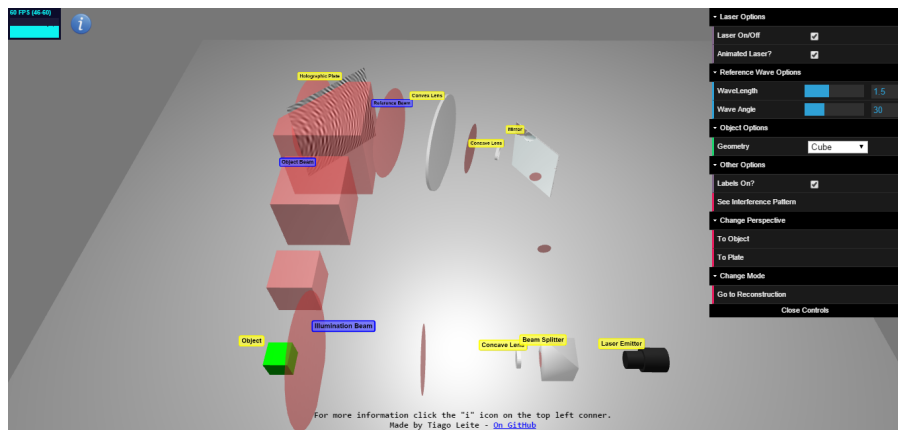
Figure 5.4: *Main perspective with animated laser and new wave parameters*
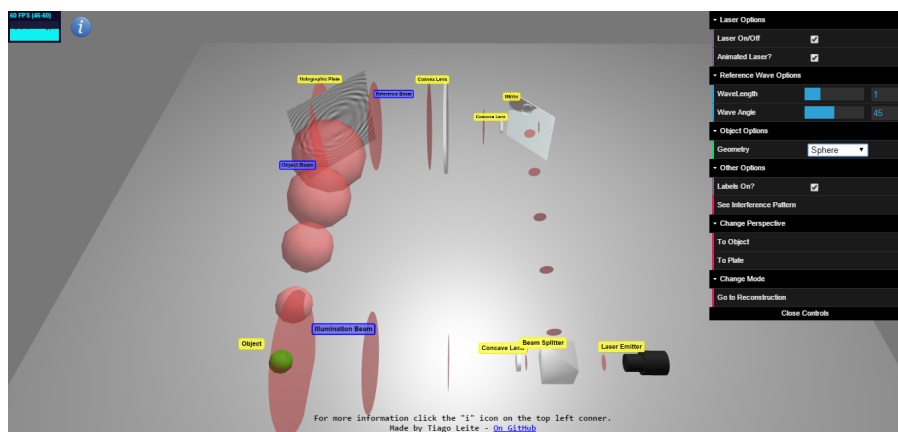

Figure 5.5: *Object changed to a sphere*

As on the main perspective, the user can turn on/off the laser. If the laser was turned on previously it will keep turned on, and will continue to use the laser type chosen before. Figures 5.8 and 5.9 show the two lasers types.

The options available on this perspective are the possibility of changing the object and changing the level of detail of the object. Figures 5.10 and 5.11 show these options.

The third perspective available is the plate perspective. This perspective focuses on the plate and on this perspective the camera is disabled. Figure 5.12 shows this perspective.

On this perspective the wavelength, reference wave angle, object, and level of detail of the object can be changed, as shown on Figure 5.13.
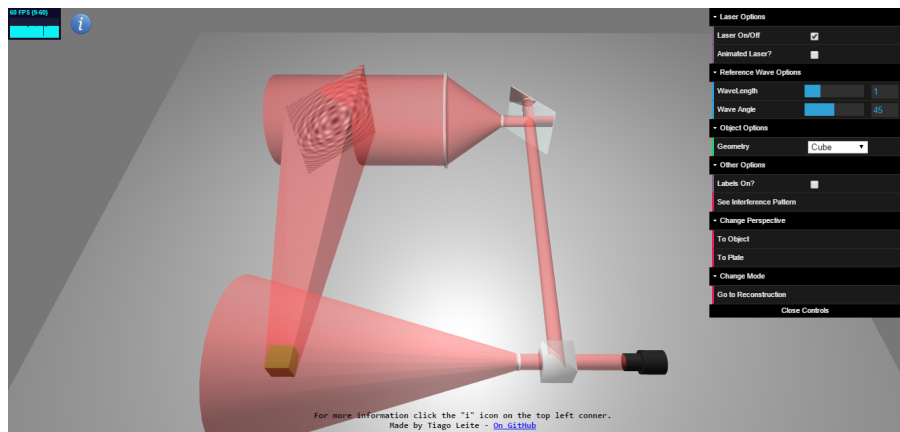
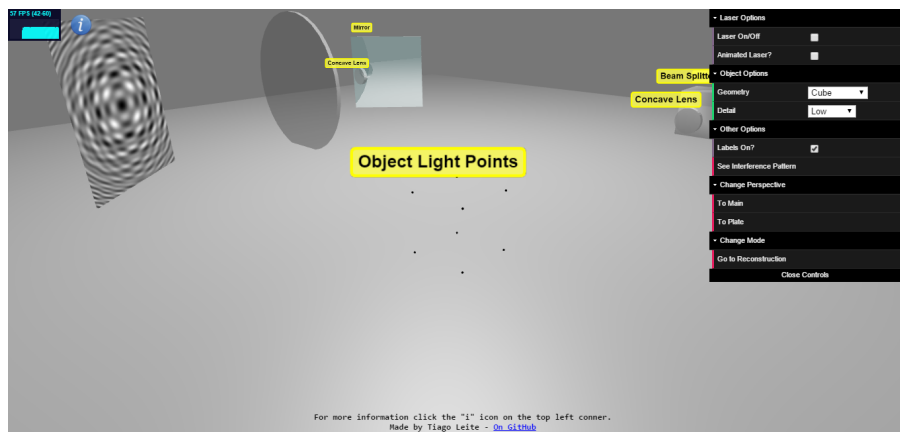Figure 5.6: *Main scene without labels*

Figure 5.7: *Object perspective*

When a interference pattern is generated it is possible to change the mode of the main scene to reconstruction mode. This mode only uses the main perspective and will change the scene in order to be possible to illustrate the reconstruction process of a interference pattern. Figure 5.14 shows this situation.

Here, the only options available to the user are the possibility of turn on/off the laser (simple or animated) and the possibility of turn on/off the labels. Turning on the laser reconstructs the object wave recorded on the interference pattern and makes a virtual object to appear where the real object was. Figures 5.15 and 5.16 show the two laser types being used.

From here, the user can go back to the recording mode to change the interference pattern, if they choose to.
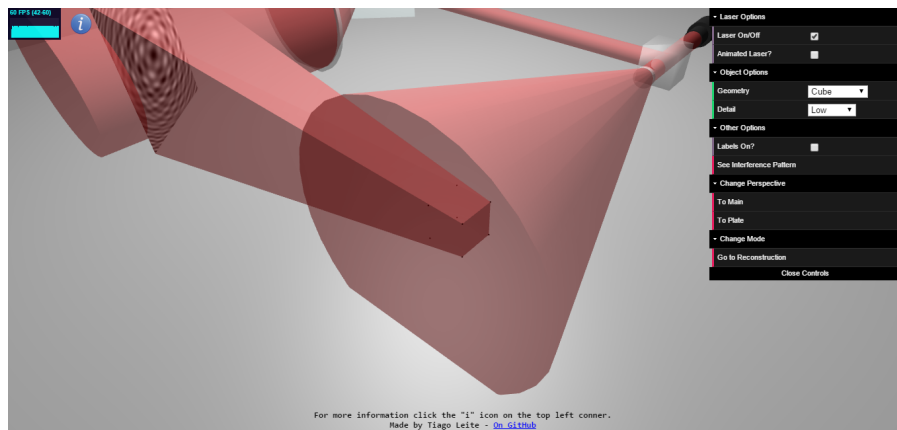
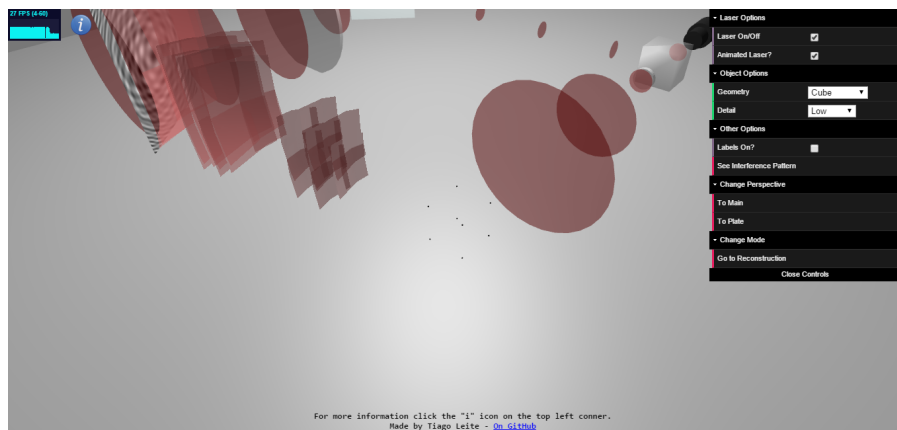Figure 5.8: *Object perspective with simple laser*

Figure 5.9: *Object perspective with animated laser*

In every perspective and mode the user has the possibility of seeing extra information about that perspective or mode on an information box by clicking on the *"i"* button. Figure 5.17 shows one of these information boxes.

The laboratory can be accessed at `http://tiagoleite07.github.io/cgh-lab`. The source code is available at `https://github.com/TiagoLeite07/cgh-lab`.
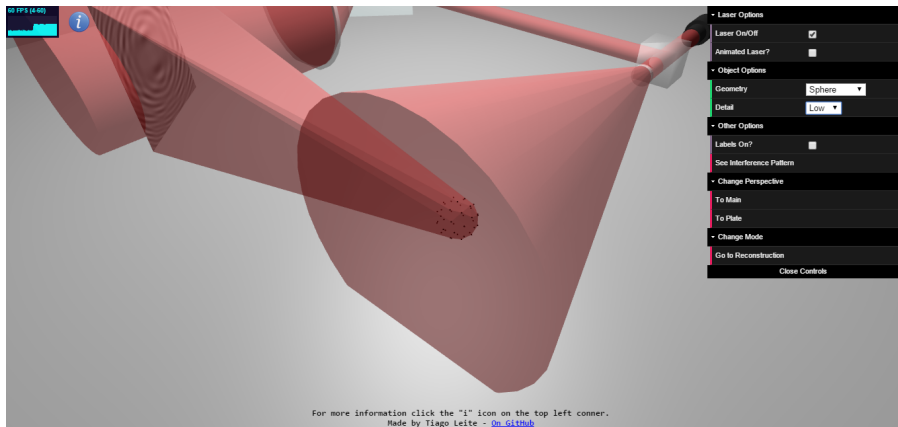
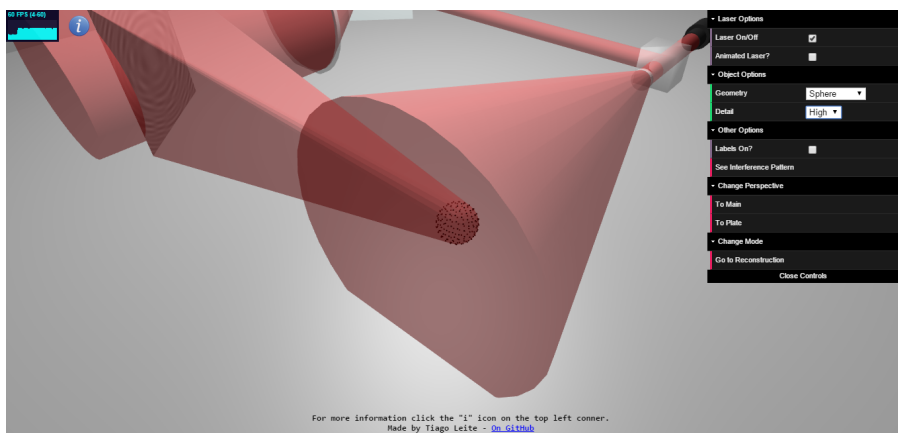Figure 5.10: *Change the object on the object perspective*

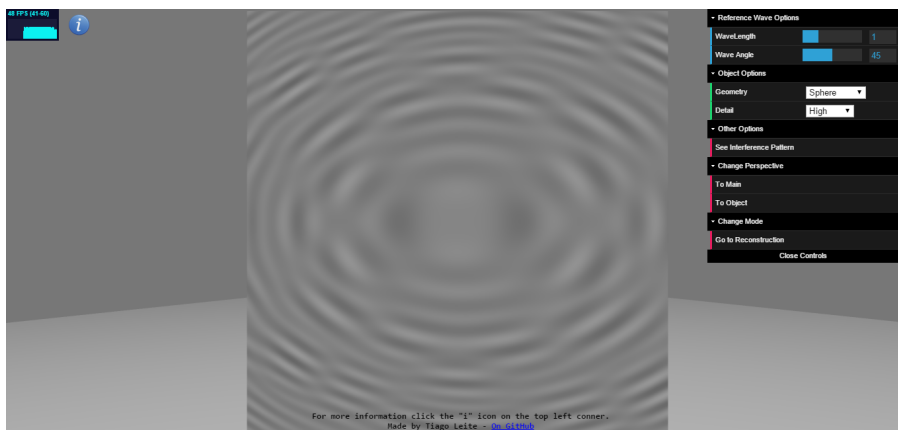Figure 5.11: *Change the level of detail of the object on the object perspective*
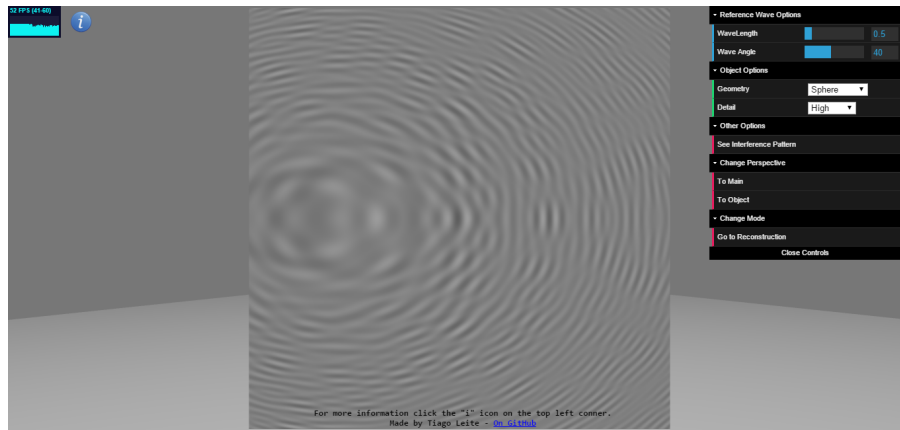

Figure 5.12: *Plate perspective*

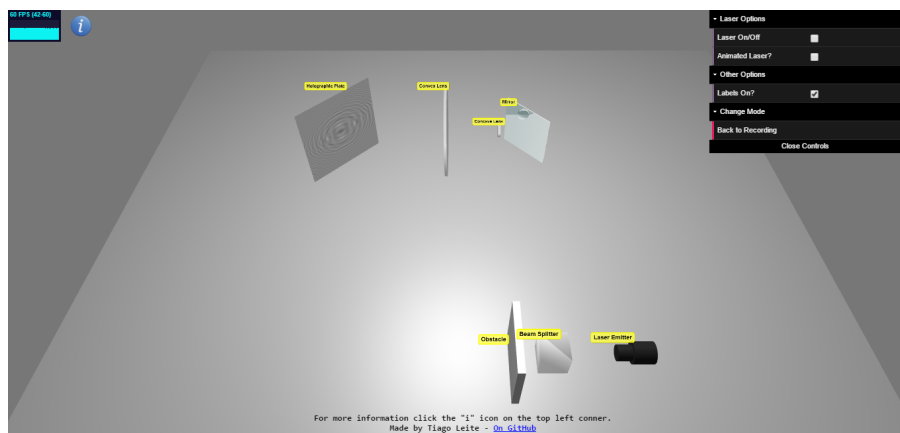Figure 5.13: *Plate perspective with parameters changed*

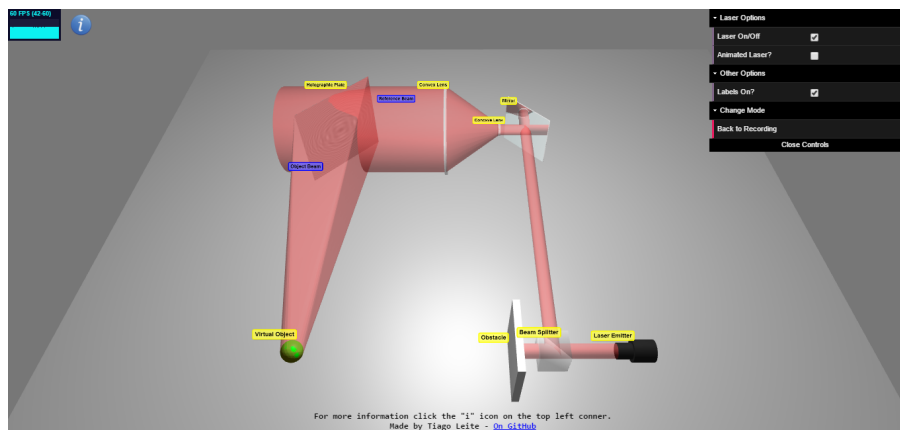Figure 5.14: *Reconstruction mode*

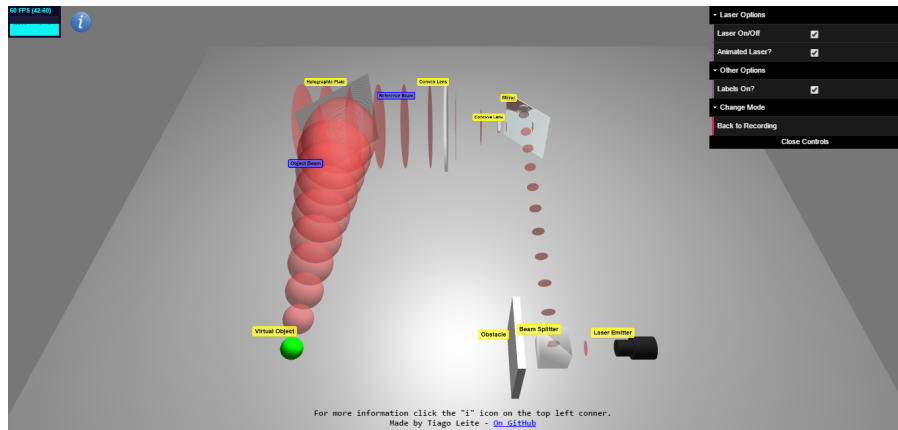
Figure 5.15: *Reconstruction mode with simple laser*

Figure 5.16: *Reconstruction mode with animated laser*

Figure 5.17: *Information box of the main perspective while in recording mode*

Chapter 6

Virtual lab evaluation

6.1 Limitations

The biggest limitation of the laboratory is the number of light points that can be used to create an interference pattern. The shader responsible for the generation of the interference pattern has a maximum number of variables (uniforms) that can be passed to it. This number can vary, depending on the browser, operating system and graphics card used. Thus, a number that would work on the maximum number of computers possible had to be found. After some tests with different numbers, the maximum number of light points chosen was 162.

The number of light points also provides another limitation. While on the object perspective with the animated laser turned on, every light point emits a wave. Therefore, there are potentially 162 waves being emitted at the same time. This has a great impact on the performance, as will be shown now.

The laboratory was tested on two browsers (Google Chrome v45 and Mozilla Firefox v40), in two different operative systems (Windows 10 and Arch Linux). Two graphics cards were used to test the laboratory: a NVIDIA GeForce GT 740M and an Intel HD Graphic 4600. The computer used was the same.

The number of Frames per Second (FPS) obtained for each combination, on the object perspective, using the maximum number of light points, and with the animated

laser turned on, is shown on Table 6.1.

GPU + OS	Google Chrome v45	Mozilla Firefox v40
NVIDIA + Windows 10	16 FPS	15 FPS
Intel + Windows 10	2 FPS	2 FPS
NVIDIA + Arch Linux	14 FPS	11 FPS
Intel + Arch Linux	2 FPS	2 FPS

Table 6.1: *Number of FPS obtained on the object perspective while the animated laser is being used, using the maximum number of light points possible*

On Windows, a better performance was obtained. One possible explanation for this is the existence of more optimized drivers for the graphics cards on Windows.

In general, the laboratory runs better on Google Chrome, both on Windows and Linux. While using Mozilla Firefox on Linux, the overall performance of the laboratory was worse than Chrome's, even with the laser turned off. Lower FPS values were obtained with Firefox. Also, it was detected that the laboratory can run very poorly while using Mozilla Firefox (on both operative systems), even using a more powerful GPU. While testing the laboratory on other computers, the results were very unstable with this browser, having some odd cases where the performance while using a NVIDIA GPU was far worst than using an Intel GPU.

With these results, the recommended way to run the laboratory is using Google Chrome and a dedicated graphics card, such as NVIDIA or ATI.

6.2 External evaluation and feedback

During the process of creation of the laboratory, the feedback of external sources was always important to guarantee that the user would have the best possible experience. The feedback of other students was taken into account in order to create a better user experience. The laboratory was presented to students with little or no knowledge of holography. Most of their feedback was towards the user interface being a bit confusing and some elements or events on the scene not being explained in the best

way. The menus were reworked a couple times in order to avoid confusion by the user and information boxes explaining each perspective were added in an attempt to eliminate the doubts referring to the events happening on each perspective.

Chapter 7

Conclusion

When someone is trying to understand how holography works in order to introduce himself to the field of computer generated holography, several obstacles can be found, like the use of the wave model of light transportation and all that comes with it. In this dissertation, a tool was developed to help new people that want to understand the basics of holography and learn a simple method of creation of CGHs that simulates the real world. By doing so, the hope is that new users can understand better the processes involved in the creation of holograms, and at the same time, users with more experience can also use the tool to quickly help them with their projects.

To achieve this dissertation results, the web-based laboratory was thought in a way that would show people how to generate CGHs and, at the same time, not move away from the traditional holography setup, as the method used is physically based. A scene that focuses on the most important components during the creation of holograms was developed to explain in a deeper level all the processes involved. The user experience was one of the main focus during the development of the laboratory. In order to new people understand what it is being presented to them, the scene was created using simple geometry, a simple interface was created and an easy control system of the scene is given to the user. During the development process, external feedback was always taken into account in order to improve the user experience. That way, the interface went through several phases until it reached its final form.

Some limitations for the tool were found. The number of light points used to create the interference patterns has proved to be a technological limitation for the shader that creates the interference pattern, because of the maximum number of variables (uniforms) allowed to be passed to the shader. The number of light points used is also a limitation in terms of performance due to the fact of the number of point light waves emitted being too large, while on the object perspective with the animated laser turned on.

The tool was developed using the javascript language and the WebGL framework *three.js*. This framework allowed a much quicker and easy development of the tool.

This tool is a contribution to those who are beginning their studies on the field of computer generated holography. When searching for this subject, tools like this are hard to find. It is hoped that this tool is able to give its users a new perspective regarding what is seen on websites and books.

7.1 Future Work

There are many things that can be improved or added to the laboratory.

- One of the things with room to improve is the reconstruction mode. At this moment, the reconstruction mode is no more than an illustration. There is no physical simulation here. In the future this mode can be reworked and an actual simulation can be implemented, just like in the case of the recording mode and the generation of the interference pattern.

- Right now, the tool only has one method to create the interference patterns, however, it is ready to easily receive new methods. In the future, new methods can be added to the tool, and by doing so, the users would be able to see the difference between them, and choose the one they think is the best for their situation.

- The addition of different setups, and the improvement of the geometry used are other examples of potential improvements. The user interface has also

space to be improved, according to the changes that the tool may suffer.

Bibliography

[AE07]   Gerhard K. Ackermann and Jurgen Eichler. *Holography: A pratical Approach.* Wiley-VCH, 2007.

[BBJ08] Stephen A. Benton and V. Michael Bove Jr. *Holographic Imaging.* Wiley-Interscience, New York, NY, USA, 2008.

[BL66]   B. R. Brown and A. W. Lohmann. Complex spatial filtering with binary masks. 5(6):967–969, 1966.

[BL69]   B. R. Brown and A. W. Lohmann. Computer generated binary holograms. 13:160–167, 1969.

[Bur70]   C. B. Burckhardt. A simplification of lee's method of generating holograms by computer. 9:1949–1951, 1970.

[dSS12] Pedro Miguel da Silva Santos. Holografia gerada por computador e reconstrução do sinal original: um estudo exploratório. Master's thesis, University of Minho, January 2012.

[Ein09]   Albert Einstein. The development of our views on the composition and essence of radiation. 1909.

[FS86]   Brill D.R. Falk, D.S. and D.G. Stork. *Seeing the light: optics in nature, photography, color, vision and holography.* Wiley, 1986.

[GM69] N. George and T. J. McCrickerd. Holography and stereoscopy: the holographic stereogram. 13:342–350, 1969.

[Har02]   P. Hariharan. *Basics of Holography.* Cambridge University Press, 2002.

[Klu]   Helen Klus. Newton's theory of light. `http://www.thestargarden.co.`
          `uk/NewtonAndLight.html`.

[Lee70]   W. H. Lee. Sampled fourier transform hologram generated by computer.
          9:639–643, 1970.

[Luc]   Mark Lucente.   *Diffraction-specific fringe computation for electro-*
          holography. PhD thesis, Cambridge, MA, USA.

[Luc92]   Mark Lucente. Optimization of hologram computation for real-time display.
          1667:32–43, 1992.

[Poo06]   Ting-Chung Poon. *Digital Holography and Three-Dimensional Dispaly:*
          Principals and Applications. Springer, 2006.

[SCS05]   Chris Slinger, Colin Cameron, and Maurice Stanley. Computer-generated
          holography as a generic display technology. 38:46–53, 2005.

[Wal66]   J. P. Walters. Holographic image synthesis utilizing theoretical methods.
          9:405–407, 1966.

[Wal68]   J. P. Walters. Three-dimensional fourier transform method for synthesizing
          binary holograms. 58:1284–1288, 1968.

[Yat76]   Toyohiko Yatagai. Stereoscopic approach to 3-d display using computer-
          generated holograms. 15:2722–2729, 1976.

[You02]   Thomas Young. The bakerian lecture: On the theory of light and colours.
          92:12–48, 1802.