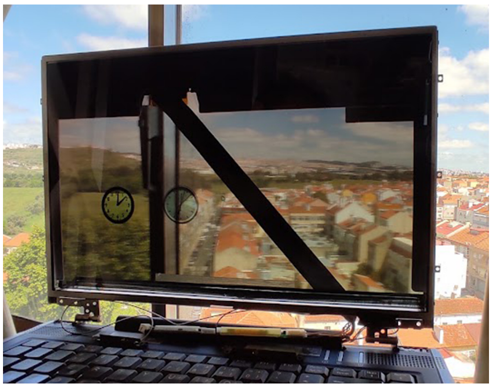


Studies for a Spatial See-Through Augmented Reality System to support Manufacturing Operations

Diogo Vicente Cleto dos Santos de Sousa Rodrigues

Thesis to obtain the Master of Science Degree in

Mechanical Engineering

Supervisors:	Prof. Mário António da Silva Neves Ramalho
	Prof. Virgínia Isabel Monteiro Nabais Infante

Examination Committee

Chairperson: Prof. Carlos Baptista Cardeira
Supervisor: Prof. Mário António da Silva Neves Ramalho
Member of the Committee: Prof. Susana Margarida da Silva Vieira

December 2021

Dedico aos que tanto me dedicaram.
Em especial, aos Especiais.
G

Acknowledgments

The project success relies on the support, motivation, and inspiration behind it. Hence, the names of those who were always present and whose dedication was an unbreakable constant must therefore be mentioned.

My sincere thanks go to my esteemed supervisors, Prof. Mário Ramalho and Prof. Virgínia Infante, for their constant encouragement, advice, attention, and availability.

I am grateful to the people I value the most, my parents and my brother, for their confidence, inspiration and comfort in knowing that I was never alone. I gained the strength and courage to overcome any challenge I faced because of their love for me.

Thanks to my friends, who were by my side whenever I needed it, especially Batalhadores, Clara, Inês, Lourenço, Miguel, Paulo, Rafael, Sofia and Solange.

I thank the others who accompanied me over the years and those who were part of my academic path.

Finally, I would like to thank the laptop grantees who made the practical and experimental development of the research work possible.

Resumo

O presente trabalho de investigação propõem-se a estabelecer a prova de conceito de um sistema de Realidade Aumentada suportado por um monitor transparente. O monitor responsabiliza-se por projetar, na posição correta, o conteúdo virtual gerado e modelado no computador, com o objectivo de se sobrepor ao ambiente real visível através do ecrã.

Com base em estratégias distintas, foram apresentadas duas abordagens que visam a alcançar os objectivos definidos para este trabalho. Cada estratégia é fundamentada pelas áreas de Visão Computacional, Percepção Monocular de Profundidade e Monoscopia 3D. Destas áreas, diversos métodos foram revistos, desenvolvidos e adaptados, especificamente atendendo o enquadramento do problema.

A sinergia dos três métodos em conjunto, em ambas as abordagens, viabiliza o facto de ser possível para o utilizador não ser limitado da sua percepção espacial natural enquanto interage com informação virtual, sem que seja necessário recorrer a um dispositivo adicional. Prevê-se, portanto, que a instalação do sistema tenha um impacto mínimo no processo e ambiente de trabalho do utilizador.

O sistema demonstrou a sua fiabilidade, pelo que os resultados apresentaram exatidão na localização dos elementos virtuais projetados, em função dos objectos reais atrás do monitor. Relativamente ao conforto e bem-estar do utilizador previu-se a possibilidade de surgirem alguns sintomas associados à utilização prolongada do sistema.

Palavras-chave: Realidade Aumentada, Transparência Espacial, Visão Computacional, Estimativa de Profundidade Monocular, Monoscopia 3D, Produção

Abstract

See-through devices are a particular type of Augmented Reality technology, developed and used with the aim of enhancing visual perception of the physical world with additive light.

The present research work was motivated by the objective of establishing the proof-of-concept of a proposed framework for a Spatial See-Through AR system supported by a transparent screen. The screen is responsible for displaying, in the correct position, virtual information generated by the computer, intended to enhance the world behind itself. The virtual content has the particularity of adapting and interacting with the user and the reality that surrounds it.

Two specific approaches were presented to achieve the defined objectives through different strategies. Each strategy relied on Computer vision, Monocular Depth Perception and 3D Monoscopy, from which different methods were reviewed and specifically developed and adapted to the problem statement.

The synergy of the three methods in both approaches was responsible for fulfilling the objective of allowing users to employ their natural spatial perception while interacting with digital information, without additionally resorting to any other device. Hence, installation of the system was expected to be minimally disruptive to users' work processes and shop floor layouts.

The system demonstrated its reliability by accurately displaying virtual elements relative to the position of the real object behind the screen. Regarding the users' comfort and well-being, some symptoms were pointed as potential issues after prolonged use.

Keywords: Augmented Reality, Spatial Optical See-Through, Computer Vision, Monocular Depth Estimation, 3D Monoscopy, Manufacturing.

Contents

List of Figures

Chapter 1

Introduction

The introduction will be divided into four distinct sections designed to provide the reader with a contextualization of the theme discussed throughout the work.

In particular, the Motivation section will expose the relevance of the subject and the reasons that have led to the development of the project under the terms it is presented. Then, in the Topic Overview, a brief review of the topic will be made in order to identify the current status of the topic studied. This section will be dedicated to summarizing the historical context, referring to basic concepts, technological development and evolution of areas of interest related to Augmented Reality. Content related to the methods and algorithms applied in the final implementation of the project will not be included in this chapter. The objectives set to be achieved with this thesis will be expressed in the third section. These will be complemented with a brief description of the reasons behind their establishment. Finally, for the Thesis Outline, the structure of the document will be briefly described and the contents developed in each chapter will be briefly explained.

1.1 Motivation

Industry is steadily moving towards smart factories to achieve effective communication between the physical and digital world. This concept has been expanding for the past few years and aims to develop and explore intelligent manufacturing as a promising solution capable of increasing autonomy and adaptability.

Augmented reality (AR) is an effective and innovative tool to support this vision. Rather than replacing humans this technology is considered the only one whose purpose is to improve the integration and interaction between humans and intelligent systems. Thus, AR is very promising for many different fields of application.

Implementing AR technology in manufacturing applications is a strong and growing area of research. As a result, it is globally used to provide simulations, assist manufacturing processes, and contribute to reducing lead times, lowering costs and enhancing quality. Even so, AR does not appear to be ready for industrial deployment in some areas, as there are still some issues to be addressed.

Within AR systems, optical see-through devices are becoming increasingly popular and widely used in enterprise environments for solving some of those challenges. Head-mounted displays (OST-HMDs) are one example of the most common ones. Nevertheless, spatial see-through systems, whose advantages stand out when comparing it with the majority of augmented reality devices, seem to be neglected in modern research works. Therefore, hereby emerges the relevance in exploring the potential of leveraging those advantages to deal with the challenges that arise in areas where AR hasn't yet reached its full potential.

The research work presented was driven by a perspective of making a contribution to the advancement of technology in the Augmented Reality field. Thereby, leading to the development of viable systems for future industrial applications.

Throughout this project, AR technology's evolution within the manufacturing industry, in recent years, ought to be taken into account. The aim was to propose solutions for the limitations that prevent this technology from being widely implemented as a support tool. Additionally, within the current and future context of Industry 4.0, there are needs associated with the development of intelligent systems intended to increase the autonomy and flexibility of processes. These needs must also be the focus of attention and are expected to be properly tackled based on the main problems highlighted in the most recent research works.

The prominence of technological advances in the manufacturing industry does not diminish the importance of human workers in this field. In this regard, user acceptance is crucial, as it may be responsible for hampering the technology's efficiency. Consequently, for this work, a user-focused approach will be taken in the development of the augmented reality system. As part of this approach, adversities identified as responsible for limiting the application of AR technologies in shop-floor operations and processes, were addressed.

1.2 Topic Overview

AR refers to a set of technologies aimed at enhancing our perception of the physical world by using computing devices. In 1968, computer graphics pioneer Ivan Sutherland designed the first AR prototype, which displayed 3D graphics using an optical see-through helmet [1].

See-through devices, nowadays, are still developed and used due to their particularly interesting functionality. Using these devices, visual perception is augmented using additive light. The image visualized is the result of combining projected imagery, light from environmental sources, and the surrounding background [2].

AR applications started becoming more rapidly developed only in the late 1990s due to the availability of free software prepared suited to this technology. During that period, the development of camera systems increased making it easier to analyze the environment and assess object positions in real-time.

At the time, the emergence of AR led to explore its potential in applications from a wide range of areas. Among them are medicine, maintenance/repair, annotation, robotics, entertainment, and military settings[3]. New fields of application were identified as promising, throughout the time, for AR implementations such as outdoor, mobile AR, collaborative AR, learning and education [4].

As a tool for supporting the development of products with added value from raw materials, AR solutions initially targeted the manufacturing industry. This technology served many purposes and provided a number of advantages in the sense that promising outcomes were obtained for multiple tasks. It was and still is employed in monitoring and controlling process flow, evaluating real-time plant layout, ensuring plant and machinery maintenance, and improving industrial safety [5].

Currently used AR systems have ergonomic issues that must be overcome. Although an array of options exists that feature hands-free operation, most of the AR devices are criticized for restricting head and neck movement, due to the weight of the helmets. Meanwhile, there are several reports that present that virtual information during operations causes workers to become disoriented and distracted, due to the limited field of view.

Problems related to processing power highlight the influence of the surrounding environment on the reliability of tracking objects of interest, mainly for tracking applications that use marker-less technology. As compared to marker-based systems, these solutions require substantially more computing resources, because there are still limitations to computer vision-based tracking methods related with their lack of robustness and stability.

Unprepared environments are still highly challenging for tracking, so evaluation of the impact of implementing AR systems is imperative. It may require significant organizational changes within the firm, which have a profound effect on work processes, information exchange processes, and even on individual operators.

1.3 Objectives

In light of the preceding sections, from this project, it was proposed a conceptual framework for an AR system capable of enabling the user to interact effectively with virtual information during manufacturing operations. Furthermore, it should achieve a sense of presence in the real world by allowing users to employ their natural spatial perception. Installation of the system is expected to be minimally disruptive to shop floor layouts and users' work processes.

Explicitly, the project aims to establish the concept's proof-of-concept by studying the development of an Spatial See-Through system grounded by methods from several subjects and supported by a transparent screen. These methods include: Computer vision, Monocular Depth Perception and 3D Monoscopy. The screen will be responsible for displaying, in the correct position, information, images or any other type of virtual scene generated by the computer, intended to enhance the world behind the screen. The virtual content will have the particularity of adapting and interacting with the reality that surrounds it. In other words, any user movements or changes in space must be interpreted and processed by the computer so that the image on the screen remains updated and adjusted to the panorama of each moment.

1.4 Thesis Outline

The document is divided into five distinct chapters.

Chapter 1 refers to this Introduction to the project.

Chapter 2 - Literature Review, is intended to expose the complete literature review that was carried out in the context of the project. The reviewed articles focus on the historical and technological evolution of Augmented Reality associated with the Manufacturing Industry. In this chapter, the themes considered essential to contextualize and guide the research work will be briefly addressed according to the general panorama in which this technology fits in the current and future of Industry. Additionally, the usual methods for evaluating the performance of AR systems, the problems highlighted by the other authors and the acceptance of the implementation of AR solutions in the shop-floor will be reviewed. All topics reviewed will influence the decisions and approaches described in the next chapters.

Chapter 3 - Methods, begins with a framing of the problem. This first section is intended to elucidate and illustrate the context and objectives of the project. This section will be the starting point for the development of the SST system, as the three main areas of the project will be stipulated. Namely, Computer Vision, Depth Estimation and Screen Output. Each area is associated with a set of methods and algorithms that will be presented and developed. These methods will be subject to comments and tests in order to predict which ones are suitable for future implementation in the final prototype.

Chapter 4 - Implementation and Results, it dedicates to explain the ways in which the chosen methods and algorithms were implemented during the development of the prototype that aims to achieve the proof of concept of the conceived project. The chapter is divided into three sections, the first two of which refer to two distinct approaches to project development, based on the methods presented in the previous Chapter 3. In these two sections the procedures carried out will be described and the decisions taken will be explained. Finally, the last section focuses on a detailed analysis of the results obtained with the developed prototype. The results that are presented come from theoretical evaluations, which aim to predict performance problems associated with errors and deviations in the presentation of elements on the screen, and from observations and interactions with the system that allow relevant conclusions to be drawn.

Chapter 5 - Conclusion, presents the objectives achieved with the development of this project, in relation to what was initially stipulated to achieve in the Introduction chapter. From the results obtained, there will be room to discuss the areas of application of the developed technology, given its advantages and problems that were highlighted in Chapter 4. At the end of this chapter there will be some comments regarding future ideas for work that can be developed within the subject of this project.

Chapter 2

Literature Review

Throughout this chapter it will be briefly reviewed the evolution of world's Manufacturing Industry through the multiple changes in engineering. In fact, this preliminary review will be crucial to understand the appearance of Augmented Reality (AR) for the first time in scientific researches, as well as the needs that triggered its development and later applications in real engineering and industrial projects.

During this state of the art, the main point is to study efficiently and meticulously Augmented Reality. The primordial concepts and theory established, the basic components of its definition, the technology development and evolution and the fields of application in projects are considered the most important subjects, as they are going to be reported more carefully in this chapter.

Attention will also be given to the methodology and metrics used during research works to test and evaluate AR systems' performance before and during applications. Consequently, interest arises around the issues and challenges in AR applications and the general acceptance of AR, from a company and user point of view. Finally, a brief discussion on statements related to the future directions and trends within the AR technology, based on several research papers and scientific opinions, will close this chapter.

2.1 Industrial Revolutions and Industry 4.0

The evolution allowed developments in terms of techniques that led to major changes in production processes.Having pioneered steam power and mechanized production during the first industrial revolution, the manufacturing industry is one evidence that innovation on an ongoing basis is crucial, as this industry is undergoing through changes until today [6, 7]. After shifting away from artisanal production and craft, about 2 centuries later, assembly lines and electricity were introduced to factories during the second industrial revolution. Mass production was, then, able to be achieved by specializing employees in manual labor. Advanced electronics and information technology only came into play in the 1970s, enhancing automation and improving coordination in manufacturing processes, thus triggering a third industrial revolution, the "digital revolution." [8] However, at this stage, despite being centrally connected and automated, the production systems were still inflexible. This issue leads us to the present initiatives to incorporate intelligence in the manufacturing industry.

Today, Industry 4.0 envisages integrating digital technologies into manufacturing environments to enable cyber-physical production systems (CPPS) [9], capable of decentralizing decision making and increasing data generation and processing. This way, as manufacturing's intelligence develops, we can enhance its adaptive capability, autonomy and flexibility, therefore contributing to the fourth industrial revolution [10, 11].

2.2 AR Primordial Concepts and History

Virtual reality inventions date back to 1832, but Ivan Sutherland, years later, stood out as the great pioneer mainly because in contrast to existing systems, he stressed that "the user of such an ultimate display should be able to interact with the virtual environment" [12]. All of his work and research led him to develop, right before the third industrial revolution, the very first Augmented Reality see-through prototypes, around 1960[13]. More precisely, he was capable of building the first functioning Head-Mounted Display (HMD) using half-silvered mirrors as optical combiners, allowing the user to see both the computer-generated images and objects in the room, simultaneously [12]. During the 1970s and 1980s, researchings around the "ultimate display", as it was known at the time, kept going managed by the U.S. Air Force's Armstrong Laboratory, the NASA Ames Research Center, the Massachusetts Institute of Technology, and the University of North Carolina at Chapel Hill [14]. Nevertheless, it was only in 1992, when two scientists at the Boeing Corporation, Caudell and Mizell, developed an experimental AR system to help workers put together wiring harnesses, that the term "augmented reality" was conceived by Caudell and Mizell [6, 15, 16].

After Caudell and Mizell's definition, many other authors try to caracterize AR and compare it with Virtual Reality. The opinions diverged, as some agreed to classify AR as separate from VR, and, therfore, define them as two completly different tecnhologies, because rather than immersing a person into a completely synthetic world, AR attempts to embed synthetic supplements into the real environment [17] .On the other hand, some believed there was a relationship between AR and VR [18]. Milgram and Kishino, agreed that AR and VR could be associated with each other and that this argument should be considered valid from a scientific point of view. Hence, decided it was convenient to view them as opposite ends of a Mixed Reality (MR) environment continuum [19].

Nonetheless, the differentiation between AR and AV is not distinct along the continuum [6]. Clarifying, as long as the real content is dominant, it is AR. In another matter, at the spectrum's midpoint, ideally this would be relative to an environment where real and virtual are undistinguishable.

Nowadays, augmented reality is defined in quite distinct ways, however briefly we can describe it as a technology which combines interactively, in real-time, real world environment elements with computer-based scenes, images and potentially all other kind of senses to deliver a unified but enhanced perception of the world [20, 3, 14]. Yet, some researchers relate to AR only when the content is displayed in 3D [3].

Concerning true mobile AR, eventhough mobile devices had started to come into play around the early 80s, computing and tracking hardware wasn't sufficiently powerful and small enough to support graphical overlay. Hence, the first prototype of a mobile AR system (MARS), was only achievable in 1997. Feiner et al. allowed the device to register 3D graphical tour guide information with buildings and artefacts [21].

Since the late-1990s, AR has been a specific field of research, as demonstrated by the fact that several conferences have been held with focus on this technology, such as the International Workshop and Symposium on Augmented Reality, the International Symposium on Mixed Reality, and the Designing Augmented Reality Environments workshop [6]. In the meantime, this emerging technology became possible to rapidly develop thanks to freely available softwares and researchs [14]. For that reason, it is classified by the European Union as one of the main technologies that will drive the development of smart factories [20].

2.3 Technology Basic Requirements and Development

AR has come successfully prominent very fast in the last decades [15], as previously stated. Hence, hardware and software that support AR applications are both evolving into complex systems, despite the fact that these kind of tools are, increasingly, more widely available [22].

Each particular application requires different components to implement the adequate AR solution, as it depends on many parameters. Those parameters may focus on the place, on the complexity of the virtual scene, on the applied [23]. However, it is consensual within a big part of the authors that the minimum and basic requirements of any standard AR system includes four major components: the visualization system, the capturing system, the interaction system, and the tracking system [24, 25, 26, 27, 28].

During the following section it will be explored the technology developments that allowed the integration of AR solutions outside the lab environment. As it is possible to infer, this review will be presented divided in the essential components of AR defined previously. This way a deep understanding of the existing devices and systems used in AR applications can be achieved, as the information is logically structured and organized. Nevertheless, all the four components are heavily connected and for this reason it will be impossible to discuss each one of them without mentioning the others.

It is important to mention that this section will focus on hardware technology developments. This decision was made based on the vast and extensive variety of commercial software that support AR. However, additionally to the four subsections, there will be space to briefly review AR software systems on a fifth subsection.

2.3.1 Capturing

Capturing technologies refers to any system or device responsible for acquiring visual evidence of a surrounding scene or physical environment. It should be able to allow analysis in real or deferred time so it can be posteriorly displayed to the user with overlaying information.

Within this definition, the main and most common technology is the camera or the alternative camera connected to a monitor, like laptops or even smartphones. As a matter of fact, the term "camera" nowadays includes an endless list of diverse and advanced devices and technology developments. However, for this review the only relevant systems covered by the previous description should be able to capture and perceive reality as trustworthy as possible, with minimum or no filter, distortion or any other kind of modification. AR systems nowadays rely on this type of technology as a way of integrating virtual objects with reality. Devices such thermal cameras, for example, should not be considered as, by it else, may be already seen as a complete AR device.

Although Optical See-Through (OST) devices are reviewed in the following section, related to visualization systems, is justifiable to mention it because OST tech for AR may not demand any kind of capturing component besides it own transparent or semi-transparent lens/screen, which allows the user to visualize directly the external and evolving environment, with no intermediary. In short, the particularity of having a see-through component implies that the user's eyes are the only "technology" responsible for perceiving the scene. Thus, it is predictable for OST systems that the integration of a robust tracking system is a must, in order to acquire useful data about the place and objects around the user. Tracking may include sensors or even any kind of camera, as will be discussed next.

2.3.2 Visualization

Visualization systems are key in an Augmented Reality application. Through these, the digital content is displayed to augment the environment. Scenes may also be presented as the result of image processing [6, 18].

To successfully view an augmented reality demo, yet is possible to choose from three alternatives. The first, video see-through, refers to an approach very similar to virtual reality, whereby the real environment is indirectly visualized in the form of a video captured live by a camera. The video is overlaid with images, text or any other type of visual information, in real-time. The second goes back to the beginnings of augmented reality as it derives from prototypes developed by Sutherland. Known as optical see-through, it leads to perception of the real world overlayed with AR renderings by means of transparent mirrors and lenses.The third strategy involves projecting the overlay the AR directly onto the real objects themselves [14].

Video See-Through

According to the brief description above related to Video-See-Through (VST) technology, through a camera that frames the user's surroundings, or at least part of it, the captured image is replicated with virtual models superimposed on it, in a video streaming device [23, 3].

VST's, even though they are well suited to virtual reality applications, appear in the industry as good augmented reality solutions in different contexts as well. Usually, the most common cases that fit for the implementation of VST AR require hardware, such as portable devices, which include one or more cameras and a screen whose interaction with could be possible. So it's only natural that smartphone apps are recurring examples of VST. Thus, emerges the interest in quickly exploring the area of Mobile Augmented Reality, framed in the world of VST's.

Mobile AR (MAR) is one of the fastest-growing segments of augmented reality, mostly because of the widespread use of mobile hardware, like smartphones and tablets, as it is already known [29]. According to some authors, these devices rely on specific requirements, such as computational framework, wireless networking, and data storage and access technology [4, 30, 14]. Besides those mentioned, registration, interaction and tracking are also indespensable to implement in industry applications [31, 15].

Optical See-Through

OST can make use of a set of technologies developed specifically for supporting AR needs, especially the technologies related to projection techniques that generate great interest in industry and between researchers and developers [21]. The simplest solution is also one of the first, and was previously refered as Sutherland's prototype. It projects images using the half-mirror technology, nonetheless it suffers from a great disadvantage related to the limited field of view, that can be overcome by increasing the complexity and using free-form shaped mirrors.

Newer and more reliable technologies lay on waveguide grating. This holographic and diffractive optics based solution blends the light into a thin layer of glass. By replicating the phenomenon that happens in a lens , the light is reflected and redirected towards the user's eye [23].

From among the several OST hardware developed and implemented in industry applications, a description of some relevant ones will take place in this section, starting with the most dominant of all the Head-Mounted Devices(HMD) [6].

The OST HMD is a wearable device, similiar to some kind of helmet, which, due to its transparency, the user is able to see the environment through it complemented with aligned virtual computer-generated information [22]. Some of the reasons why HMD OST are so successful and are slightly increasing with time, rely on the fact that it is easy to transport and does not require cameras or monitors to be installed in the testing area [15]. Interestingly, technologies that merge Eye-tracking and OSTHMD systems are being developed to keep up with the evolving interest this area [32].

Head-up displays (HUDs) come around as one less known and used alternative to HMD's, and despite being a technology under development, there are already examples of applications that make it a promising solution in the future, such as the implementation of spatial optical see-through adaptions in military cockpits or projections of navigational directions in commercial car's windshield [33, 14].

Spatial see-through (SST) displays work as a fusion of two technologies in this section, namely, OST and Spatial AR. Instead of projecting directly onto the real objects, in SST the projections or the rendered images supposed to augment the surroundings are displayed on a transparent surface/screen, which is placed between the user and the scene. However this concept may generate some distortion or displacement issues due to the user's movement in relation to the “invisible interface” [34] and perspective. Yet, 3D holographs are capable of minimizing or even solving this alignment problem [35].

Static screens which are one of the most common displays in SST are usually used as prototypes for the proof of concept before further developing and implementing HMD systems, or even HHDs [6].

Spatial Augmented Reality

The Projective displays or better known as Spatial Augmented Reality devices [12] are currently used to enhance the accuracy in industry processes [36] and also in the area of entertainment and culture for shows and exhibitions, for example.

These have the great advantage of using only one or more projectors to produce the augmented reality effect, so there is no need for any type of monitor or device that must be held or worn by the user, such as Head-mounted/Hand-held devices [6]. This solution also stands out for being quite suitable for applications involving large-area surfaces, regardless of their color and whether these are flat or complex [37]. Contrary to what was stated in some older researches, this technology is not limited to indoor spaces [14], so it can be used outside in very low brightness conditions, usually at night, so that way it is possible to guarantee a good contrast in the projection of images. The greatest regard to be taken with SAR's is calibration, especially when the environment or projection surface changes and moves.

2.3.3 Interaction

Interaction devices are the most important key in the linkage between the user and the AR technology. The usability of these devices has a huge role in the AR application performance and acceptance, [38, 39] because the interaction systems are responsible for reading and understanding the user's input and consequently influence the information processing and displaying [15].

There is a wide range of ways that can be implemented to allow interaction with AR technology. The possibilities are countless and new developments have been coming up frequently in the last years, [40, 41, 42, 43, 44] however, some are used more frequently than others. From the most standard interaction devices, like mechanical keyboards, mouses, remotes and controllers, technology evolved to new complex and sophisticated alternatives within the Tangible User Interface (TUI) devices.

Formerly known as Graspable User Interface (GUI) devices, in which users interact with digital information through the physical environment, these can be divided into kinaesthetic (force, motion) and tactile (tact, touch). TUI's, among the multiple existing equipments, include the currently most available worldwide, those defined with bidirectional and programmable communication through touch, the haptic devices [42, 45, 46]. These refer to any handheld or mobile device technologies that rely on tactile feedback, like smartphones, tablets, and so on [14].

Still, following up TUI's, data gloves appeared to revolutionize the interaction technology. Instead of using our hands to hold up devices that allow us to control and communicate with the system, data gloves, turn it possible to do it through hand gestures, wearing only a pair of gloves. Nowadays, it is considered a very reliable, flexible and widely used in VR for gesture recognition, yet for AR applications, it may not be as suitable, for its usability imply that our hands are impeded to be used directly with real world tasks, which can be a determinative factor.

Eventhough researches on gaze tracking (tracking of the eye movement) remote to the decade of 1950's, this technology is only starting to become available to implement nowadays, as it emerges as an alternative to allow the user to have their hands completely free, even while interacting with the AR system [47]. This solution provides the answer to modern user interests and needs, making it very promising to future contexts related to HMI [14]. Interestingly, gaze tracking is heavily related to the next section.

Given the disadvantage presented in relation to hand-worn devices, it is opportune to present technologies that allow interaction using the hands, without being equipped or occupied, thus allowing them to be free to carry out other activities [48]. This concept can be easily accomplished through the recognition of gestures captured by a camera [49]. A considerable percentage of this kind of applications are based on equipment that requires to be "worn" by the user, such as head-worn or collar -mounted camera pointed at the user's hands and wrist-mounted cameras [50].

On the other hand, there are also proposals for 3D environments manipulation that assume user input through gesture and touch recognition in relation to a specific device in the possession of the user [51, 52]. In this example the touch is used to select a virtual object, whereas the hand's movements result in the manipulation of the selected object. By using these solutions, virtual objects can be interacted with, in a natural and intuitive manner [15] providing new opportunities for future researches, developments and potential applications

It is very important to realize that an interaction between the user and the hardware that supports the augmented reality application is influenced by the software used. Keeping faithful to what was mentioned at the beginning of the chapter, a brief review on AR software systems will take place in this section to discuss subjects like interaction and others equally relevant.

2.3.4 Tracking

The tracking technology job in an AR device is crucial, because it is responsible for reading the scene, previously captured by the capturing component, and correctly identify and recognize the key features [6]. When this task is done accurately, the system is able to to provide properly the digital content when augmenting the scene itself [15]. Other essential requirement in tracking technology's job is to locate the user's position inside the environment. Considering that the big majority of AR applications work in real-time, the importance of this technology became prominent due to the need of take the minimal time possible to synchronize and display the virtual and the real worlds [22].

Within the tracking system there are two possible classifications to caracterize it based on its tracking method, specifically, on the usage of marker.

The first one, the mainly used from among industrial applications is the Marker-Based Tracking. The usage of this method has been referenced in older AR researchs and it has increased over the years [15]. Marker-Based Tracking can be easily defined as the name is quite intuitive. The system is configured to recognize specific objects and assets physicaly tagged, in the environment, with unique markers, such as barcodes/QR codes, fiducial markers, optical markers, physical markers or any other that fits within the application.

The second one is the Markerless Tracking, which is a more modern solution that can be implemented in several ways to track the elements of interest [53]. The main applications may rely on methods with feature or area-based tracking), [54] and even in some it may consist in merging the two [55]. Another approach is to use natural markers, however it's noteworthy that usually this may imply the usage of optical tracking. Hereupon, in order to work, it requires to resort on physical pictograms or objects, available on-site, that stand out for their color or shape [56, 57, 58].

Beside the two main types of tracking approaches discussed above, there are many tracking systems available to be applied. The most common will be individual presented and described:

Mechanical Tracker

Most mechanical trackers, as the name may suggest, rely on mechanical component, such as telescopic booms/arms and/or string pulleys. Using this kind of tracker, user's position can be determined by forward or inverse kinematics. These trackers are usually used for Virtual Reality applications.

Inertia Tracker

A typical inertia tracking system features a rigid body equipped with gyroscopes and accelerometers oriented according to an orthonormal 3D referencial. The tracking process determines the object's orientation and angular displacements given the angular velocities, determined with the help of the rate gyroscope through integral's calculation.

Acoustic Tracker

Acoustic tracker due to its compact-size and low-cost is a technology that despite being a great attraction to develop in modern researches, it has tested by Sutherland [13], since the origins of AR itself. It may be also known by ultrasonic tracking (or positioning) and is defined by propagate ultrasonic waves/pulses to, then, use the echo scheme and multiple sensors to triangulate the 3D position of the specific object [59].

Electromagnetic Tracker

Electromagnetic trackers [60] measure distances and estimate positions determining the intensity and direction of the electromagnetic fields. It also benefits from infrared, visible, and radio waves identically. Interestingly, carrying out with electromagnetic trackers may be done in two distinct ways.

On one hand, the so-called outside-looking-in method requires the sensors installed all around the environment so they can be able to track the emitters placed on the user.

On the other hand, the inside-looking-out method contrasts entirely with the previous one and the reason for that is because the sensors are on the user.

These two are both quite used, but the first is applied used for motion capture in the entertainment industry while the second is more requested in mobile augmented reality and machine vision in robotics [22].

Optical Tracker

Optical Tracker are very common in marker-less tracking application, although it is a completely viable method for marker-based tracking, too. It relies on computer vision supported by usually more than one camera (only one camera is also possible to implement) responsible for capturing the scenes, just like explained above in previous sections. The images of the environment are analysed so the targets get successfully detected and its position and orientation can be estimated. Some of the commercial systems available use infrared optical camera(s) to track objects over optical tracking.

Hybrid Systems

This systems are very useful as they, usually, can rely on the advantages from each technology merged [61]. One good example of hybrid system is the motion tracking that most often derive pose estimates from electrical measurements of mechanical,inertial, acoustic, magnetic, optical, and radio frequency sensors, which grant this complex tracking system robustness in tracking all six degrees of freedom, very fast and accurately at a low cost [62, 63, 61].

2.4 AR and Industry 4.0

During the following sections several implemented models in modern engineering applications will be presented, not only to to review AR applications, but also to give a reasoned statement about the versatility of this technology and the vast fields it currently covers, and promises to cover in a near future. However, it is important to realize that despite its vast employment range, AR's main objective is to provide a rich audiovisual experience. [20], and this aspect lays on the basis of this technology, contributing to its popularity within the manufacturing industry, and others.

When Augmented Reality was first defined by Caudell and Mizell, it was also immediately conceived as a supporting technology meant for workers [3]. In particular, the first application example presented by the authors refers to the use of a first HMD prototype capable of dynamically marking the position of a hole [16]. Since then, AR has been aiming to come out with new and leading technological solutions that contribute to the development of intelligent manufacturing systems and factories.

The emergence of AR innovations and its success led to numerous and diverse proposals , during the last decade, to adopt augmented reality in an Industry 4.0 vision as a training tool suited fro improving workers' perceptions of their interactions with the environment [64, 65, 66, 67]. Today, right beside big data analytics, Internet of Things, additive manufacturing, smart sensors, machine networking and self-monitoring, AR is treated as a major pillar for the fourth industrial revolution.

With the increasing of product's complexity and number of variants, the manufacturing industry is becoming much more challenging than it ever was. Thus, the interest in developing and studying both Industry 4.0 and AR is also rising, leading many researchers to see AR a promising solutions for those challenges [68, 69, 70]. Curiously, although the variety of technologies already mentioned above play a huge role in the fourth industrial revolution, the only one focused on improving the synergy between humans and machines is AR[6, 20]. This fact shouldn't be surprising, because augmented reality has been proven as to be capable of decreasing the learning curve and being an effective tool to help correcting and preventing mistakes, among many other advantages [71, 72].

Despite of what was said, it has been stated by researchers AR and Industry 4.0 aren't yet being studied together as much as it would be expected, but instead they are increasingly being developed separately by the majority of the academics [23].

2.5 AR in Manufacturing

The manufacturing industry, described in a very synthetic way, is responsible for developing products with added value from raw materials. It is a complex process that can be defined by the different phases that constitute it, namely, design, prototyping, production, assembly, maintenance, among others [15].

The integration of AR in manufacturing comes up as an evolution from Virtual Reality applications. Before augmented reality emerged, VR was already being implemented very successfully in several manufacturing tasks, over product lifecycle stages, by completely immersing users in faithfull simulations and virtual environments [73, 74]. Using this method, the effort focused on accurately rendering real world simulations intended to reproduce real-life existing environments. However, as processes and machining work-cells in factories get more complex and difficult to virtual represent, a deal-breaker limitation on VR arises. Virtual reality technologies are very expensive to recreate demanding physical and dynamic behaviours, because it requires much time and effort to remodel real objects and working spaces, which are useful and crucial to the application itself [22].

To contrast, it is not necessary to model the real world using AR technology, since AR aims to enhance rather than replace the real environment like VR. This distinctive feature is considered very important, mainly because of the applications when modeling and maintaining the real world is too complex or when it depends upon a high level of accuracy.

However, there are more justifications that support the acceptance of the integration of this technology in the industry

As previously mentioned, there is an increasing need to develop this area in order to respond to the demand at a global level, for this, manufactures require technologies that allow them exchanging data at real-time along the product lifecycle, keeping in mind that humans are still a critical component in manufacturing operations [75, 76, 77]. Hereupon, augmented reality appears to try to solve the problem. AR applications are no way near to be implemented to replace human labor. Instead, the main goal with its employment in manufacturing is, precisely, to help workers adapt to the technology-focused approach the world is currently following, by providing users with information from virtual objects that can support them while completing almost any product-related task [3, 78, 22].

2.6 Testing Methodologies and Metrics

After laboratory researches are finished and studies had achieved the proof-of-concept of an innovative AR technology, naturally, progresses towards the development of a scalable solution and its implementation in the real industrial context are expected afterward. In order to accomplish it, prototypes and final products must be assessed by a series of testing and evaluations [5].

Different methodologies and metrics are used to conduct this kind of test, and even though each author proceeds with his own unique experiment, specifically suited for his product, there are ways to categorize and analyze this subject in a systematic manner, according to researchers [15, 6].

Guiding through the structuralization established by Egger and Masood [6], there are four categories. The Laboratory experiment, which includes tests conducted within a laboratory setting; The Field experiment, for studies in factory environments or experiments in which technicians and professionals participate in simulating shop-floor tasks in laboratory environments; Simulations, for completely digitally modeled tasks, using VR [79, 80]. Finally, Pilot project, which refers to prototypes at early stages that aren't extensively tested yet [81, 82, 83, 84].

Nevertheless, the above categories by themselves do not provide a full perception of studies that aim to assess AR technology solutions, as each one of them may be either focused on either two aspects. More specifically, researchers can carry out experiments to show how the application works and evaluate the solution's effectiveness. In this case, we may entitle it as a technical study [15]. On the other hand, when experiments lead to understanding the benefits generated when AR systems are implemented as a supporting tool for users, in comparison with traditional methods, these are designated as user studies [15].

Despite the fact, most cases focus on the improvement of a particular task through augmented reality, each study has a different purpose. Thus, emerges the need to stipulate metrics that aim to quantify and subsequently evaluate the performance ease of usage or the possibility of reducing errors of the AR prototype in question. Since there is an indefinite number of metrics possible to use during the testing stage, in this section, the attention will converge to only three. Namely, Time, Error rate [85] and NASA Task Load Index (TLX) [38]. This decision was based on the fact that these metrics may be considered the most prominent measures used among a wide range of researches. Interestingly, these metrics are commonly applied simultaneously throughout the same study.

Examples of other less frequent, but still quite used, metrics rely on user surveys [85], marker decoding distance, marker decoding time [56], and head movement [79].

> "The Official NASA Task Load Index (TLX) is a subjective workload assessment tool which allows users to perform subjective workload asessments on operator(s) working with various human-machine interface systems."
> Definition from [86]

2.7 Challenges and Acceptance

The majority of the challenges were still regarded as technological issues, despite the maturing of AR systems. The processing speed of the hardware has improved over the years, no and when using marker tracking technology is no longer presented as an issue. However, when using marker-less technology, the in-built processing power is still an issue.

Ergonomic design plays a role in user acceptance of an AR system. There are AR devices that stand out due to the inconveniences associated with their weight and the restriction of the field of vision and movement they provide to the user while using them, namely HMDs. However, with MAR solutions, these problems are no longer evident, so they are more easily accepted by users. Ergonomics can also be evaluated at the interface level. One of the most recurrent problems related to user comfort, after prolonged use of the equipment, refers to the symptoms of cybersickness, such as nausea, visual fatigue, among others. At the interface level, it is also possible to point out that, depending on the type of design of visual cues present on the monitor or on the equipment's lenses, the acceptance and impact on the performance of AR technology may be different.

Following the previous paragraph, regarding the interface, there is a need to assess the influence of the software. With the advent of digital manufacturing systems, certain standards must be followed by the modeling interfaces and data structures, so that intelligent manufacturing systems can be implemented. However, there are still no dominant standards that allow for a completely intuitive use of the software. Likewise, there are still no standards for the process of user interaction with the system, especially for scenarios where interaction is done through gestures.

As for acceptance by users, even privacy and its protection are among the most sensitive issues raised by trial users during surveys [87, 88, 89]. The performance-critical component of this system is the ability to track indoor location, tasks, and errors. With that capability, superiors can more easily monitor users [57, 84].

Regarding the acceptance and performance within organizations, the identified challenges by authors in their research are mainly related to the user's perspective, especially in laboratory settings where it is difficult to identify challenges that are specific to the company. However it was established by some authors that not all tasks can be supported by AR solutions. An increasing level of task complexity was suggested to be associated with the effectiveness of AR [90, 88, 91]. Therefore, it is predictable that the level of acceptance will vary depending on the company's activity sector.

2.8 Future Directions

Multiple directions for the technological development of Augmented Reality systems are foreseen. Among them:

With regard to fields of application, certain sectors are identified that arouse interest in developing and implementing AR solutions. These sectors are identified as emerging or trending within the research work. These include: Training and Learning, Maintenance and Remote Maintenance, Facility Layouts, Ergonomics and Safety, and Production Management.

The implementation of an AR system by a company has associated costs. These costs refer not only to the installation but mainly to the development and design of the systems themselves in order to adapt them to the work environment. Therefore, progress is foreseen at the level of economic assessment of the costs and savings generated by the AR implementation. A possible approach is to consider the associated costs as an investment, but evaluating such outcomes will require a large number of AR applications over the coming years

For future research work it is required that the interest of the manufacturing industry towards the use of AR solutions be continuously evaluated. Through empirical analyses, targeting manufacturing companies that have already implemented AR solutions, it will be possible to guide technological developments with a view to increasing the performance and acceptance of this type of systems in a context and work environment. Currently, interest seems to be directed towards the development of intelligent manufacturing applications and equipment that aim to mitigate problems associated with information processing. There is also a strong tendency to develop user-centric applications in order to promote greater acceptance of the technology.

Chapter 3

Methods

The following sections are intended to present and discuss the methods studied, developed and used over this project. The methods exposed are, for the most part, the result of a literature review carried out in order to assess the most appropriate solutions for each area of the project. Hence, each method will be subject to analysis and comparison between the others used and revised, in the context of the specific area in which they fit.

As mentioned in the previous paragraph, the project was divided into distinct areas. For this chapter, three stand out and can be perfectly integrated within the structure described throughout the literature review chapter. These will be presented in the following order: Computer Vision, Depth Perception and Screen Output, and are associated, respectively, with three of the four basic components of an augmented reality system: Tracking, Interaction and Visualization.

Firstly, each method will be followed with a theoretical component that aims to clarify the concept, the approach, and the final objective intended with its implementation. Then a mathematical and/or logical development will reflect the entire process as well as the algorithm needed to support the chosen methodology. It should be noted that the entire discussion will be, in due course, grounded and commented on according to the literature review carried out. This review focused on authors who developed and applied identical methodologies or different ones, which had the same final objective as those presented. Therefore, there will still be room for the necessary observations regarding the implications associated with the execution of the method in question. Observations may target advantages, disadvantages, comparisons or any other type of comment that may be considered relevant.

Regarding the fourth base element of an AR system, Capturing, as explained in the appropriate chapter, technologies dedicated to image capture refer to systems responsible for acquiring visual information from the surrounding physical environment. In other words, it is related to hardware, which is not included in this chapter intended for the study and development of methods. Thus, topics whose content is based on the implementation of this type of system will only be addressed later, in due scope.

3.1 Problem Framing

Before starting the analysis of methodologies, it is crucial to understand the problem framing and get to establish the type of methods necessary for the implementation phase, so it can be possible to obtain a complete, cohesive and functional system.

The project aims to study the development of a SST system, supported by a transparent screen. The screen will be responsible for displaying, in the correct position, information, images or any other type of virtual scene generated by the computer, intended to enhance the world behind the screen. The virtual content will have the particularity of adapting and interacting with the reality that surrounds it. In other words, any user movements or changes in space must be interpreted and processed by the computer so that the image on the screen remains updated and adjusted to the panorama of each moment.

Once the final expected achievement to be reached with the development of the present research work is defined, it is then necessary to stipulate the requirements of a functional SST system, as outlined previously. For this, the organization of the basic and essential elements in AR projects is used, as discussed in the literature review of Chapter 2. To summarize, most AR systems must include the following four components: Capture, Interaction, Location and Visualization.

By implementing the four components in this SST system, it is immediately possible to infer that the capture area refers to the equipment that will be used to capture images and information from the user and the surroundings. Since the emphasis of this chapter is on analytical methods, and the Capture component, of this specific project, doesn't resort to any, the discussion of this topic will be postponed to the next chapter aimed at describing the process of implementing and testing the SST system. In light of the above, the only observation to note refers to the fact that throughout the next sections of this chapter it will be assumed that image capture will be carried out in real-time by a conventional webcam that is fixed on the plane of the transparent screen. Further details about the webcam itself, the screen, and their positions relative to each other will be included in the next chapter.

Regarding Interaction, it is important to establish that in this SST system there won't be any additional devices that allow user-computer communication, such as helmets or hand-held devices. The interaction between the user and the image displayed on the screen will, as previously mentioned, be done only through the user's own movement. Thus, the identification and recognition of specific movements that aim to transmit some type of information or input are also discarded. In particular, the objective will be to locate the user's head in real-time so that it is possible to estimate the user's point of view in relation to the transparent screen. Therefore, having knowledge of the aforementioned parameter, as well as the panorama of the surrounding environment, more specifically of what is behind the screen, it will be possible to adapt the screen image so that it is always adjusted to the user's vision. Consequently, with the correct implementation of this SST system concept, it will be possible to realize the augmented reality effect without compromising or restricting the person from performing a certain task, not requiring any type of training to use the technology. Aside from that, the environment is also free from interference, i.e., it is not subject to external disruption. The Depth Estimation section will develop methods related to this component.

Despite their distinctness, the components presented do not correspond to independent subsystems of the AR system. As a matter of fact, they are closely related and in constant communication. The preceding paragraphs already provided a brief overview of Tracking. A section dedicated to Computer Vision will develop methods for this component. As mentioned, in this project, visual information from the surrounding space and the user will be captured by a webcam. From the acquired images, it will be necessary to estimate the position of key objects and user's head, so that the perception of reality is enhanced from the perspective of the person interacting with the system. The Tracking subsystem will be responsible for analyzing in real-time each frame, providing the Visualization subsystem with the coordinates for each object and for the user's head.

From the Visualization component, it will be required to process the information received and adapt, to the panorama of each moment, all the content that is displayed on the transparent screen. Therefore, it may be possible to ensure that all the virtual images displayed to the user will coincide perfectly with the environment behind the screen. For this purpose, aspects related to variation in perspective and perception associated with key objects, both from the user's and camera's point of view, need to be addressed in the definition and subsequent implementation of the methods to be discussed.The methods for this component will be developed in the Screen Output section.

The order of discussion of each area will be logically guided as the development of the SST system progresses. During the entire process, in order to facilitate understanding of the reasoning that is intended to be transposed throughout the report, it will be taken into account an illustrative context. This fictitious scenario (Figure 3.1) will also guide the progress of this project:

> Consider only the following elements inside a closed space: a person, a red ball and a transparent screen with a conventional webcam incorporated. The three elements are properly spaced apart and positioned in a straight line, with the transparent screen placed between the ball and the person.
>
> The person's field of vision is intersected by the screen, but the particularity of its transparency allows the ball to be visible through the screen itself, as if it were a glass window.

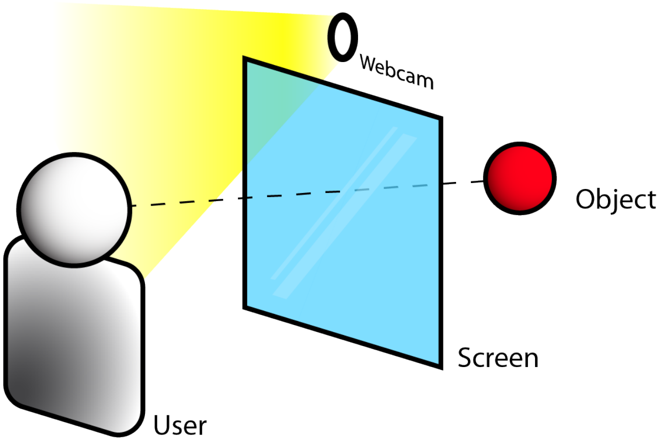


Figure 3.1: Representative fictitious scenario.

During the discussion of the methods, for purposes of simplifying the reasoning, it will be considered that the webcam integrated into the screen is turned on and facing only the user so that it is possible to capture in real time all his/her movements and displacements within the closed space. It is also established that the person may leave the initial alignment, no longer being in a straight line with the other two elements. However the screen will never cease to be arranged in such a way as to intersect his/her field of vision whenever he/she looks at the ball. Finally, the ball's position in relation to the screen is fixed and known in advance.

3.2 Computer Vision

To start with, the camera is turned on and therefore it is recording the person live. Keeping this general parameter in mind, the objective of methods to be developed in this area dedicated to Computer Vision is straightforward: Identify the user's head in real time. More specifically, in the two-dimensional coordinates of each frame captured by the webcam, the method should be able to define the area that limits the head.

However, the focus of this project is on determining the coordinates of the person's point of view. Later on, these coordinates will allow adapting the screen image to a prediction of what the user can see through the transparent screen. This, therefore, means that not only is the head area sufficient, but it will then be necessary to determine the location of the eyes. Hence, a problem arises that must be addressed. In contrast to HMD devices where each eye is forced to see an image different from the other, in this project, as the screen is properly away from the user, it makes no sense to follow the same stereo image approach, so a single image is displayed. However, a person usually has two functional eyes, which translates into two different points of view.

This fact can be easily verified if we try to look, individually with each eye, at two objects that are at different distances. For example, if you hold a pen in front of your face, with your arm outstretched, and observe its position relative to a cabinet at the back of the room, you will see that the pen appears to be at different positions depending on the eye that is closed or covered. This difference in perception between the two points of view in relation to a fixed scenario proves to be quite important.

The above observation concludes that it is necessary to develop a possible method of allowing the screen to project an accurate and faithful image. This image will be defined according to what the user is expected to be able to see from his current position. For this, it will be necessary to determine the coordinates of a point that can be used as a point of view equivalent to the result of the conjugation of two simultaneous points of view of a person, namely the two eyes.

Throughout the project, by approximation, the coordinates of the equivalent point of view to be considered will be defined by the midpoint of the straight line segment delimited by the center of each user's eye (Figure 3.2). Alternatively, the coordinates of just one of the eyes could have been chosen, however, this decision would imply that the screen was immediately designed to be adjusted only when the user covers or closes the other eye.

The next subsections will present two different methods that were duly tested during their develop-

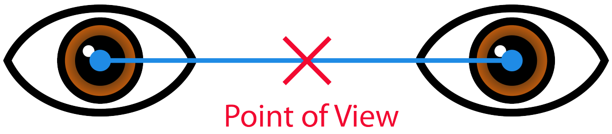


Figure 3.2: Point of View representation.

ment in order to verify their feasibility for a future implementation together with the methods in other areas. The first is distinguished by applying image arithmetic as a means of obtaining the silhouette of the user's head. On the other hand, the second method is based on using face and eye detection algorithms in order to directly extract the coordinates of the area of the image that corresponds to the person's face and eyes.

3.2.1 Image Subtraction

The first method presented portrays an approach that uses arithmetic applied to images. Image arithmetic is known for being the basic technique to detect moving objects. It is defined as any type of image processing technique that involves basic arithmetic operations, such as addition, subtraction, multiplication or division. In this context, the method to be applied is restricted to the subtraction of images. Therefore, it may be possible to distinguish, within every frame captured by the webcam, the background from the objects of interest, namely the user. Ideally, the silhouette of the person can be extracted regardless of the scenery behind them.

Many authors dedicate to the development of this technique to detect and extract, in real time, moving objects through dynamic masks [92, 93, 94, 95, 96]. Typically, within the research works reviewed, two approaches are used to achieve this end. More specifically, background scenario subtraction and temporal differentiation. Respectively, the first approach involves subtracting a frame that portrays only the background from original frames that capture the presence of the object of interest. In the second approach, more suitable for applications where the background is not constant, the subtraction is done between consecutive frames throughout the captured video.

However, regardless of the approach, the underlying concept of this method is always the same. Nevertheless, in order for it to be understood, it is first necessary to carry out a brief theoretical introduction.

Colorimetry and image processing were areas of research from an early age that were dedicated to developing methods and codes that would enable the numerical representation of colors [97]. The application of this type of codes becomes pertinent in several areas, as its existence allows an easy communication and manipulation of information and data related to the colors of an image.

Color can be codified in different ways, being the most common the RGB. In the next paragraphs the focus will be directed to the RGB code since it is the color model used in digital content by most monitors and computer graphics systems. In short, this model is a color additive process based on three primary colors. From weighted combinations of the three colors that give the code its name (R-red, G-green, B-blue) it is possible to generate a wide variety of colors. Each generated color is defined by the sum of different intensities of red, green and blue.

Generally, the RGB model is represented by the independent coordinates from, a unit cube in an ortho-normal referential defined by having the three primary colors distributed with maximum intensity on the vertices that intersect the positive semi-axes of the referential (Figure 3.3). The black color corresponds to the origin of the referential and white is located at the opposite vertex of the cube's spatial diagonal [98].

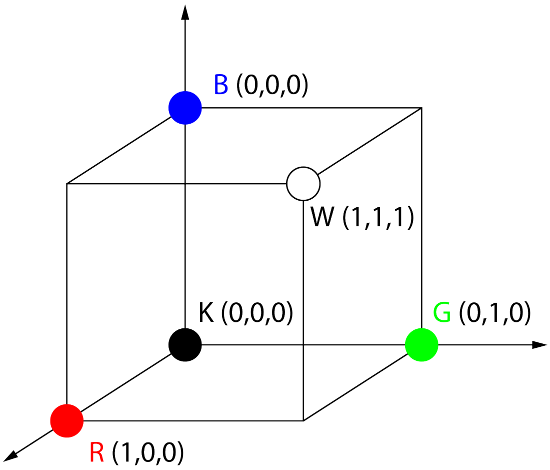


Figure 3.3: RGB three-dimensional cube model.

With the basic theoretical concepts well defined, it's now possible to resume the discussion of the image subtraction method. As explained each pixel of a color image is defined, in RGB, by a three-dimensional vector. Analogously to the graphic representation of RGB, each coordinate of the vector corresponds respectively to the intensity value, from 0 to 1, of the colors red, green and blue. Therefore, a color image of H x W pixels resolution will be easily represented by a three-dimensional matrix of dimensions W x H x 3.

In order to simplify image processing, it is required that initially the figures are converted from color to a gray scale, where the intensity varies from 0 (black) to 1 (white). One way to process this color scale conversion is through the following weighted sum [99]:

$$I(x, y) = 0.299R(x, y) + 0.587G(x, y) + 0.114B(x, y) \tag{3.1}$$

Thus, even if some image information is lost, it is possible to convert an initial matrix W x H x 3 to an W x H x 1 matrix. This way the method will be much less demanding at a computational level, facilitating its implementation with response in real time.

Proceeding with the method, consider that the subtraction will be made between image A and image B. Both images have the same resolution, H x W pixels, have already been converted to a grayscale scale and can be defined in a matrix as follows:

$$\{\forall x \quad \forall y \quad 0 \leq A(x, y) \leq 1 : x \in [1, W], \quad y \in [1, H]\} \tag{3.2}$$

$$\{\forall x \ \ \forall y \quad 0 \leq B(x,y) \leq 1 : x \in [1,W], \ \ y \in [1,H]\} \tag{3.3}$$

The subtraction of images will be processed, therefore, in a very intuitive way, since it corresponds exactly to the arithmetic operation between the two matrices that define each one of the images. In this case to subtract image B from A, we obtain the following matrix C:

$$C = \left| \begin{pmatrix} A(1,1) & \cdots & A(1,H) \\ \vdots & \ddots & \vdots \\ A(W,1) & \cdots & A(W,H) \end{pmatrix} - \begin{pmatrix} B(1,1) & \cdots & B(1,H) \\ \vdots & \ddots & \vdots \\ B(W,1) & \cdots & B(W,H) \end{pmatrix} \right| \tag{3.4}$$

It is possible, given everything that has already been mentioned, to analyze matrix C, in a context of subtraction of a background. Considering that matrix B translates a background and matrix A represents an image of an object on the same background, we can easily infer that at least all pixels in image A that match the background will be represented in matrix C by null elements. In other words, a pixel of the same color at position (h_1, w_1) in images A and B will imply an element with the same value at position (w_1, h_1) in matrices A and B. Therefore, at position (w_1, h_1) of matrix C, we will obtain a null element which will correspond to a black pixel in the final image. Therefore, matrix C will be null in all elements, except those that, in matrix A, represent pixels belonging to the object of interest. Exceptionally, there may be null elements in matrix C that correspond to the object's pixels. This will be verified if any of those pixels in A have, by coincidence, the same color as the scene pixel, in the respective position, in B.

The matrix C thus represents the final image. However, in order to highlight the silhouette of the object of interest, it is necessary to establish that all non-null elements of matrix C must be represented as white pixels. In this way, ideally it will be possible to obtain an image with only two colors, black and white, where the entire white area will correspond to the outline and due filling of the object of interest. On the other hand, the black color region refers to the background.

$$C(x,y) = \begin{cases} 1, & if \quad C(x,y) > 0 \\ 0, & otherwise \end{cases} \tag{3.5}$$

The image subtraction process itself can be ended. However, in image C, a perfect selection of the area corresponding to the object is rarely immediately obtained, without the final image being subjected to additional treatment beforehand. Therefore, as several authors point out, it is essential to proceed with certain adjustments to the results obtained in order to increase their quality and, consequently, improve the performance of the method's implementation [92, 93]. The aforementioned adjustments that need to be applied are characterised by varying a lot depending on the type of images used. However, the reason for its application is generally due to common problems in image arithmetic methods. Thus, different tests were carried out as a way of surveying the problems that were evident and the subsequent adjustments required.

The tests were carried out according to the two possible approaches indicated above, with the support of *MATLAB* [99]. Both had common aspects to be corrected, the most obvious being evidence of noise in the final image, that is, instead of obtaining a well-defined white region, the image had a white main area, but deformed and surrounded by others silhouettes of the same color, only comparatively residual in terms of size.

Given this description, several measures were implemented in order to obtain the best possible final result. Of which the following stand out:

- Adapt the defined threshold value to impose the white color on the pixels of the C image. As described, all non-null elements would be considered ”white”, but with this change, a margin of error is established for all elements whose intensity value is sufficiently close to 0. All those that are non-null, but that fall within this margin of error, will be considered “black”;

$$C(x,y) = \begin{cases} 1, & if \quad C(x,y) > T_d \\ 0, & otherwise \end{cases} \tag{3.6}$$

  where, T_d is a difference threshold value.

- As mentioned before, by coincidence, it is possible that pixels that belong to the object are shown in black. With the implementation of the previous measure, this effect will become even more evident, generating small black areas within the white region. Therefore, it is necessary to impose the white color on all black areas that are completely surrounded by white pixels;

- A detail to note refers to the fact that the white area is not well delimited on the margins due to the existence of small white areas generated by noise. Therefore, it is necessary to force the black color for all white regions that are connected to other regions of the same color, by pixels that do not establish contact in both vertical and horizontal directions simultaneously. This measurement will be applied only to white regions whose area is less than a certain number of pixels;

- Finally, it is crucial that in the final image the number of white regions shown is equal to the number of objects that are intended to be identified. Thus, it will be imperative to eliminate all independent white regions that have an area smaller than the largest white region in the image. This measure obviously assumes that after all the adjustments made, the larger or larger regions (in case more than one object is to be highlighted) do not correspond to any interference in the image nor have they been caused by noise.

The problems described in the previous paragraphs are common to other implementations, so the measures presented were implemented under the influence of the suggestions pointed out by the respective authors, and also by their own strategies in an attempt to develop new image subtraction algorithms. More specifically, according to [96] the threshold value was characterised as a very important parameter for noise mitigation, but difficult to stipulate its optimal value. No algorithm was presented to predict the optimal threshold value, so the author suggests going by trial and error until obtaining satisfactory results. On the other hand, [96, 92, 93] refer to factors that must be taken into account in order to achieve quality final results. Of these factors, the following stand out:

- The attention required to variations in luminosity in the space where the tests are being carried out. A case with two images of the same space captured with the same camera under different lighting conditions is enough for the perception of colors to be different. Consequently, during image processing, it is extremely difficult to obtain a reliable selection of objects of interest;

- The care to be taken with shadows, since the presence of shadows in the image can be interpreted as the presence of an important object, or it can even deform the selection of the main object itself;

- Detection of transparent or overly reflective objects (mirrors, for example) may be difficult because they do not have a color that distinguishes them from the surrounding scenery. However, the situation of using transparent objects did not arise, nor will it arise as a problem in this context since the "object" of interest is the user's head. The transparent screen that will be used in the project will never be captured since the webcam is located in the plane of the screen itself, oriented perpendicularly to it;

- Finally, and probably one of the most important aspects mentioned in most research works, the fact that the method's performance depends on the quality of the background used. This factor was, without a doubt, one of the factors that most affected the tests performed. It was possible to verify that the use of a white and smooth wall is preferable due to its peculiarity of being completely distinctive from the user's head, in this context. On the other hand, backgrounds with very strong textures and different colors often disturb the final head selection, overlapping even when covered by the person.

The use of a camera with higher resolution can be an appropriate suggestion to improve the quality of the results, however it won't be enough to eradicate all the problems mentioned above.

In general, this method proved to be faithful in detecting the user for conditions where the background was fixed and was simple in terms of texture and color, contrasting with the person in front. On the other hand, under the same conditions using the second approach, where subtraction is done between consecutive frames along the captured video, an additional problem emerged. The fact that the subtraction is done between consecutive frames implies that there has to be constant movement in the captured video for an object to be identified. What was found, in the context of the tests carried out, was that when the user was still or his movement was not detected by the camera, the entire image was considered the background and therefore the person's head was no longer identified. This is understandable since during several frames in a row the image remained the same and therefore the subtraction will result in a null matrix during those moments. That said, this approach is excluded for future implementation in the project.

Nevertheless, the first approach to this method, in the modes in which it was presented, is quite unreliable in conditions of variable scenarios, complex in terms of textures and colors. It is for this reason also unsuitable for future implementation. Reinforcing this decision even further, it should be noted that it will be necessary to opt for a computer vision method that will subsequently allow the identification of the position of the user's eyes with some accuracy. By the image subtraction method, the position of the eyes would have to be estimated and would be subject to many assumptions and associated errors. In case the final image region includes more than from the neck up, this task would be even more complicated.

3.2.2 Face Tracking

For the second method in Computer Vision's area, the field of facial identification and tracking is used. Despite the enormous evolution and progress that have been developed, facial detection is considered one of the most complex and challenging fields of investigation in computer vision [100]. Facial detection is characterised by being just a branch within a large area of research and technological development dedicated to the identification and tracking of objects. What sets it apart from the rest is its focus on working only with faces through their features[101]. It is defined, therefore, by the various technological methods of image processing [102, 103, 104, 105, 106, 107, 107, 108], which aim to detect and track human faces in photographs, videos or real-time applications [109, 110].

The Viola-Jones method [111] (Figure 3.4) is one of the most used developed (e.g. mobile phones applications). It is a robust and computationally undemanding facial detection algorithm that is still globally studied and implemented in highly successful software [109]. This will be explored during this section as a way to assess its feasibility for future implementation in the project. To this end, it is pertinent to dedicate the following paragraphs to a theoretical introduction to the method.

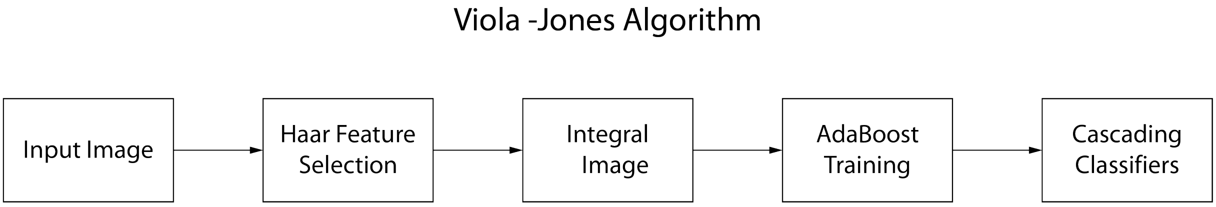


Figure 3.4: Viola-Jones Algorithm schematization..

Although some authors are dedicated to detecting faces through color photos [112], Viola-Jones' original method is based on the processing of images converted to gray scales, in the same way as previously demonstrated with image subtraction. The algorithm that is presented is sequentially organized and starts with a step of identifying features in the respective image for the detection of a possible face. This selection is performed through rectangular elements with black and white areas, placed on the image, called Haar patterns. The application of these elements proceeds through the sum of all pixels covered by each of the colored areas, followed by the subtraction of the calculated values of the black area from those of the white area. The simple arithmetic that is required allows us to assess which areas of the regions covered by the image are lighter or darker. However, Haar features do not cover the entire image and have variable dimensions according to the size of the visually relevant pattern identified. Therefore, it is possible to infer that there are numerous ways to arrange the various Haar features on an image. Many of these shapes may not cover regions of the image that include parts of a possible face, so not all of them will be useful for the intended purpose. This last aspect pointed out implies that the process becomes time consuming, even though the additions and subtractions are relatively simple.

Therefore, there is a need to accelerate this first step and restrict it to regions of interest only.

Given the aforementioned need, the next step focuses on calculating an image integral in order to increase the speed of the previous step. The image integral was the concept developed by Viola and Jones as an active solution in the implementation of their method. Image integrals in a pixel (x,y) are calculated by adding the intensity of each pixel that is in the rectangular region of the image delimited by pixels above the same column and to the left of the same row of (x,y).

With this approach, the sum of the pixel values covered by the Haar patterns is simplified so that fewer values will be needed to determine your result. For example, if it is necessary to calculate the sum of an area of 100x100 pixels, namely 10,000 pixels, only four values will be required through the image integral solution.

Regardless of the number of Haar features applied to an image, the image integral only requires to be calculated once.

It was also concluded that despite the countless ways to cover the image with one of the Haar feature sets, not all of them are useful. In framing this problem, Viola and Jones highlighted the need to train the algorithm to use the features only in regions of the image that were apparently relevant, that is, that could represent parts of an eventual human face. To implement this training in the algorithm, they proceeded to manually identify an extensive set of images with faces (4916 photographs) and one without faces (9544 photographs).

With the processed training made it possible to automatically distinguish, with some reliability, the regions in the image of potential interest. However, it is understandable that not all identified regions of interest will be equally important in determining the presence of a face. According to the proposal presented by the authors, in order to distinguish the importance of the regions of interest, weak regions would be defined as those that allow the detection of another additional object in the image, in addition to a face. However, when several weak regions are identified together, it is considered to be in the presence of a strong classifier. Complementing the qualitative classification of regions, the implementation of a parameter alpha (α_n) that quantifies the importance of each weak region was stipulated.

$$h(x) = \alpha_1 f_1(x) + \alpha_2 f_2(x) + \alpha_3 f_3(x) + ... \tag{3.7}$$

The determination of the respective weights that define the importance is guided by the AdaBoost algorithm described below:

1. Given example images $(x_1, y_1), ..., (x_n, y_n)$, where $y_n = 0$ for images that do not contain a face and $y_n = 1$ for those that do;

2. Initialize weights

   $$w_{1,i} = \frac{1}{2m}, \qquad w_{1,i} = \frac{1}{2l} \tag{3.8}$$

   for $y_i = 0, \quad y_i = 1$ respectively,

   where m and l are the number of negatives and positives respectively;

3. For $t = l, ..., T$

(a) Normalize the weights,

$$w_{t,i} \leftarrow \frac{w_{t,i}}{\sum_{j=1}^{n} w_{t,j}} \tag{3.9}$$

where w_t is a probability distribution

(b) For each feature, j, train a classifier h_j which is restricted to using a single feature. The error is evaluated with respect to w_t,

$$\epsilon_j = \sum_i w_i |h_j(x_i) - y_i| \tag{3.10}$$

(c) Choose the classifier,ht, with the lowest error ϵ_t

(d) Update the weights:

$$w_{t+l,i} = w_{t,i}\beta_t^{1-e_i} \tag{3.11}$$

where $e_i = 0$ if example x_i is classified correctly, $e_i = 1$ otherwise, and

$$\beta_t = \frac{\epsilon_t}{1 - \epsilon_t} \tag{3.12}$$

4. The final strong classifier is:

$$h(x) = \begin{cases} 1, & \sum_{t=1}^{T} \alpha_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^{T} \\ 0, & otherwise \end{cases} \tag{3.13}$$

where,

$$\alpha_t = log\frac{1}{\beta_t} \tag{3.14}$$

After implementing the AdaBoost algorithm, the quickest approach to check whether a set of regions of interest contains a face or not is through a sequential check. Logically, the verification will be guided by the value of the region's weight of importance, being verified from the largest to the smallest. In this way it will be possible to discard the sets whose totality of regions is not approved. In other words, the regions will be validated one by one until some gets rejected and, only when the last one is tested, will it be considered that a face has been detected. After a verdict on the facial identification of the respective set is taken, it proceeds to the validation of the next set, if any.

The method was then tested as a means of evaluating and predicting its performance in future applications of this project. With the help of *MATLAB* [99] and the various tools and functions that are accessible through this software, it was possible to implement the entire algorithm previously presented with readily. Thus, the code developed and the effort involved were greatly simplified, as *MATLAB* itself provides an object detector which already integrates the Viola-Jones detection algorithm and a classification model previously trained and prepared to identify any object, but by default targeted specifically at human faces.

During the conducted tests, the user's face was immediately and accurately identified, regardless of the background. Considering the conditions in which the tests were carried out, the fact of using a facial identification model trained by third parties raised questions about whether the algorithm developed was robust enough. In particular, the possibility of detecting faces where they did not exist was considered, given the textures present in the surrounding environment. On the other hand, it could also occur that the face being detected fails to be identified.

Tests were performed on several images that were captured immediately after starting the program and before proceeding with the Viola-Jones algorithm. Images that allowed us to ascertain and assess the limits of the program were submitted to the tests. More specifically, several variations were experimented, such as different, complex and basic backgrounds, varied lighting conditions and even different head positions, including variations in the distance between the user and the webcam.

From the extensive testing process it was possible to reach conclusions mentioned in the research papers reviewed. In addition to the positive points described above, the speed of the proposed method was verified. Although after starting the program it was necessary to wait a few seconds to obtain the results, these moments were mainly due to the opening time of the image preview window. That said, it does not present itself, so far, as a concern for a future real-time application.

However, problems were witnessed with variations in luminosity and especially with the location of the light source in relation to the face [100]. The method was found to be reliable in a wide range of light levels in the space. Intuitively, the algorithm failed, given such levels that, from the face in the image, it would be difficult for a person to identify little more than the sclera of the eyes. The big limitation related to the illumination that was detected refers to the moments when the light source is located behind or to the side of the user's head, creating a backlight effect. In these cases, the image results in a mostly illuminated environment, with the exception of the face that is completely or partially eclipsed. According to [113], it is mandatory to ensure proper lighting during test times as a way to ensure reliable detection results.

Finally, regarding the position of the head in relation to the camera, it was quite obvious that the previously trained classification model that was used was prepared to detect mostly faces parallel to the webcam plane. This conclusion was inferred from the fact that the program inherently has difficulty identifying a face that is rotated approximately more 45 degrees in relation to the vertical axis (turning the head to the left or right). However, rotations on the vertical axis are much better tolerated by the program compared to tests performed on images in which the head rotates about the remaining two horizontal axes (the one parallel to the webcam plane and the perpendicular one). Therefore, as evidenced by some authors [101, 113] faces will only be identified according to the type of training to which the model used was submitted. In this case, best results will be obtained when the head is directed approximately towards the camera, so that both eyes, nose and mouth are clearly visible in the image.

Regarding variations in the distance from the camera to the head, surprisingly, the algorithm was faithful in all the tests performed, considering that it was ensured that the face would always be completely visible in each attempt and that the tested distances fall within a range of approximately thirty centimeters and three meters.

For academic curiosity, situations in which the user wore a surgical mask were explored very briefly, but the results were not conclusive to the point of being able to predict whether in real-time applications the method would be prepared to detect the face, since only some of the images resulted in success, although only the eyes were visible.

In conclusion, although some negative aspects have been highlighted, this method, for the context in which it is inserted, surpasses the image subtraction method. It is, therefore, a suitable method to be implemented later in the development of the SST system.

Once the method chosen for facial detection is implemented, the need arises to track, over time, the previously identified face. Although it is possible to apply the Viola-Jones method to each frame, it becomes very demanding at a computational level, especially considering that a camera records videos at dozens of frames per second, at least. It would be inconceivable for most applications and especially for those intended for systems that work in real-time. On the other hand, an eventual attempt to apply the method in each frame could result in failure due to the limitations previously presented, namely if the user turns or tilts his head.

Therefore, it was necessary to explore an algorithm developed to apply Viola-Jones only once. The Kanade Lucas Tomasi (KLT) algorithm was introduced by Lucas and Kanade [114] and later developed by Kanade and Tomasi [115]. It is characterised by detecting distinct points of a certain object in an image, which are recognizable by their texture, to then be located along the frames [101, 116].

Briefly, after the user's face has been identified, within the area of the image where the face is located, several feature points, that stand out and can be reliable for the future track of the face, will be detected. There are several identification methods for this type of points, among which the work developed by Shi and Tomasi stands out [117], but it won't be targeted analysis in this document.

Then, the geometric transformation, between translation, rotation, enlargement or reduction, to which the set of identified points was submitted between each frame, will be estimated. The geometric transformations of the set translate the user's head movement relative to the camera. Once the old points, or at least most of them, are detected in the new frame, it is possible to predict the position of the person's face, even if part of it has been hidden during the head movement.

That said, the fact that, for various reasons, the feature points initially identified may not be located in consecutive frames is highlighted. These reasons, in most cases, are associated with the partial concealment of the face, or even cases where the system has not been able to predict the displacement of the respective point. Thus, the tracking algorithm will proceed in the same way, but with fewer points to control. Note that in the event that a point has been covered and reappears in the image, it won't be included again in the algorithm's detection cycle. Consequently, it is possible that the number of points at a certain point is close to zero or even zero, which implies that the face is no longer located and detected.

As previously performed, the proposed method was tested again, and this time the Viola-Jones program was reused. By doing so, it's possible to be aware of the limitations and capabilities of the program used to make the initial detection of the user's face. Evidently, *MATLAB* [99] was a crucial tool for carrying out the tests, and it should be noted that the pre-defined functions available in the software were again useful to simplify code development.

The tests conducted weren't meant to explore the application of the methods in previously recorded videos, so it does not fit in the context of this project. That said, experiments were only carried out with live videos captured by the computer's webcam where the *MATLAB* program was running.

Starting by highlighting the apparent improvements that were verified in the performance of the Viola-Jones method. Facial detection is only performed once, ideally when the head is facing the webcam, as described before. Taking this into account, during the test time it was possible to rotate the head much more freely without the detection failing. This is because the number of identified feature points is large enough. Thus, it is possible to continue to predict the location of the user's face even when some of the points are no longer controlled by the algorithm, due to head movements. However, detection ends up failing when too many movements are performed in a single test, causing a gradual but significant reduction in the number of points under control, implying a break in the tracking sequence. It should be noted that improvements in the Viola-Jones algorithm are apparent, given that the method remains unchanged and in turn the highlighted improvements result only from the synergy between the Viola-Jones and KLT algorithms.

Regarding the problems related to lighting that were defined above, no changes were detected, so the suggestion to ensure that there is adequate lighting for head orientation, always favoring the lighting of the face, remains crucial.

The speed of video presentation with the face and characteristic points identified is stable and perfectly compatible with real-time applications, as the delay in reaction and processing time is practically imperceptible. Again, the only moment that proves more costly in terms of time refers to the moments dedicated to starting the program itself and opening the display player. Occasionally, small seconds are detected in which the video with live results temporarily freezes.

Once again, the surrounding environment was not presented as an obstacle to the success of the program, so the tests did not highlight any aspect that could interfere with the performance of the algorithms. However, curiously the identification of feature points, in the region where a face was detected, sometimes defined points in the background in areas where the texture was more prominent. These points were quickly eliminated with the first movements of the head and never affected the functioning of the program or the quality of the results.

There was, however, one problem that stood out during the tests. The previous paragraphs not only foresaw this, but also briefly discussed it. As the number of feature points continuously decreases or even gets to zero, implies a complete collapse in the face tracking process. After this failure the video continues to be presented live, but without any additional information, even when the face is perfectly oriented towards the camera and visible. This was an aspect that was never highlighted by any author within the reviewed works.

That said, the need to solve the problem arises. To define a plan that is successful in this situation, it is important to understand the origin of the failures that have emerged. In particular, facial detection fails because the number of feature points decreases substantially. This is because the program is not constantly identifying the region of interest to renew the definition of feature points. However, it was immediately established that constantly identifying the region of interest, that is, detecting the face in every frame, was unthinkable. Therefore, there is a need to create a cycle that allows the area of interest to be located with some regularity. The tests showed that there is only a risk of collapse when the number of points is very low, meaning that the need is not periodic over time, but in terms of the number of feature points that are still being controlled by the algorithm. Thus, the cycle will be defined by re-detecting the user's face whenever a minimum of feature points is not met, so that they are renewed and inserted in the sequence of tracking more and new points (Figure 3.5).

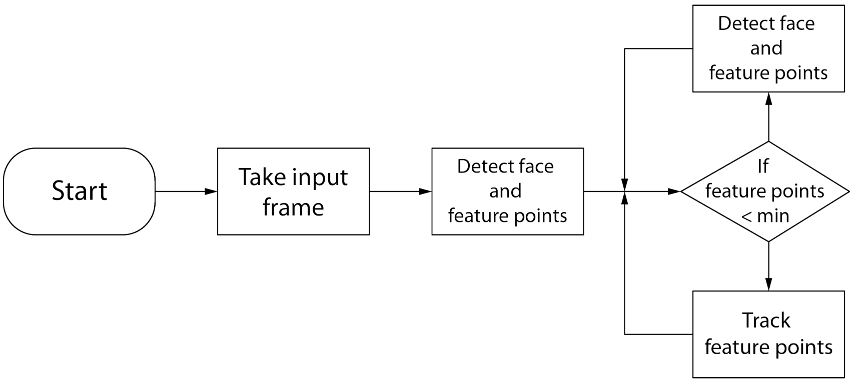


Figure 3.5: Schematization of the Method of Face Detection and Tracking.

In order to determine the minimum number necessary for the success of this solution, it will be indicated to use the same approach adopted to define the optimal threshold value in the image subtraction suggested by [96].

The fact that the problem, now solved, has never been highlighted by any author may be due to the ease in proposing simple strategies to solve it.

To end this section dedicated to computer vision, the feasibility of implementing the Viola-Jones and KLT algorithms together as a way to respond to the first need to detect the user's faces is evident. Additionally, it is important to reinforce the idea expressed above referring to the fact that *MATLAB* [99] is prepared to identify any object other than a face and, fortunately, the software also provides a trained model to detect the eyes. This detail will be fundamental in the future development of the SST system, thus further highlighting the advantages of opting for these methods, in unequivocal contrast to image subtraction.

3.3 Depth Estimation

Once the user's face is detected and, consequently, the earlier defined equivalent point of view, emerges the need to be able to convert the coordinates of the point in the image captured by the webcam, to spatial coordinates. That is, consider a three-dimensional referential whose origin is located exactly in the geometric center of the screen, assuming that it is a flat rectangle. With the implementation of the method explored in this section, it will be possible to obtain the coordinates from the user's point of view in accordance with the screen's referential.

The major limitation that is clearly evident in this area dedicated to Depth Perception is based on the fact that the only resource and information that is acquired about the objects of interest is a single image captured by only one conventional webcam. Thus, the only parameters that will allow the estimation of the object's spatial coordinates will be its apparent dimensions in the image and its offset from the center of the image itself. That said, it seems intuitive that to estimate the distance of an object from the webcam, under these conditions, it will be necessary to extract the coordinates in the image of at least two points of the object. This conclusion is justified by the fact that the location of a single point does not provide any dimensional information about the object. However, with the location of two points, it becomes possible to assess the distance between them in the image referential and process this data with the real distance, on the physical object, between the respective two points identified in the captured image. Alternatively, if the real distance between both points is not possible to determine, the dimensions of another element present in the image should be used, whose real distance between other points that belong to it is possible to calculate. Thus, according to the method explored in the following sections, it will be possible to estimate the distance to the camera from the initial object of interest through an analysis of the perspective's perception of the environment captured in the image.

There is also another factor to be taken into account associated with the fact that the specified purpose is being served only by one conventional webcam. The implemented method is expected to be highly dependent on the characteristics of the webcam itself, given that the resolution, calibration, lenses, filters and many other aspects are completely responsible for defining the perception of the environment captured in the image. Any variation in a parameter of the listed characteristics will contribute to the distance in the image between any two points being different, as in accordance to what was discussed in the previous paragraph. In order to obtain reliable and close-to-reality results, this factor will greatly influence the development of the depth perception method.

In this area, only one method will be explored and discussed. This one, ideally, will analyze and represent the perspective's perception of reality captured by the webcam through non-linear statistical regressions obtained empirically for specific characteristics of the webcam itself.

3.3.1 Exponential Fit

Currently, the evolution of Computer Vision's success in developing technologies aimed at estimating depth is unquestionable. However, the determination of dimensions in relation to a camera is still a complicated objective to achieve, despite the extensive research work that has made it possible for this technology to be used in a wide range of applications. Its versatility of implementation is due to the fact that increasingly, with the integration of intelligent systems in industry and even in everyday life, depth perception is a basic requirement for tasks associated with navigation and planning, or even to map a real three-dimensional environment, for example.

There is an endless list of methods intended for this purpose that are supported by the use of equipment from the most conventional to the most complex. From commercial cameras, to stereo or 3D cameras, among other more advanced options, all are intended to deal with the problem in analyzing three-dimensional environments through two-dimensional projections.

From the existing methods, research works guided by authors who focused on estimating the distance of objects from a single camera were reviewed. This brief literature review appears as a way to understand the current state of the art of technology developments adapted to the context in which this project is inserted. Concretely, the objective will be, through the knowledge acquired with the study of the methods, exploring viable options to support the development of the depth estimation method, to be later implemented.

Alphonse and Sriharsha [118] presented a mathematical model that allows them to estimate the distance of an object to the camera, through the analysis of a fraction of the captured image. In order to accomplish this, they established a geometric relationship between the parameters of the camera lens aperture radius, the distance from the object to the camera and the dimension of the object in the captured image plane. The results proved that the approach, under the conditions of the tests carried out, presents a very high precision of 98.1%.

Seal et al. [119] developed the catadioptric stereo system, in order to estimate the position of objects, similar to methods based on two cameras and stereo cameras. The system that they propose to use requires the assembly of two pairs of plane mirrors that have been precisely designed so that in the correct position they have the capacity to generate two virtual cameras. This provided a compact and inexpensive system with a theoretical depth resolution of 5.8 cm for a working distance of 2 meters.

Ranftl et al. [120] propose a dense depth estimation technique that uses a monocular camera to perceive distances, in the images captured, through dynamic scenes. For this effect, two consecutive frames are used to produce a dense depth map by reconstructing the surrounding environment and moving objects. The presented method is guided by motion segmentation algorithm proposed by the authors. The algorithm is capable of identifying each model of optical flow independently, based on its epipolar geometry. In terms of monocular depth estimation in dynamic scenes, authors claim their approach outperforms existing methods.

In a similar approach to Alphonse and Sriharsha [118], Joglekar et al. [121] presented a geometry based algorithm meant to be implemented for specific camera parameters. This one can be distinguished from others due to its particularity for being specially designed to estimate the distance from other vehicles on the road. It uses road geometry and requires the point of contact between the road and the object of interest to be capture in the image. The authors tested the method and compared it with other linear and non-linear processes [122], claiming its accuracy (96%) exceeds the remaining ones.

Finally Rahman et al. [123] demonstrated the existence of the relationship between the physical distance that separates the object from the camera and the height in pixels of the object itself in the image plane.This relation is used to train a system that determines the physical distance given the object's pixel height. During the tests carried out by the authors, this method provided 98.76% accuracy.

Using the results of a literature review, several strategies are identified that may incorporate the future method that will be used in this study. The method to be developed will be quite demanding in terms of requirements. The main reason for this is that the future SST system will integrate different methods from the three areas covered in this chapter. The Computer Vision method, by itself, comprises the application of multiple algorithms and with the additional and simultaneous operation of the Depth Perception and Screen Output methods, the more demanding the system will become at a computational level. Therefore, it is necessary to guarantee that the integral system will be prepared to work in real-time with high precision results. Being aware that the properties of the computer's graphic capacity to be used when testing the final system are still unknown, it is necessary to define, from now on, the respective requirements of the method to be developed in this area. Although they won't be able to certify the validity of the method, these tests will be critical to allowing at least a proof-of-concept to be established. Criteria will be established based on the needs of the project and on the research findings.

From now on, it should be noted that none of the authors addressed the computational requirements of the explored methods nor their response times. Due to this fact, it will be difficult to define the criteria associated with the parameters mentioned above. However, it is possible to conjecture the performance of each approach by the type of algorithm used.

The majority of the articles reviewed deal with mathematical and geometric relationships that involve two or three parameters. These parameters are apparently easy to obtain, considering that they refer to intrinsic characteristics of the camera itself or dimensions extracted directly from the image to be analyzed.

The method developed by Ranftl et al. [123] isn't included in what was described in the previous paragraph. This method resorts to complex processes of segmentation of dynamic scenes in independent rigid motions, where each motion is represented by a fundamental matrix. Although no comment by the author has evidenced it, it will be acceptable to consider that this method stands out for being more demanding and time consuming given the mathematical complexity involved. These may even reveal a slower response speed that prevents it from being useful in real time to tens of frames per second.

- For the first criterion, it is defined that the method should be based on a geometric and mathematical relationship involving a maximum of three parameters.

Regarding the type of parameters that can be used for the defined criterion, it is important to establish that these must be coefficients or characteristics, known *a priori*, involved in the system, or that can be obtained directly during the process. In this description, although parameters such as the camera aperture radius or the lens focal length may be included, these won't be considered in the defined criteria. The reason for this decision is based on the fact that, as already mentioned, the camera to be used will be a webcam integrated in the transparent monitor. To clarify, the two parameters pointed out are characteristic of photographic cameras and not webcams. Thus, methods that relate specific parameters of equipment not adopted in the system will be excluded.

Additionally, Joglekar et al. [121] establishes as necessary, for their method, to capture the point of contact of the object of interest with the ground. However, in the context of this project, considering the user's head as the object of interest, if the image does not capture the person's feet, it becomes impossible to apply the method. The second criterion is thus established:

- The parameters must be values obtained directly from the camera's characteristics, or any other object/equipment involved in the process. They may also be extracted directly from the captured images.

Furthermore, the method needs to produce results of high precision. The requirements will be defined by a minimum precision value of the final results. This value will be established through the minimum precision value obtained by methods that are based on geometric and mathematical relationships involving three or fewer parameters. Consequently:

- The method should have a minimum precision of 96% in the final results obtained.

According to the defined criteria, it is possible to verify that the method developed by Rahman et al. [123] it's the only one that fits perfectly. Thus, there is a need to review the method presented by the author in more detail.

Rahman et al.'s method attempts to determine the distance of an object, of which physical dimensions are known, to the camera. The object is only considered to move in the same horizontal plane as the camera. The distance is calculated by the Pythagorean Theorem, so the hypotenuse is the total distance, and each leg refers to the longitudinal and lateral displacement of the object in relation to the camera position.

The author conducts two practical tests as a way to evaluate and compare the pixel dimension of the object at different distances from the camera.

For the first test, the object is placed in front of the camera, with zero lateral displacement, in fourteen different positions that correspond to different longitudinal displacements. This test resulted in a quadratic regression that expresses the longitudinal distance, in centimeters, as a function of the height of the object, in pixels.

In the second test, for each of the fourteen distances above, the object was moved laterally multiple times. For each position, the quotient between the height of the object, in pixels, and the lateral displacement, in cm, was calculated. Then, for each of the fourteen depths, the average of all the respective quotients was calculated. This test resulted in another quadratic regression that allows expressing the mean of the quotients (px/cm) as a function of the longitudinal distance (cm) at which the object is located from the camera.

Through the expressions obtained, the author defined a method that allows him to determine the relative position of the object to the camera, given only its height in pixels in the captured image.

After a more in-depth review, there is a need to highlight some relevant aspects for the following paragraphs referring to the method to be developed.

For Rahman et al. the process is developed with a focus only on a single object and its dimensions for different distances. Consequently, to estimate the distance of another object with different physical dimensions, it will be necessary to carry out two tests again. These allowed us to obtain two new quadratic regressions to determine the lateral and longitudinal displacements of the new object specifically.

On the other hand, the method was developed without any reference to the characteristics of the photographic camera, as the only information provided is the resolution of the images, namely 640X480 px. This aspect becomes fundamental for the interpretation of the presented method, as it will be impossible to assume that the expressions obtained from the non-linear regressions are accurate for a different camera, even if the resolution and the object of interest are kept constant. The same reasoning can be extrapolated to eventual changes in the camera's resolution, lens or aperture radius. In other words, the method is restricted to the camera with which the images were taken and the parameters that characterise it, during the tests carried out.

In general, for any change that is made to the conditions under which the method was developed, it will be necessary to carry out new tests. This fact allows us to understand that the method, despite its high precision in the results, is not characterised by its versatility or efficiency.

One last aspect to point out about the method presented by Rahman et al. is highlighted. The application of quadratic regressions may be considered inappropriate to the context in which it operates. As a way to substantiate the previous statement, it will be pertinent to individually analyze the two tests from which the regressions came.

> For the first test, the longitudinal distance is expressed as a function of the height of the object by a second-degree function. This approximation function implies that for object heights greater than the abscissa of the vertex of the parabola, the expected distance from the object to the camera increases rather than decreases.
>
> For the second test, the mean of the quotients is expressed as a function of the longitudinal distance that the object is from the camera. When representing this relationship by a quadratic function, it is inferred that for distances greater than the abscissa of the vertex of the parabola, the height in pixels of the object increases, which in fact is not what happens.

Hence, the method may only be applied to objects that are at a longitudinal distance of less than 183.63 cm and whose height is less than 301.79 px. More specifically, from an interpretation of the values it's possible to conclude that the object's position must be between 72.28 and 183.63 cm in depth in relation to the camera. These values were obtained from the vertices of the functions resulting from the regressions. Since the resolution of the image is 640x480 px, it would be pertinent to opt for a non-linear regression that would allow the applicability of the method for a range of values that includes any object completely captured by the camera. Everything that has now been pointed out highlights even more limitations, in addition to those that have already been presented.

The development of the method to be implemented in the system begins. The entire discussion of methods and definition of criteria will be crucial to guide this stage of the project. Thus, it was decided to establish that this method should result in a mathematical relationship involving the longitudinal distance of the object to the camera (cm), the physical dimension of the object (cm) and the dimension of the object in the image plane (pixels). This will ensure that, regardless of the object and its dimension, only one relation is suitable to assess the depth, as long as it is fully captured by the camera.

The expression relating the parameters is therefore a function in $\mathbb{R}^3$, as follows:

$$f(v,p) = g(x) \tag{3.15}$$

where:

- v represents the virtual dimension of the object, in pixels;

- p represents the physical dimension of the object, in centimeters;
- x represents longitudinal distance from the object to the camera, in centimeters.

However, it is valid to simplify the function and convert it to $\mathbb{R}^2$. To clarify the last sentence, the concept is illustrated as follows:

> Consider two spheres, A and B, of different diameters that are at the same longitudinal and lateral distance from a camera. The diameter of sphere A is twice the diameter of sphere B. Both have been photographed side by side multiple times. Each image was captured at a greater longitudinal distance than the previous image, with the lateral distance always remaining constant. At the end of the experiment, it was found that in all images the dimension, in pixels, of ball A was always twice the dimension, in pixels, of ball B.

Note that the previous paragraph is just a figurative scenario. Although the experiment has not been carried out, it corresponds to an illustration of reality and this concept can be easily verified.

As demonstrated by the example given, dimensional variation of objects in images, depending on their distance from the camera, is the same regardless of their physical dimension. More specifically, in a image, the representation in pixels of a unit of physical length will be the same for a given distance, regardless of the object.

Therefore, it is possible to simplify the function and convert it to $\mathbb{R}^2$ by creating a variable that is defined by the ratio between the pixel length of the object in the image and the physical length in centimeters. This variable is characterised by corresponding to the inverse of the reality representation factor (δ_x) in the image plane, in cm/px, in the object plane, parallel to the image plane. Thus obtaining a relationship between only two parameters, since one of them is dependent on the aforementioned lengths and the other is the distance from the object to the camera.

$$f\left(\frac{1}{\delta_x}\right) = g(x) \tag{3.16}$$

where:

- δ_x represents the reality representation factor, in pixels/centimeters.

To clarify, δ_x factor is simply characterised by the photographic scale of a plane parallel to the photographic plane, at a distance x from the camera. This factor shows that, for the respective plane, a pixel on the image corresponds to a specific length in reality, in centimeters.

Note that the calculated distance, referred to in the previous paragraphs, is not the total distance, but only the longitudinal distance. This is because it is assumed during the development of this method that lateral and vertical displacements have a negligible influence on the dimension of the object, in pixels, in the image. This was a condition verified by Rahman et al. [123]. In this way it will be possible to determine the vertical and lateral displacements by the deviation of the geometric center of the object in relation to the center of the image. Given that, as noted above, in the entire plane of the object, the reality representation factor is considered constant. The vertical and lateral displacements are obtained by the product of the representation factor (cm/px) with the vertical and lateral deviation (px), respectively.

The method should be appropriate to the widest range of longitudinal object distance values possible and, ideally, be suitable to present results for any object fully captured in the image, within precision criteria. To achieve this goal, it is necessary to define the relationship with a function that assumes a behavior similar to the dimensional variation of objects in the images, depending on their longitudinal distance.

To define the behavior of the dimensional variation of objects, the following principles are established. These may also reflect the behavior of the function in $\mathbb{R}^2$:

- Since the dimension of an object will never be negative, it is obvious that the function must have a horizontal asymptote on the abscissa axis, reflecting the convergence towards a null dimension as the longitudinal distance of the object tends approaches infinity. It is assumed that the function will never be null, since, theoretically, the dimension of the object in the image will only be null when it is not present in the image.
- It is not relevant whether the function intersects the vertical axis x=0 or not, since for values of longitudinal distance very close to zero it is foreseeable that the object will not be completely captured.
- The dimension of an object at a certain distance from the camera tends to gradually decrease as distance increases. For this reason, the function should be characterised by being strictly decrescent.

It was also established that the method would be developed for a specific camera and consequently for its respective characteristics. This decision is due to the fact that the camera to be used in the SST system is the webcam integrated in the transparent monitor and, therefore, the need to change the camera or its parameters is not foreseen. Therefore, it is perfectly feasible to opt for a method under these conditions.

In order to apply all the theory developed so far and obtain the function that characterises the desired relationship, a practical test was performed, similar to that conducted by Rahman et al..

Two objects, of different heights, were photographed at twenty-one different longitudinal distances from the camera each. The images were later analyzed individually and the height of each object, in pixels, was measured and associated with its respective longitudinal distance. For each measured height, the quotient between the dimension, in pixels, and the height of the respective object, in centimeters, was calculated. Thus, a set of forty-two ordered pairs (x,y) was obtained, where x refers to the distance from the object to the camera and y refers to the quotient defined for the respective distance x. Then, the ordered pairs were represented in an orthonormalized referential, as shown in Figures 3.6 and 3.7.

Regarding the data from the test carried out, the following aspects are highlighted:

- The longitudinal distances considered during the test were set between the lengths 50 and 250 centimeters, inclusive, with increments of 10 centimeters separating each one;
- The physical heights of the objects are respectively 21.75 and 16.50 centimeters;

  resolution of captured photos is 1280 x 720 and the camera has a resolution of 0.9 MP.

The first observation to be highlighted, before continuing with the development of the method, refers to the fact that the ordered pairs, corresponding to a certain distance, are represented by practically coincident points for both objects. It is a predictable result that highlights the fact that a unit of length in a image is independent of the object, regardless of longitudinal distance. In other words, the representation factor and consequently its inverse, represented by the ordinate of the ordered pairs, are constant for each object's depth value, represented by the abscissa of the ordered pairs.

As part of the process of defining the mathematical function that aims to fulfill the previously defined behavior, a non-linear regression was carried out, on the point cloud representation of the data obtained from the test. With all criteria considered, an exponential regression was chosen as it meets the requirements presented. The simple exponential regression is defined as follows (Figure 3.6):

$$f(x) = a \cdot e^{b \cdot x} \tag{3.17}$$

where a and b are the independent coefficients to be determined.

The following results were obtained:

Coefficients	(with 95% confidence intervals)
$a = 22.05$	$(20.43; 23.67)$
$b = -0.009256$	$(-0.009956; -0.008555)$

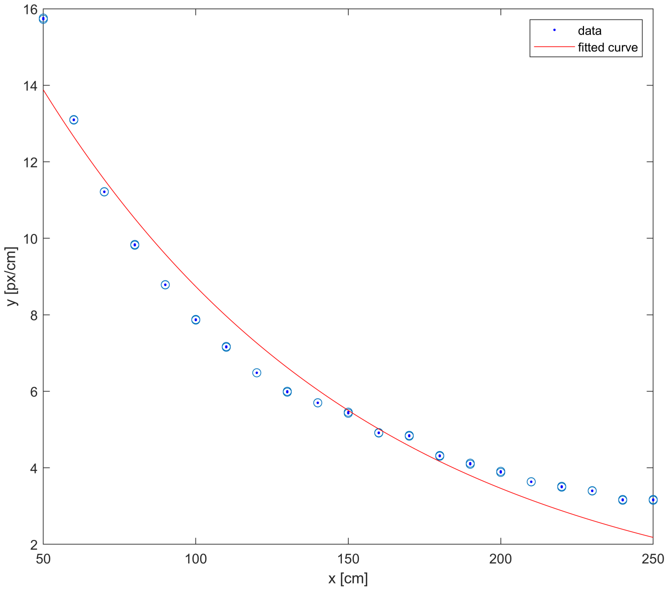


Figure 3.6: Simple Exponential Model fitted to the Data set.

It remains to assess the accuracy of the method. Although the coefficient of determination R-Squared is not adequate to evaluate the correlation of nonlinear regressions [124] it was the method used by the reviewed authors to define the accuracy percentage of the method they developed. The determination of the coefficient of determination will also be carried out so that it is possible to make a comparison with the results obtained, through the respective process, in the methods studied above. The coefficient obtained will not be valid according to Spiess et al. [124], and therefore will not be considered for any additional occasion that transcends the comparison between methods.

The calculation of R-Squared and Adjusted R-Squared was done as follows [125] for a data set with n values y_i, each associated with a fitted value f_i:

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}} \tag{3.18}$$

where the sum of squares of residuals is given by,

$$SS_{res} = \sum_{i}^{n} (y_i - f_i)^2 \tag{3.19}$$

and the total sum of squares by,

$$SS_{res} = \sum_{i}^{n} (y_i - \overline{y})^2 \tag{3.20}$$

where $\overline{y}$ is the mean value of the acquired data:

$$\overline{y} = \frac{1}{n} \sum_{i}^{n} (y_i) \tag{3.21}$$

$$Adjusted\ R^2 = 1 - (1 - R^2) \frac{n - 1}{n - k - 1} \tag{3.22}$$

where k is the number of independent predictor variables in the model.

$$R^2 = 95.67\% \qquad ; \qquad Adjusted\ R^2 = 95.44\%$$

As a way of validly evaluating the accuracy of the developed method, it was decided to determine the Residual Standard Error (S) and the Root Mean Square Error (RMSE). In accordance with the literature [126] these are adequate and statistically accepted processes to evaluate predictive methods, whether linear or not.

$$RMSE = \sqrt{\frac{\sum_{i}^{n} (y_i - f_i)^2}{n}} \tag{3.23}$$

$$S = \sqrt{\frac{\sum_{i}^{n} (y_i - f_i)^2}{n - k}} \tag{3.24}$$

$$S = 0.726430\, px/cm \qquad ; \qquad RMSE = 0.708923\, px/cm$$

In general, the coefficients of determination R^2 and Adjusted R^2 are characterised by representing the percentage of variation of a dependent variable, relative to a regression model established by independent factors [125].

On the other hand, the Residual Standard Error (S) and the RMSE are two statistical parameters that define the standard deviation of the residuals. More specifically, they assess the difference between the actual values and the respective ones coming from the predictive model. In this way, it is possible to characterise the density of points around the curve defined by the regression [127, 128].

For most predictive models, it is most recommended to evaluate the accuracy by the Residual Standard Error and RMSE (Jim frost, Regression analysis). These parameters quantitatively classify the mean error of the regression, in the units of the variable itself. The smaller the value of S and RMSE, the greater the accuracy of the regression. While R^2 and Adjusted R^2 do not allow us to have such an objective and concrete perception of the method's level of accuracy because they are dimensionless parameters. The precision will be greater the closer the coefficient value is to 100%.

Compared with the accuracy shown for the revised methods, and taking into account the defined minimum criteria, for the method that is intended to be developed in this area, we find that the value of R^2 is less than 96%. Although this percentage is not mathematically valid, for the reasons already presented, in relation to the others that used the same coefficient of determination, the present method presents a lower accuracy. In light of this, there is a need to obtain another function that results from a non-linear regression and allows to obtain results with a smaller average deviation from the data acquired from the practical test.

We opted for a compound exponential regression, identical to the previous one, however, the approximation function will be the result of the sum of two exponentials (Figure 3.7):

$$f(x) = a \cdot e^{b \cdot x} + \ c \cdot e^{d \cdot x} \tag{3.25}$$

where a, b, c and d are the independent coefficients to be determined.

The following results were obtained:

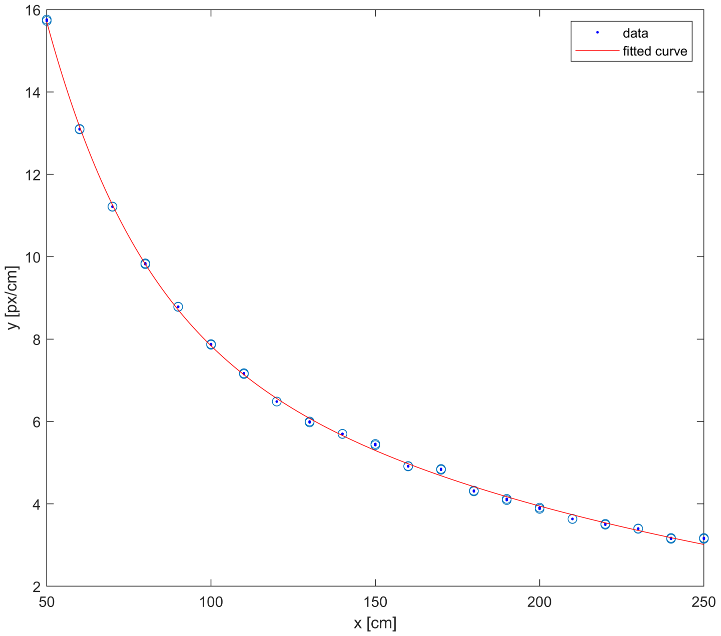


Figure 3.7: Compound Exponential Model fitted to the Data set.

Coefficients	(with 95% confidence intervals)
$a = 41.12$	$(37.34; 44.90)$
$b = -0.0351$	$(-0.03787; -0.03233)$
$c = 11.17$	$(10.39; 11.94)$
$d = -0.005246$	$(-0.005583; -0.004908)$

The coefficients of determination, R^2 and Adjusted R^2, were calculated:

$$R^2 = \ 99.9452\% \qquad ; \qquad Adjusted\ R^2 = \ 99.9392\%$$

As well as the Residual Standard Error and the Root Mean Square Error:

$$S = \ 0.0817\ px/cm \qquad ; \qquad RMSE = \ 0.0797\ px/cm$$

The presented results demonstrate that it is possible to implement a method that fulfills all previously established criteria.

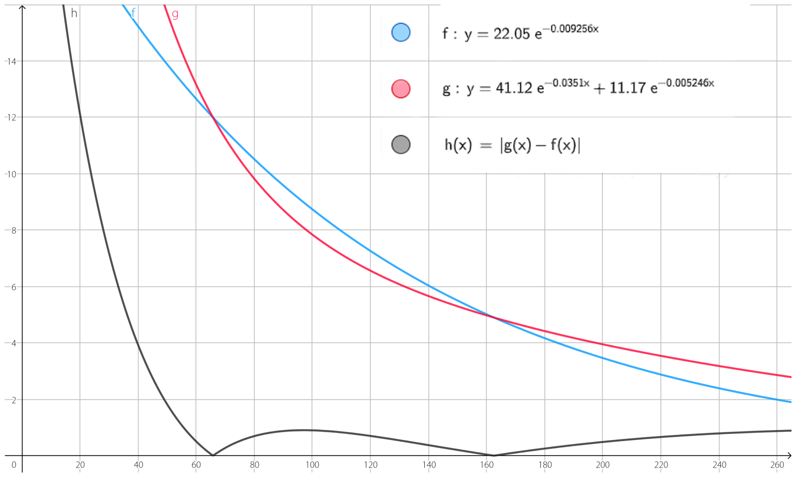


Figure 3.8: Comparison and Representation of the Absolute Error between the Simple and Compound Exponential Models .

3.4 Screen Output

Finally, after interpreting and analyzing all the data extracted from the images captured by the webcam, it becomes possible to proceed with methods that adapt the image on the transparent screen according to the user's field of view.

The approach chosen for displaying images on the screen, as explained above, is based on approximating the user's field of vision to a single point of view instead of two (the person's eyes). This approach will be referred to as monoscopic rendering because the screen will be configured to display a single image instead of displaying an image specially designed for each eye (stereoscopic rendering). The concepts now presented will be defined and discussed in detail in the following sections.

In light of the previous paragraph, it is also important to remember that, for this project, the user declares himself/herself free to have any device that allows him/her to conveniently deal with the SST system. Therefore, it excluded the exploration of methods that require the use of helmets, remotes, glasses, among others. This means that the aforementioned monoscopic rendering does not translate into an image that results in the superposition of two, corresponding each to the different points of view, respective of each eye. If so, the SST system required the user to interact with the screen through glasses identical to those provided in 3D cinema sessions. To clarify, the screen is not a 3D imaging system, but a normal 2D screen. Therefore, the objective will be for the user to observe the real 3D environment augmented with 2D virtual elements that are superimposed and displayed in the monitor.

This approach presents itself as a simpler option to develop due to the type of equipment that is required, among other several advantages. However, opting for monoscopic rendering methods also comes with its limitations and even disadvantages compared to stereoscopic applications. In the following sections, we will discuss the implications of implementing such methods.

Two distinct methods will be explored and developed in this area dedicated to Screen Output. The first will be dedicated to dynamically arrange virtual markers whose location on the screen is determined by calculating the intersection of the line of sight, which joins the user's eyes and the object of interest, with the plane of the screen. The second method is based on a 2D graphical representation of a three-dimensional virtual environment. This virtual world aims to reflect the real scenario that is on the opposite side of the user in relation to the transparent monitor. As a result, the user would be able to see a window into that world through the screen, corresponding to what is expected to be visible through it. The image will be defined according to the position and orientation of the user's head, delimited by the rectangular geometry of the screen.

3.4.1 Line of Sight Intersection

For the first method presented, the coordinates of the point located by the methods of the previous sections will be considered. Namely, the one that is defined by the midpoint of the segment limited by the user's eyes. Henceforth, it is possible to predict the user's point of view in relation to the environment that surrounds him, but mainly, to what is on the other side of the transparent screen. Since the coordinates of the objects of interest that are on the opposite side of the screen are known, it becomes possible to establish the parametric equations that define the user's line of sight for a specific object.

The line of sight parametric equations will be suitable for whatever the position of the user or the object. For any movement that is detected on either side of the screen, the previous methods will be responsible for identifying and locating the new position of the respective object or of the user itself. Thus, the new coordinates will be extracted and updated in the parametric equations so that the line of sight is renewed over time.

Note that the equations fit into a three-dimensional coordinate system (Figure 3.9) where both the user and the object are defined by two points that establish the extremes of the line of sight, represented by a straight line. In the same reference, it is also possible to define the equation of the plane that represents the screen in its position and orientation in relation to the user and other objects.

The purpose of this method will be to dynamically display virtual markers on the transparent screen so that, from the user's point of view, they coincide with the objects of interest behind the screen. To successfully achieve this result, it is necessary to determine the intersection point of the line of sight with the transparent screen. Transposing this concept to the three-dimensional referential, the intersection of the straight line with the defined plane will result in the representation of a point. The coordinates of this point correspond to the position of the virtual marker for the respective object (Figure 3.9).

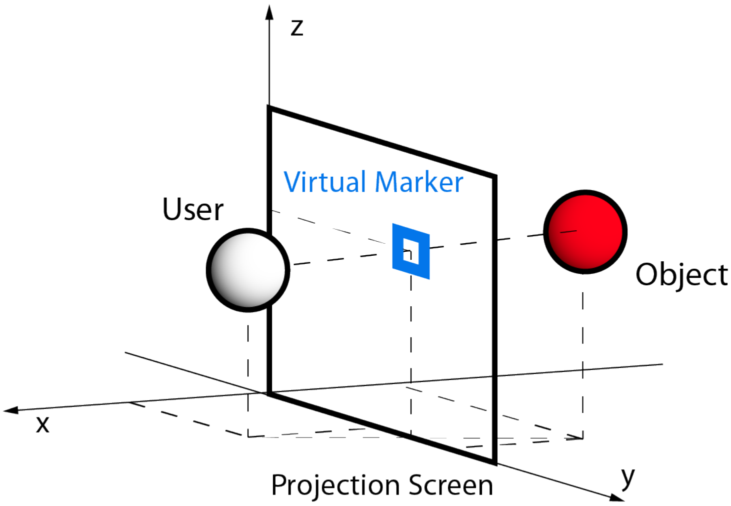


Figure 3.9: Virtual Marker's point of intersection inside the projection area.

In this case, there is no possibility of the line being parallel to the plane, since the object and the user are always on opposite sides of the screen. Therefore, it will always be possible to calculate the coordinates of an intersection point between the two elements. However, knowing that the transparent screen has a limited area, the point of intersection of the line with the plane may be located in a region of the plane that is not covered by the limits that define the screen area. In these cases, it will result that the virtual marker of the respective object will not be visible for the specific positions of the user and object, under those conditions (Figure 3.10).

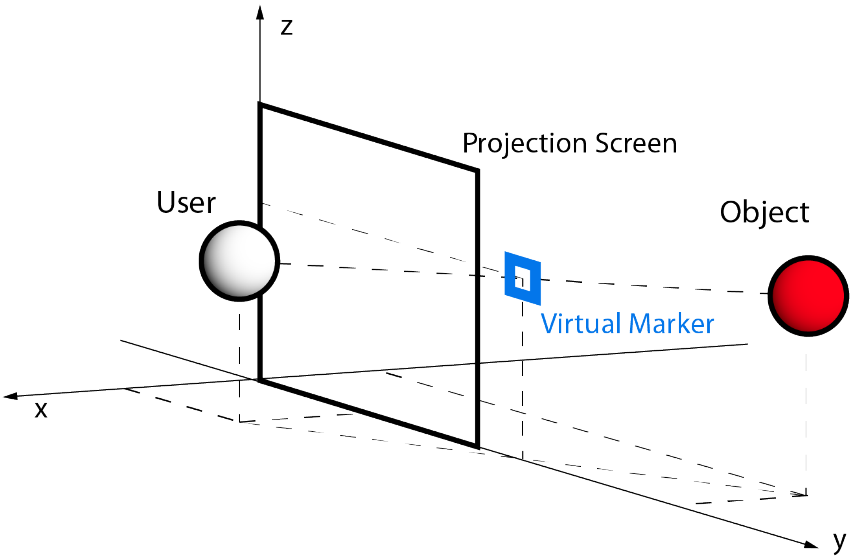


Figure 3.10: Virtual Marker's point of intersection outside the projection area.

The present method, described throughout the previous paragraphs, is developed as follows:

The point Q (Q_x, Q_y, Q_z) defines the position of the user that allows evaluating his/her point of view. The point P (P_x, P_y, P_z) defines the position of a specific object on the other side of the screen. Both points are movable over time and their coordinates are updated, as stated above.

The transparent screen is defined by an infinite plane, α, in the frame, as shown:

$$\alpha : \quad ax + by + cz = -d \tag{3.26}$$

where a, b, c and d are the parameters responsible for establishing the position and orientation of the screen in relation to the user and object. These parameters are constant, as the screen is not expected to move while using the SST system.

It should be noted that a, b and c correspond to the coordinates of the plane's director vector, $\vec{s}$, which is represented orthogonal to the respective plane: $\vec{s} = (s_x, s_y, s_z)$.

The line of sight is defined by the line h, which connects P and Q. The line h can be represented vectorially as follows:

$$h : (x, y, z) = (Q_x, Q_y, Q_z) + \lambda(n_x, n_y, n_z) \quad , \lambda \in \mathbb{R} \tag{3.27}$$

where $\vec{n} = (n_x, n_y, n_z)$ is the direction vector of the line defined by:

$$\begin{pmatrix} n_x \\ n_y \\ n_z \end{pmatrix} = \begin{pmatrix} P_x \\ P_y \\ P_z \end{pmatrix} - \begin{pmatrix} Q_x \\ Q_y \\ Q_z \end{pmatrix} \tag{3.28}$$

To simplify the calculations that follow, it will be assumed that $\lambda = 1$.

From the vector equation of the straight line,

$$\frac{x - Q_x}{n_x} = \frac{y - Q_y}{n_y} = \frac{z - Q_z}{n_z} \tag{3.29}$$

Transposing the presented equalities to a system of 3 equations:

$$\begin{cases} xn_y - yn_x = Q_x n_y - Q_y n_x \quad \leftarrow \\ zn_y - yn_z = Q_z n_y - Q_y n_z \quad \leftarrow \\ xn_z - zn_x = Q_x n_z - Q_z n_x \end{cases} \tag{3.30}$$

Of the three equations that are presented, only two will be necessary to proceed with the method, since the third is linearly dependent on the remaining two.

The point of intersection between the plane α and the line h is defined by $I(I_x, I_y, I_z)$. The coordinates of point I are obtained by solving the following matrix equation:

$$\begin{pmatrix} n_y & -n_x & 0 \\ 0 & -n_z & n_y \\ a & b & c \end{pmatrix} \begin{pmatrix} I_x \\ I_y \\ I_z \end{pmatrix} = \begin{pmatrix} Q_x \cdot n_y - \ Q_y \cdot n_x \\ Q_z \cdot n_y - \ Q_y \cdot n_z \\ -d \end{pmatrix} \tag{3.31}$$

Where,

$$\begin{cases} n_x = \ P_x - \ Q_x \\ n_y = \ P_y - \ Q_y \\ n_z = \ P_z - \ Q_z \end{cases} \tag{3.32}$$

Considering that the screen will always be installed vertically, we can proceed with a simplification that defines the screen plane parallel to the zOy plane. Specifically, plane h may be represented as:

$$\alpha : \quad x = \ -d \tag{3.33}$$

Implying,

$$\begin{pmatrix} n_y & -n_x & 0 \\ 0 & -n_z & n_y \\ 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} I_x \\ I_y \\ I_z \end{pmatrix} = \begin{pmatrix} Q_x \cdot n_y - \ Q_y \cdot n_x \\ Q_z \cdot n_y - \ Q_y \cdot n_z \\ -d \end{pmatrix} \tag{3.34}$$

3.4.2 Virtual Real World

The line of sight intersection method, although suitable for a future implementation in the SST system, is limited to just displaying a marker on the screen. This method is independent of the type of objects on the other side of the screen. Additionally, the fact of displaying only a 2D marker over a 3D environment prevents the system from interacting with the objects of interest in a more personalized manner.

The method presented in this section appears as a response to the limitations pointed out in the previous method. Thus, it is proposed the development of a method that aims to display on the transparent screen images of a three-dimensional virtual world that exactly represents the real environment on the opposite side of the screen. The virtual environment will be fully modeled with the objects of interest, assuming that they are known *a priori*, as well as their dimensions. The position of the objects will be updated and defined by the coordinates obtained through the methods developed in the previous areas. The coordinates related to the position of the user's head will be used to establish the position and orientation of the virtual camera, which is responsible for capturing the images of the modeled environment that will be displayed on the screen (Figure 3.11).

This concept allows the final system to be sensitive to certain depth cues, which would never be perceptible by the Line of Sight Intersection (LSI) method. Object depth is estimated by depth cues in the human visual system. Among them: [129, 130, 131].

- Occlusion: For LSI method occlusion is not considered. Given the coordinates obtained from the user and the object, the marker will be displayed on the screen, even if from the user's point of

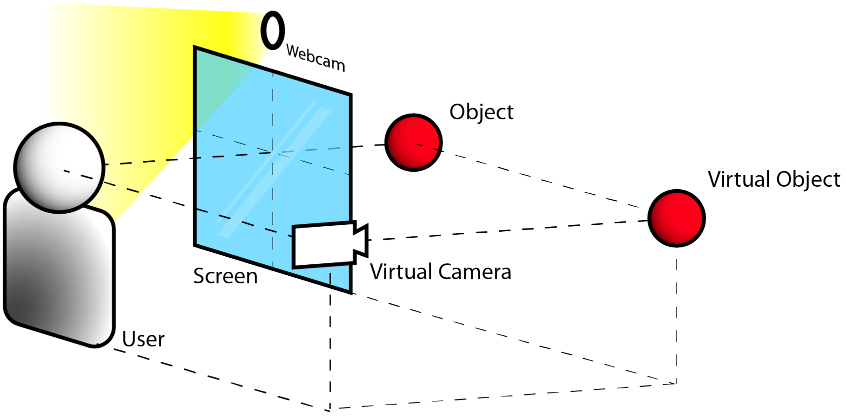


Figure 3.11: Representation of the Method Virtual Real World.

view the object of interest is hidden by an obstacle. In the present method of the Virtual Real World (VRW), the environment will be modeled so that, in the presence of an occlusion, from the user's point of view, the virtual camera will capture the image so that the visibility of the object of interest is obstructed. This effect is guaranteed in VRW because the virtual camera, ideally, captures exactly what the user sees in reality. It is important to remember that the user's head determines the orientation and position of the virtual camera in the modeled environment.

- Perspective: In LSI the virtual marker is the same regardless of the distance that the respective object is from the user. Perspective justifies the fact that an object appears smaller as it is further away and parallel lines give the illusion that they intersect at a vanishing point. By VRW, these visual effects are achievable, as the virtual environment itself has integrated its own perception of perspective.

- Geometry and detail: As noted in the previous point, in LSI, the marker is indifferent to the type of object it is dedicated to locate. Therefore, using VRW, it is possible to virtually model the object so that it coincides with the respective one behind the screen. The geometry of the object itself will be more detailed for when the object is closer to the user.

- Relative movement: Considering two objects at different distances, the user's head movement will allow him/her to perceive which one is farther away. The reason is that more distant objects appear to move more slowly. This relative movement in the LSI method will be noticeable only when two or more markers are displayed on the screen. On the other hand, in VRW, since the environment is modeled together with the object(s) of interest, it is possible to detect the relative movement more easily, using as a reference other elements present in the virtual scene.

There are other depth cues such as: atmosphere, luminosity, shadows, stereopsis, convergence and accommodation. None of these will be possible to verify in the SST system that it proposed to be developed, for reasons associated with the transparency of the screen. In particular, the last three cues require a separate analysis of each eye's point of view rather than bringing the field of view closer to a singular point. 2D images do not adequately convey these cues [132].

According to the last paragraph, there are two ways of analyzing the user's field of vision and presenting images on the transparent screen to generate the augmented reality effect. More specifically: Monoscopy and Stereoscopy (Figure 3.12).

In Stereoscopy, left and right images are rendered independently for each eye, each portraying its own perspective. In the virtual scene, each eye represents the respective physical position of a different camera.

Monoscopy is characterised by rendering a single image that aims to represent an estimate of the user's field of view. The image is rendered from the position established in the previous sections, which refers to the coordinates corresponding to the midpoint between the user's eyes, in the virtual modeled environment.

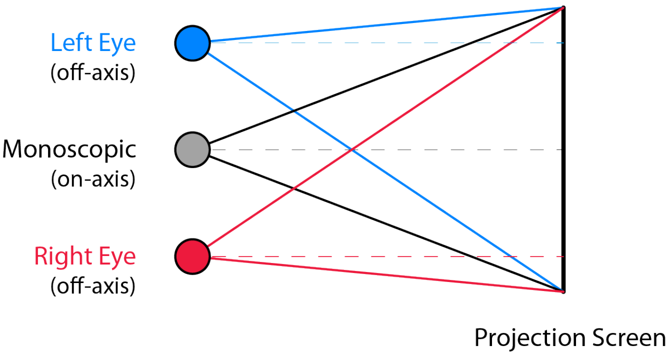


Figure 3.12: Stereoscopic view frustum vs. Monoscopic view frustum.

Furthermore, another major difference that distinguishes these two approaches is based on the type of parallax that can exist in each. Lillakas et al. [133] define parallax as follows:

> "Motion parallax refers to retinal image motion generated by head movements relative to stationary objects at different distances; the objects will be seen in depth and/or will appear to move, depending on fixation distance and the velocities of retinal image and head movements."[133]

This definition can be illustrated as well as three different types of parallax highlighted:

To interpret Figure 3.13, consider that, in the context in which this project is being developed, the objects of interest can be located not only on the opposite side of the user from the transparent screen (case A), but also between the user and the screen itself (case B). It is also possible to consider a third case, physically impossible, but theoretically relevant, in which the positions of the screen and the object coincide (case C).

Consequently [134]:

- Positive parallax corresponds to case A and is defined by the situations in which it is intended to represent points that are located behind the projection screen.
- Negative parallax corresponds to case B and is defined by the situations in which it is intended to represent points that are located in front of the projection screen.

- The null parallax corresponds to case C and is defined by the situations in which it is intended to represent points that are already located in the correct position and coincident with the projection screen.

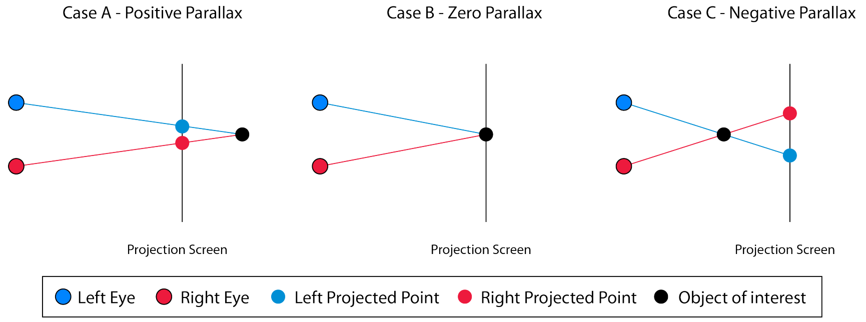


Figure 3.13: The images show points in a 3D scene being projected on the screen for the left and the right eye and their resulting parallaxes between the eye's images.

Resuming the distinction between monoscopic and stereoscopic, it is noteworthy that images rendered in monoscopic are considered to have always zero parallax. Hence, points are always perceived to be on the same plane as the screen. Stereo images have different parallaxes for virtual objects depending on their distance from the eyes. The parallaxes may also be affected by distance from screen and gap between eyes [132].

There are different methods intended to estimate the images meant to be rendered and displayed on the screen. For planar screens, there are three that are commonly used for augmented and virtual reality applications in stereo projections. Namely, the on-axis projection, off-axis projection and toe-in methods.

Since these methods are applied to stereoscopy, the implementation requires an individual analysis of each eye. In these analysis the eyes' position in relation to the screen, where the image specially rendered for each, is considered. However, for the context in which this project is inserted, it is not justified to carry out an individual analysis for each eye, so the estimated field of vision will be relative to the midpoint between the two eyes of the user. Therefore, in the next descriptions of the methods, references to the point of view relate to the previously defined midpoint.

On-axis projection (Figure 3.14) is a method that assumes a central position of the point of view in relation to the projection screen, regardless of its distance from it. That is, the point of view, projected orthogonally on the screen, will coincide with the geometric center of the rectangle that defines the area of the screen itself. In this process, a completely symmetrical pyramidal 'frustum' is established, where the rectangular edges of the screen define the base of the pyramid and the point of view establishes the vertex. The axis of the pyramid representing the field of view is perpendicular to the plane of the screen. The point that results from the orthogonal projection of the point of view on the plane itself is called the origin of the projection plane [135].

Off-axis projection (Figure 3.15) is a method that assumes any off-center position of the point of view in relation to the projection screen, regardless of its distance from it. In this case, the pyramidal 'frustum' is asymmetric and the orthogonal projection of the viewpoint no longer coincides with the geometric

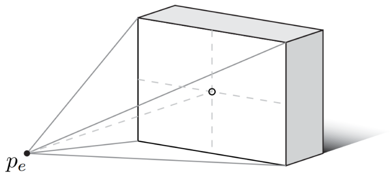


Figure 3.14: On-axis Projection. Image taken from Kooima [135].

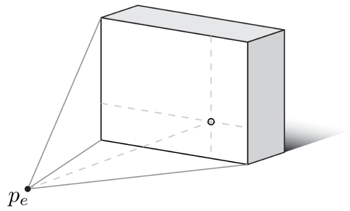


Figure 3.15: Off-axis Projection. Image taken from Kooima [135].

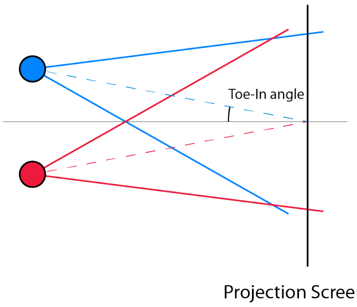


Figure 3.16: The Toe-in Method.

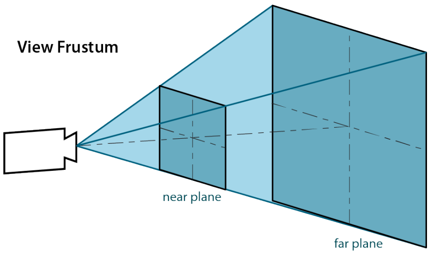


Figure 3.17: View Frustum.

center of the rectangle that defines the screen area. The axis of the pyramid representing the field of view is perpendicular to the plane of the screen. The origin of the projection plane is defined by the instantaneous position of the orthogonal projection of the point of view. Therefore, the movement of the user's head will result in a displacement of the point of view in relation to the screen [135].

Toe-in appears as a method resulting from the combination of the two previous ones. Toe-in assumes any position of the point of view, centered or not, in relation to the projection plane. The 'frustum', unlike the other methods, is defined by a pyramid whose axis will only be perpendicular to the projection plane when the point of view is central. This is justified by the fact that in this method the resulting 'frustum' is necessarily symmetric for all the positions of the point of view. For this to be possible, the base of the pyramid is no longer the rectangle that defines the area of the projection plane. The base is an imaginary plane that is perpendicular to the axis of the 'frustum' [132]. Similarly to the previous methods, the edges of the pyramid are defined by straight lines that join the point of view to the vertices of the rectangle that limits the screen. For this method it is considered that there is a rotation of the user's head in relation to the screen. The rotation is defined by the 'toe-in' angle.

'Frustum' is defined as the volume, usually in the shape of a pyramid or cone, which delimits the field of view corresponding to the image displayed on the projection screen [136] (Figure 3.17).

The method development in this section dedicated to Screen Output will be guided by the toe-in approach applied to the theory of monoscopy. However, toe-in is a process, considered by Oliver Kreylos [137] that gives incorrect results and causes discomfort to the user when compared to other methods. However, this is still a widely used method, mainly in VR systems [137] and it appears as an adequate solution when the rendering engine or library does not provide off-axis rendering capabilities [132]. For this reason it is considered a versatile and suitable option for future implementation in the SST system.

The theoretical development of the method begins. To start it is necessary to determine the angle of view, α, for the position of the point of view. The angle of view, both vertical and horizontal, will be fundamental to mathematically define the 'frustum'. The angle is defined by the segments that join the point of view to two adjacent vertices of the projection screen. Vertices that define a horizontal edge, characterise the horizontal angle of view. If the chosen vertices define a vertical edge, it characterises the vertical angle (Figure 3.18).

According to Tim Dobbert, in the context of photography, the angle of view defines the size, or extent, in angular terms of the scene viewed by a camera. [138].

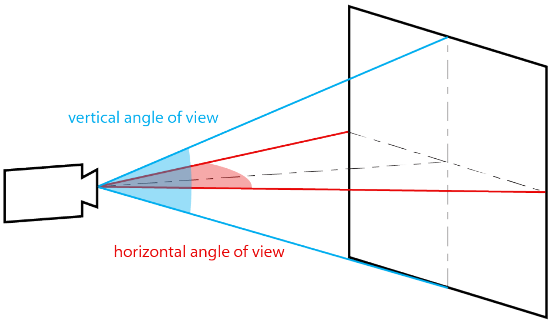


Figure 3.18: Vertical and Horizontal Angles of View

Consider the adjacent vertices $A\ (a_1, a_2)$ and $B\ (b_1, b_2)$, and mathematically, for determining the angle, it does not matter whether they define a vertical or horizontal edge. Also consider the viewpoint $U\ (x, y)$ (Figure 3.19).

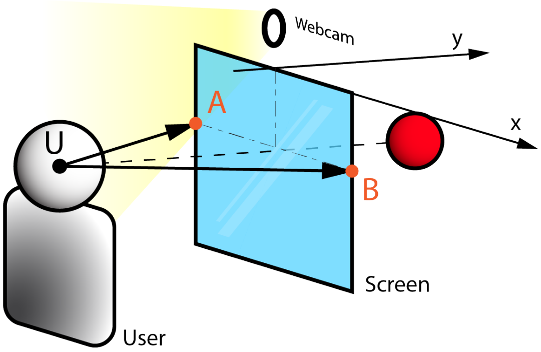


Figure 3.19: Definition of points A, B and U

The vector $\vec{v}\ (v_1, v_2)$ is defined as having an origin in U and the norm and orientation of the segment that joins the points U and A:

$$\begin{pmatrix} v_x \\ v_y \end{pmatrix} = \begin{pmatrix} a_1 \\ a_2 \end{pmatrix} - \begin{pmatrix} x \\ y \end{pmatrix} \tag{3.35}$$

The vector $\vec{w}\ (w_1, w_2)$ is defined as having an origin in U and the norm and orientation of the segment that joins the points U and B:

$$\begin{pmatrix} w_x \\ w_y \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} - \begin{pmatrix} x \\ y \end{pmatrix} \tag{3.36}$$

The angle defined by vectors $\vec{v}$ and $\vec{w}$ corresponds exactly to the angle of view α (Figure 3.20).

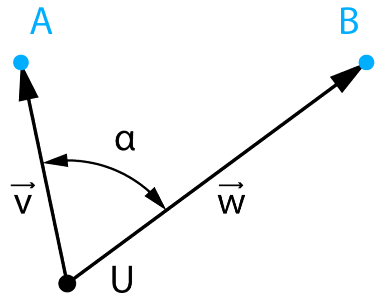


Figure 3.20: Angle of view definition

By the dot product between vectors it is possible to obtain $cos(\alpha)$ as follows:

$$cos(\alpha) = \frac{\vec{v} \cdot \vec{w}}{\|\vec{v}\| \cdot \|\vec{w}\|} \tag{3.37}$$

The dot product between $\vec{v}$ and $\vec{w}$ can be determined as follows:

$$\vec{v} \cdot \vec{w} = (a_1 - x)\,(b_1 - x) + (a_2 - y)\,(b_2 - y) \tag{3.38}$$

The norms of vectors $\vec{v}$ and $\vec{w}$ are given by:

$$\|\vec{v}\| = \sqrt{(a_1 - x)^2 + (a_2 - y)^2} \tag{3.39}$$

$$\|\vec{w}\| = \sqrt{(b_1 - x)^2 + (b_2 - y)^2} \tag{3.40}$$

Developing, we obtain the following general expression that allows us to calculate the $cos(\alpha)$ as a function of the position (x, y) of the point of view (Figure 3.22):

$$cos(\alpha) = \frac{a_1 b_1 - x(a_1 + b_1) + x^2 + a_2 b_2 - y(a_2 + b_2) + y^2}{\sqrt{a_1^2 - 2a_1 x + a_2^2 - 2a_2 y + y^2} \cdot \sqrt{b_1^2 - 2b_1 x + b_2^2 - 2b_2 y + y^2}} \tag{3.41}$$

Given the complexity of the previous expression, a simplification, in which the dimension of $\overline{AB}$ is considered to be unitary, was applied as follows (Figure 3.21:

$$(a_1, a_2) = (0, 0) \qquad \text{and} \qquad (b_1, b_2) = (1, 0)$$

Therefore, it is necessary to calculate the relative coordinates of the point of view, adequate to the implemented simplification:

$$x_R = \frac{x + \frac{L}{2}}{L} = \frac{x}{L} + \frac{1}{2} \tag{3.42}$$

$$y_R = \frac{y + \frac{L}{2}}{L} = \frac{y}{L} + \frac{1}{2} \tag{3.43}$$

Where L is the screen's Width or Height, for the Top-view or Side-view, respectively.

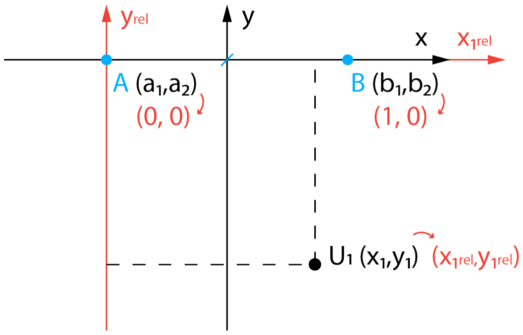


Figure 3.21: Relative coordinates from Simplification

Substituting, the following expression is obtained:

$$cos(\alpha) = \frac{-x_R + \ x_R^2 + \ y_R^2}{\sqrt{x_R^2 - \ 2x_R^3 + \ x_R^4 + \ y_R^2 x_R^2 + \ y_R^2 - 2x_R y_R^2 + \ x_R^2 y_R^2 + y_R^4}} \tag{3.44}$$

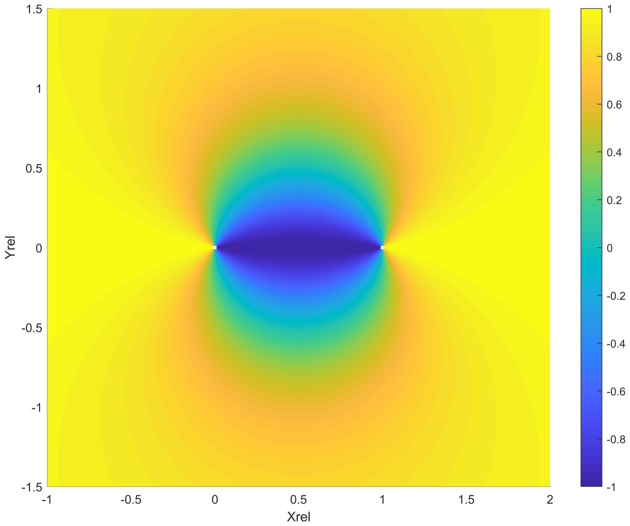


Figure 3.22: Cosine of the angle of view as a function of the relative coordinates of the user's position.

Then it is necessary to determine the 'toe-in' angle that will be represented by the camera's rotation angle γ. This will also take into consideration the simplification previously implemented.(Figure 3.23)

The angle γ is defined by the following relationship between the angle of view, α, and the angle θ:

$$\gamma = \ \frac{\alpha}{2} - \ \theta \tag{3.45}$$

where θ is the angle defined between vectors $\vec{v}$ and $U\vec{U}'$.

vector $U\vec{U}'$ originates from the point of view U and the norm and orientation of the segment joining point U to U'.

The angle θ is obtained:

$$tan(\theta) = \ \frac{x_R}{y_R} \Rightarrow \ arctan\left(\frac{x_R}{y_R}\right) = \ \theta \tag{3.46}$$

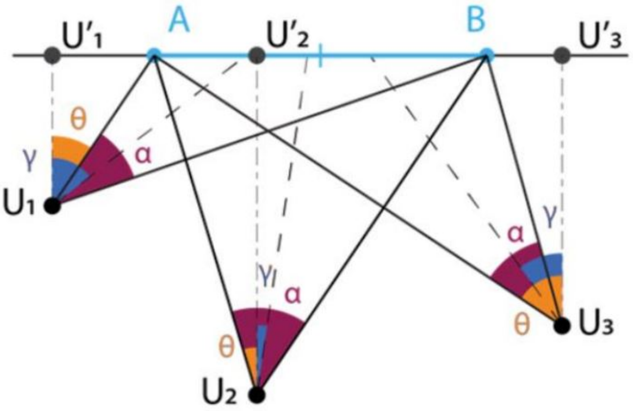


Figure 3.23: Toe-in, Camera and View Angles for different positions of the user in space.

Note that for the equations presented it is considered that rotations clockwise, in relation to the vertical axis, result in positive angles. Consequently, counterclockwise rotations, relative to the vertical axis, result in negative angles. (Figure 3.24)

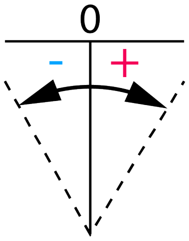


Figure 3.24: Camera rotations clockwise, in relation to the vertical axis, result in positive angles. Counterclockwise rotations, relative to the vertical axis, result in negative angles.

By this method, the angles of view α, and toe-in γ are calculated, namely the four that refer to the respective horizontal and vertical angles of each. Assuming all the necessary parameters to implement the method in the project, as will be described in the following chapters.

Chapter 4

Implementation and Results

The purpose of this chapter is to explain how the chosen methods and algorithms were implemented during the development of a prototype that achieves the proof of concept of the envisioned project. The chapter is divided into three sections, the first two of which discuss two different approaches to project development, based on the methods from Chapter 3. In these two sections, the procedures followed and the decisions made are described. Finally, the last section focuses on a detailed analysis of the results obtained with the developed prototype. The presented results come from theoretical evaluations, which aim to predict performance problems associated with errors and deviations in the presentation of elements on the screen, and from observations and interactions with the system that allow relevant conclusions to be drawn.

In order to assess the reliability of the approaches that were developed, a transparent monitor was required to project the virtual content generated by the developed programs. Another requirement was that a camera should be integrated into the transparent monitor plane. Lastly, a computer should be set up to communicate constantly and simultaneously with the camera and screen.

Currently, a transparent OLED monitor is expensive and the costs associated with the acquisition of this type of device are not justified for the proof of concept behind this project. Thus, the alternative of using a conventional laptop and adapting it to the project context emerged. Descriptively, laptops donated to the project were used to establish the particularity of having a transparent monitor. With this solution, the developed programs would be executed directly on the laptop and projected on the respective monitor. Image capture would be performed through an external or internal webcam depending on the characteristics of the laptop used.

To implement the solution idealized in the previous paragraph, we proceeded with the task of disassembling the monitor from the laptop so that it was possible to remove the plastic cover and frame that surrounds the LCD. The first complete success of this operation was achieved on a computer without an integrated webcam (Figure 4.2). Transparency was achieved simultaneously with the usual functioning of the computer, however some disadvantages were highlighted:

- Without the support or frame that protected the LCD, the laptop was exposed to the outside environment and was weakened by any rotational movements of the screen.

- The laptop was left with the electrical wires too exposed, which could present some danger during its use.
- The view through the monitor was partially obstructed by the flat cable connecting to the LCD command printed circuit (CCFL).
- The polarizing filters present were responsible for not allowing the image of the environment behind the monitor to be completely clear.

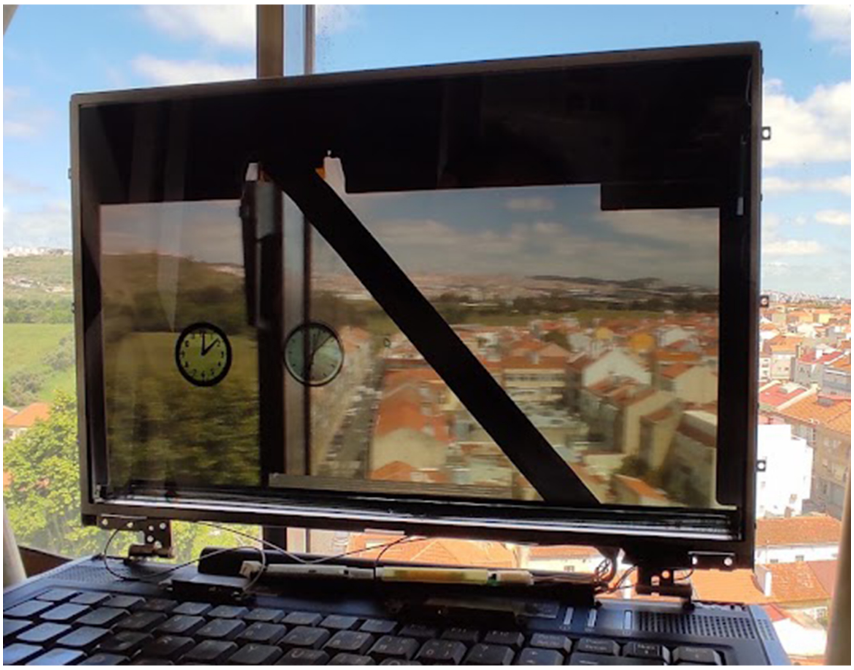


Figure 4.1: First Laptop successfully adapted into a transparent monitor .

The same operation could still be successfully reproduced on another laptop (Figure **??**). For this, the results were different mainly due to the characteristics of the laptop itself. The list of disadvantages presented above has been shortened. Namely:

- The monitor incorporated a metal frame that covered a narrow frame of the monitor area.
- The electrical cables were not exposed and the rotation movements of the screen itself did not compromise the laptop's resistance.
- The flat cable connecting the LCD command printed circuit was not installed across the screen.
- The polarizing filters on this laptop had the same impact on image clearness as was visible through the monitor.
- The laptop integrated an internal webcam that remained functional after the monitor's transformation processes.

The characteristics of the second laptop described were identical to those of the computer used during the tests, which resulted in the Exponential Fit method, developed in Chapter 3, in the Depth Estimation section.

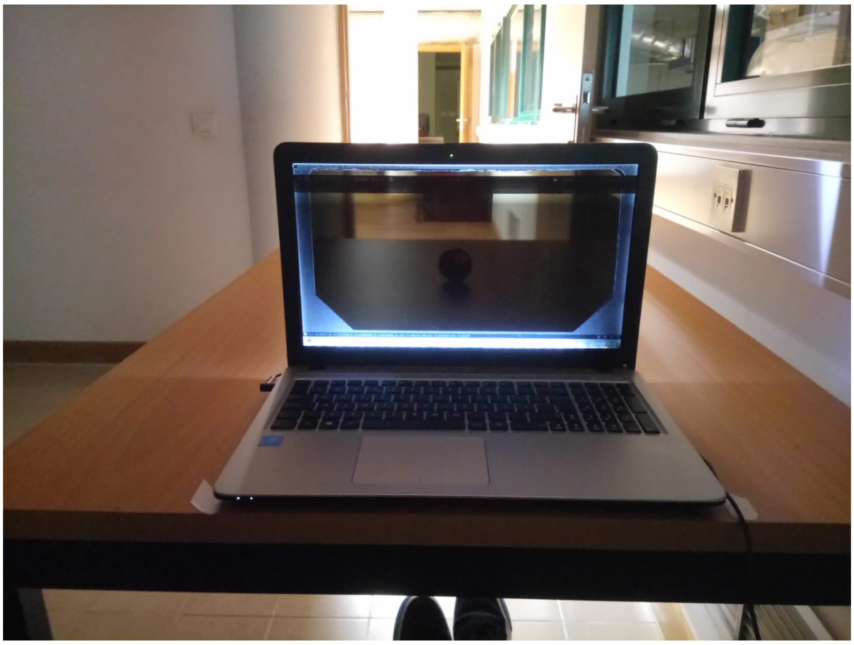


Figure 4.2: First Laptop successfully adapted into a transparent monitor.

Regarding the features of the screen:

- Projection area dimensions: 34.4 X 19.2 [cm]
- It was ensured that the monitor plane was oriented perpendicular to the keyboard plane and, consequently, to the plane where the laptop was settled.

Regarding webcam features:

- Resolution: 1280 X 720 [px] and 0.9 MP
- The webcam is located centered with the screen's vertical axis of symmetry, 0.8 cm away from the top edge
- Horizontal capture angle: 79.42 ° (obtained empirically)
- Vertical capture angle: 50.27 ° (obtained empirically)

4.1 Line of Sight-based Approach

The first approach implemented consists of a synergistic combination of three methods, respective to each of the three areas presented in the previous chapter. For the Computer Vision area, Face Tracking was chosen, namely the Viola Jones, AdaBoost and KLT algorithms. These were applied sequentially and in a structured way, exactly as described in the respective method section. The Exponential Fit method developed and presented in the section dedicated to Depth Estimation was immediately recognized as the preferred method to integrate the system due to its simplicity and effectiveness. Finally, for this approach, the Screen Output area method to be implemented will be the Line of Sight Intersection, as the title of this section suggested.

The methods identified in the previous paragraph were implemented with the support of *MATLAB*. This software is not the most suitable for computer vision applications, compared to programming languages like C and derivatives. This fact is logical because *MATLAB* is an interpreter, therefore it has slower latency and processing times. Executions in C and derivatives are absolutely faster, or even eventually in Python. However, any of them implies a more time-consuming and demanding implementation than *MATLAB* [139]. So the interpreter was chosen, as it allows to conduct the proof of concept successfully from algorithms more elementary, intrinsic to the software [140]. *Simulink* also presents itself as a useful tool, however, it was not considered in this project for presenting less satisfactory results in research works with face tracking applications, similar to what is required in this project [139].

Before the methods could be implemented, it was necessary to proceed with the installation of the following toolbox: $MATLAB\ Support\ Package\ for\ USB\ Webcams$; $Computer\ Vision\ System\ Toolbox$ [141].

- $MATLAB\ Support\ Package\ for\ USB\ Webcams$ allows importing live images from the webcam built into the laptop you are using into *MATLAB*;
- $Computer\ Vision\ System\ Toolbox$ has algorithms, functions, and apps to facilitate the design and simulation of computer vision and video processing systems. With this toolbox it becomes possible to detect, extract, and match features, as well as detect and track specific objects.

The method for detecting and tracking was explained and outlined in Chapter 3. The description previously presented (Figure 3.5) dictates the development of the created *MATLAB* code. The above mentioned toolbox functions were used as a way to take advantage of the algorithms previously developed and integrated into the software, and $vision.CascadeObjectDetector$ was specifically used for face and eye detection in *MATLAB*. This function was specifically designed to apply the Viola-Jones algorithm to detect objects.

To continue with the description of the implementation of the methods in this first approach, it will be assumed that the set of algorithms is in a phase where the detection of the face and eyes was successful and that a number of feature points greater than the minimum are identified. To clarify, during the development of the work it was established that the face detection method would need to control the position of at least 10 feature points. Whenever this minimum was guaranteed, it would not be necessary to proceed with a re-detecting of the user's face to renew and insert in the sequence of tracking more and new feature points.

When an object is detected in *MATLAB*, a region of interest (RoI) is defined from which feature points are identified, through the $detectMinEigenFeatures$ function. The region of interest is limited by the area of a rectangle defined by a 1x4 matrix:

$$RoI_{Rectangle} \rightarrow [x \quad y \quad w \quad h] \tag{4.1}$$

where (x, y) correspond to the coordinates of the upper left vertex of the rectangle, in the extracted frame reference, in pixels; (w, h) refer respectively to the width and height of the rectangle, in pixels (Figure 4.3).

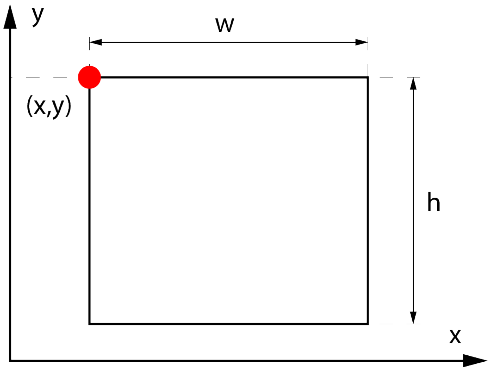


Figure 4.3: Representation of RoI rectangle's coordinates.

It becomes necessary to extract this matrix for each frame and convert it to a 4 x 2 matrix, where each row of the matrix corresponds to the coordinates of a vertex of the rectangle that defines the RoI:

$$RoI_{Rectangle} \rightarrow \begin{pmatrix} x_1 & y_1 \\ x_2 & y_2 \\ x_3 & y_3 \\ x_4 & y_4 \end{pmatrix} \tag{4.2}$$

where $(x_i,\ y_i)$ correspond to the coordinates of vertices i.

The rectangle is then presented superimposed on each of the respective frames, where its orientation may imply that its edges are neither horizontal nor vertical, if the estimated geometric transformation of the feature points ($estimateGeometricTransform$ function) justifies it. These cases correspond to times when the user tilts his/her head.

We also proceeded with the RoI obtained from the detection of the user's eyes, therefore, from this first phase of the implementation of the methods, two matrices result, one for the face rectangle and the other for the eyes rectangle. Only the face rectangle is overlaid in the captured frames.

After obtaining both matrices, the respective coordinates are inserted in the *MATLAB* function created to estimate the three-dimensional coordinates of the user's point of view in relation to the camera. In this function that has been developed, the width of the face rectangle, in pixels, is considered as corresponding to the width of the user's head. Following the procedure aforementioned and explained in Chapter 3, area of Depth Estimation, the real dimension of the user's width is required, so it was considered an average value of the width of the human head. The value used resulted from an average between the male and female average, namely 15.5 cm [142]. It was not necessary to adapt the regression previously obtained due to the conditions of the used laptop's webcam.

Thus, the estimated distance from the user to the camera is obtained by solving the equation that is presented:

$$\frac{w}{15.5} = 41.12 \cdot e^{-0.0351 \cdot x} + \ 11.17 \cdot e^{-0.005246 \cdot x} \tag{4.3}$$

where w is the width of the face rectangle, in pixels.

To estimate the remaining spatial coordinates of the user's point of view, referring to the vertical and horizontal displacement of the head in relation to the camera position, the centroid of the rectangle of the eyes is considered (Figure 4.4). Hence, the vertical and horizontal deviation of the centroid from the center of the frame is calculated. This point corresponds to the point that was initially defined in this project as the equivalent point of view to estimate the user's field of view.

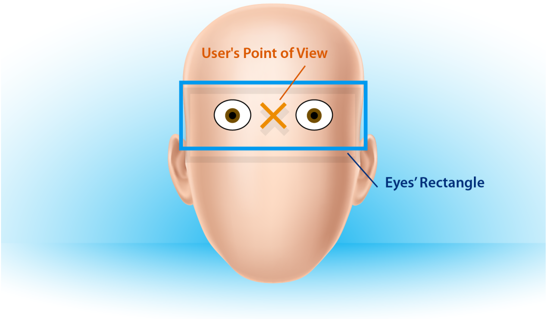


Figure 4.4: Point of View represented by the centroid of the eyes' RoI rectangle.

After estimating the coordinates of the user's point of view, it becomes necessary to define the line of sight relative to an object of interest. During this chapter aimed at developing the implementation of the methods, a fixed and arbitrary position for the object of interest was considered (Figure 4.6). Specifically, it was stipulated that the object was located at a longitudinal distance of 50 centimeters from the camera, with a negative vertical deviation of 20 centimeters from the center of the camera and without any lateral deviation. In coordinates, the described position corresponds to $P = (50, 0, -20)$. Note that the semi-positive axis of the x-axis is considered to originate in the center of the camera and oriented to the opposite side of the user. It is understood that the estimated abscissa from the user's point of view will always be negative (Figures 4.5).

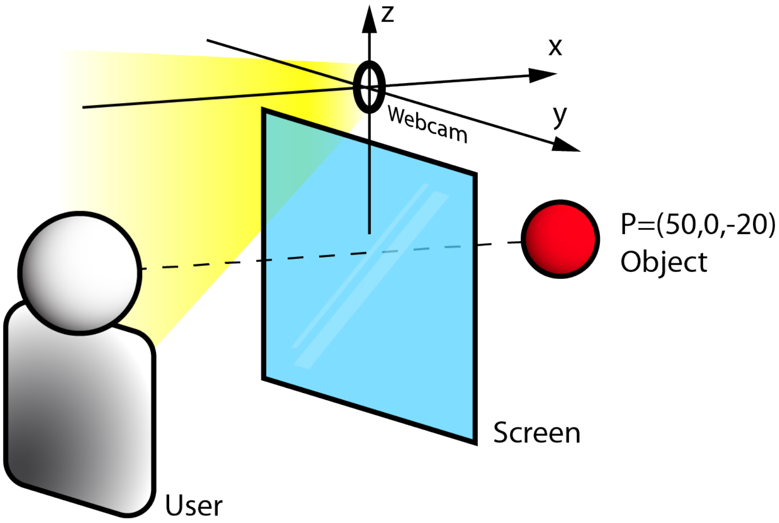


Figure 4.5: Fixed and Arbitrary position for the object of interest assumed for the results.

Both coordinates of the two points in space (user and object) are inserted in a new function that

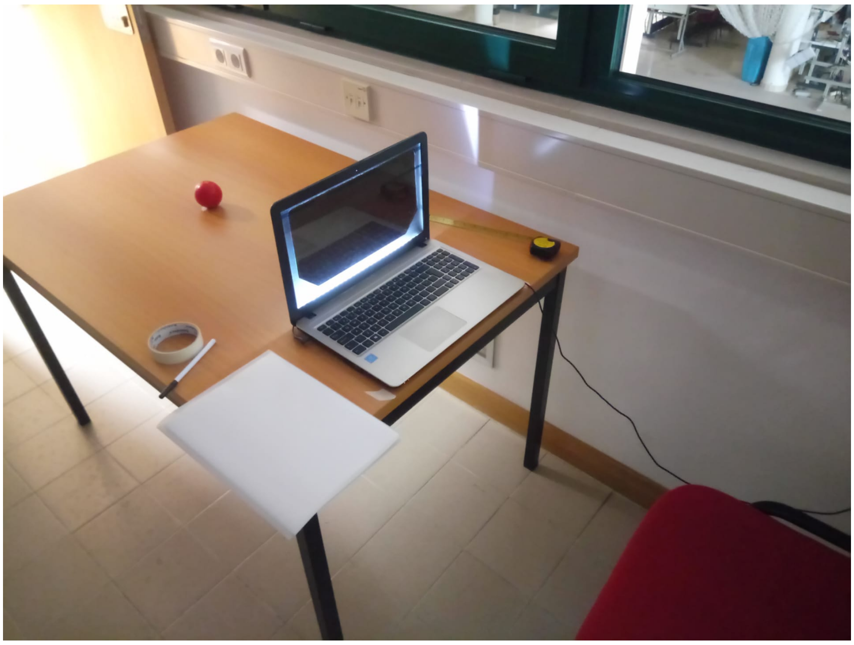


Figure 4.6: First Laptop successfully adapted into a transparent monitor .

calculates the respective line of sight and intersects it with the screen plane, previously defined. The screen plane for the considered reference is defined by $x = 0$, as previously stated.

By calling this function, you will be able to determine the coordinates, in centimeters, of a single point that will correspond to the real position of the marker to be represented on the screen. However, the centimeter coordinates of the virtual marker are not useful by themselves. It is crucial to convert them into virtual coordinates, in pixels, that take into account the dimensions and resolution of the screen where the marker will be displayed (Figure 4.8). Due to this requirement, a final function was created to check if the intersection point is visible on the screen and, if so, what its coordinates are on the screen. For this it is necessary to have defined the position of the screen in relation to the camera and the area and display resolution in which the marker can be rendered (Figure 4.7).

For the points visible on the screen, the following coordinate conversion was carried out:

$$Mv_y = \begin{cases} Q_y \cdot \frac{-100}{E}, & if \quad Q_y < \ 0 \\ Q_y \cdot \frac{100}{D}, & if \quad Q_y > \ 0 \end{cases} \tag{4.4}$$

$$Mv_z = (Q_z - \ C) \cdot \frac{-100}{B - C} \tag{4.5}$$

where, $Q = \ (Q_x, \ Q_y, \ Q_z)$ represents the user's point of view real coordinates, in centimeters;
$Mv = (Mv_x, \ Mv_y, \ Mv_z)$ represents the marker's virtual coordinates on the screen;
$B, \ C, \ D$ and E are the points that define the screen's position relative do the camera (Figure 4.8).

The coordinates of the four points are:

$B = \ (0, \ -17.4);$

$C = \ (0, \ -4.05);$

$D = \ (13.9, \ 0);$

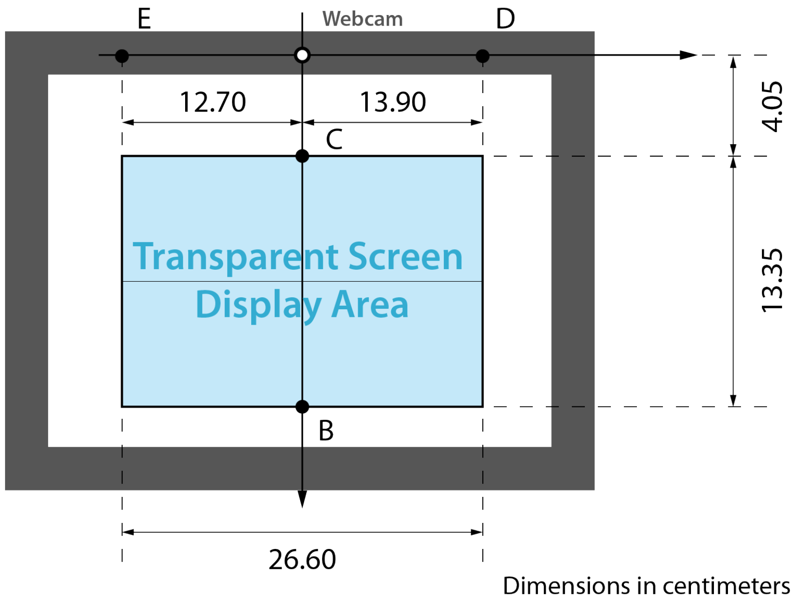


Figure 4.7: Position of the screen display area in relation to the webcam in *MATLAB*.

$E = \ (-12.7, \ 0).$

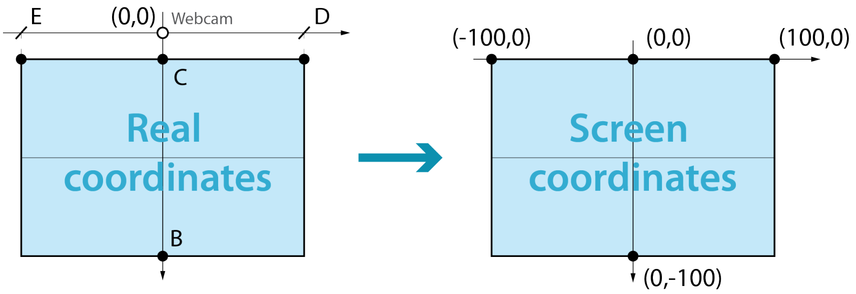


Figure 4.8: Conversion from real coordinates into screen virtual coordinates.

The virtual marker is then displayed on the screen. All the steps described in the previous paragraphs are repeated for each frame captured by the camera, when a successful face tracking is achieved, with the minimum amount of feature points (Figure 4.9).

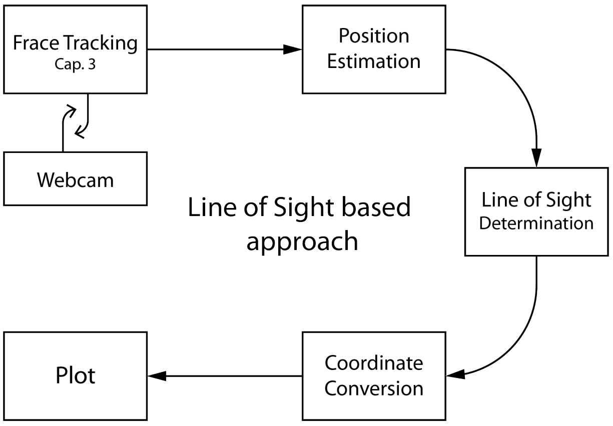


Figure 4.9: Schematization of the Line of Sight based approach.

4.2 3D Monoscopy-based Approach

The second approach also involves three methods, respectively from each area of the previous section. The big distinction between the first and second approaches developed is the method chosen for Screen Output. Both use Viola Jones, AdaBoost and KLT algorithms, for the area of Computer Vision, and also use the Exponential Fit method, to estimate the depth of objects captured by the camera. This approach implements the Virtual Real World method.

Although the first two methods are common to the first approach, the modes in which they were implemented for the 3D Monoscopy-based approach are different. What distinguishes the way these methods were implemented differently is mainly the support software used to render the three-dimensional modeled environment on the transparent screen. We chose to use Unity 3D, due to its success in other research works reviewed dedicated to the exploration of augmented reality technologies [143, 144] and the number of libraries available that aim to support computer vision and augmented reality applications. Unity 3D's user-friendly interface and tools are additionally an advantage that was highlighted which would make the implementation more straightforward. The use of Unity 3D implies that the code is written in C#, also allowing the latency and processing times to be faster, as already mentioned. Predicting that the real-time operation of the system will be successful.

The Unity 3D is a cross-platform 3D game engine developed by *Unity Technologies Co.Ltd*. Games and interactive 3D content can be created with ease using its development box. Unity 3D can append almost any type of material, scenario, sound, and animated video to a virtual environment. Moreover, it can release stand-alone executables on many platforms, such as *Windows, iOS, Android* [145].

Before starting the development of C# codes to implement the methods, it was necessary to import a support library, *OpenCv+Unity*, allowing integration of tools for computer vision applications into Unity 3D.

OpenCV (Open Source Computer Vision Library) is a cross-platform developed by Intel. The open-source library provides a variety of programming functions designed for real-time computer vision and image processing [146].

In Unity 3D each program includes a three-dimensional environment prepared to develop the projects. In this environment, a virtual camera is integrated. The image captured by the camera matches the image displayed on the screen exactly. The camera is programmable and its position, orientation and field of view are editable. After importing the OpenCV library, the respective Face Tracking algorithms are implemented as follows:

- A Game Object is added in the virtual environment on which is attached to a C# script that allows programming the Game Object's behavior during the execution of the program. A Game Object is the base class for all entities in Unity Scene.
- The script will be responsible for communicating with the computer's real webcam to capture the real surrounding environment. In this context, the chosen Game Object was a plane so that the image captured by the real webcam is projected on the plane itself and consequently is visible when the final program is running.

- On the script a pre-trained cascade classifier, provided by OpenCV is declared, similarly to what was done with *MATLAB*.
- The face tracking method is developed in the Game Object script exactly as it was done in the previous approach, using *MATLAB*, as described in Chapter 3.
- Thus, the plan is programmed to present the image captured by the webcam, where, for a detected face, the user's point of view is identified and a rectangle delimiting the RoI is represented.

From the implementation of this method it's possible to obtain, for each frame in which a face was successfully detected, the coordinates of the midpoint between the eyes, in pixels. The width of the rectangle delimiting the RoI is also obtained, corresponding to the approximate width of the user's head (Figure 4.10).

Figure 4.10: Approximate width of the user's head.

It proceeds with the estimation of the coordinates of the point of view in space. The method is exactly the same as used in the Line of Sight-based approach, but for the 3D Monoscopy-based a Dynamic-link library (DLL) was used.

In the words of Deland Han [147]:

> "A DLL is a library that contains code and data that can be used by more than one program at the same time.The use of DLLs helps promote modularization of code, code reuse, efficient memory usage, and reduced disk space. So, the operating system and the programs load faster, run faster, and take less disk space on the computer."

The DLL was compiled with the support of Visual Studio 2019, which facilitates the operation of the SST in real-time. It is also a viable option in future applications where it is necessary to calculate the position of several objects in relation to the camera simultaneously, without compromising the response time.

The DLL file was programmed to estimate the user's depth, lateral and vertical deviations using the same algorithm presented above.

The environment generated by Unity is modeled in such a way that each unit of virtual length corresponds to 1 meter in reality (1 Unity unit = 1 meter (100cm)) [148]. This way, it is possible to model the real environment, in Unity, with the real coordinates and real dimensions, in meters. Therefore, the object of interest considered fixed and of the arbitrary position will be modeled with its respective dimensions and in the position corresponding to the arbitrated coordinates, without having to resort to any type of unit conversion. Likewise, the user's point of view will be represented in the coordinates obtained from the DLL file.

The point of view will be represented by the virtual camera, whose captured virtual image is equivalent to the real image seen by the user, through the transparent screen. However, it is intended to limit the image captured by the virtual camera. The aim will be for Unity to render on screen only the user sees through the transparent screen itself, not the complete field of view of the user. Thus, it becomes essential to adjust the parameters of the virtual camera, according to what was developed in the Virtual Real World method. The parameters to be determined for each captured frame are: The angles of view α, which define the field of view of the camera; The toe-in angles γ, which define the angles of rotation of the camera, in relation to the vertical and horizontal axes.

To determine these angles, other DLL file was compiled. The algorithm executed in this DLL corresponds exactly to the calculation process developed in Chapter 3. The calculation of the angles depends on the dimensions of the rendering area of the running program. Hence, it had to be measured, since it is not the same as *MATLAB*'s, presented in the previous approach. The measurements are shown in Figure 4.11.

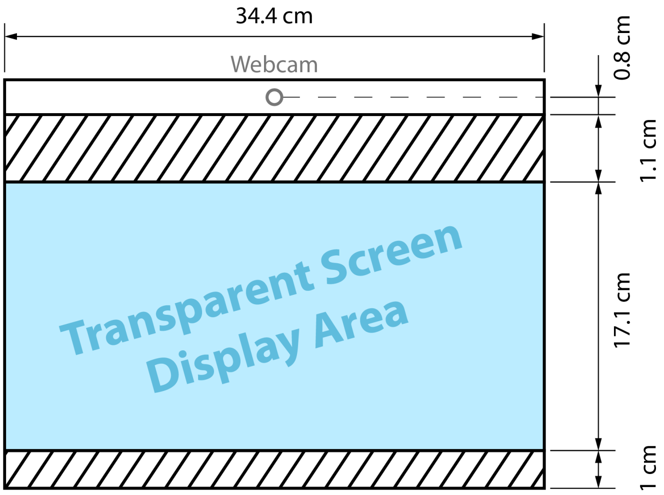


Figure 4.11: Dimensions of the rendering display area on *Unity3D*.

Once all the parameters to be updated in the properties of the virtual camera have been determined, a C# script is added to the camera allowing it to program its behavior during the execution of the program. This script will define the position of the camera, through the coordinates of the user's point of view, as well as its orientation, through the γ angles of rotation. Finally, the viewing angles will be responsible for adjusting the camera's field of view depending on its position.

The same can also be performed for an object of interest if it's considered that the object moves while using the SST system. The object's displacements would be captured by a webcam and its coordinates later estimated, in the same way as with the user's point of view. Attaching the C# script to the modeled object of interest, in the Unity environment, would allow updating its position in real-time.

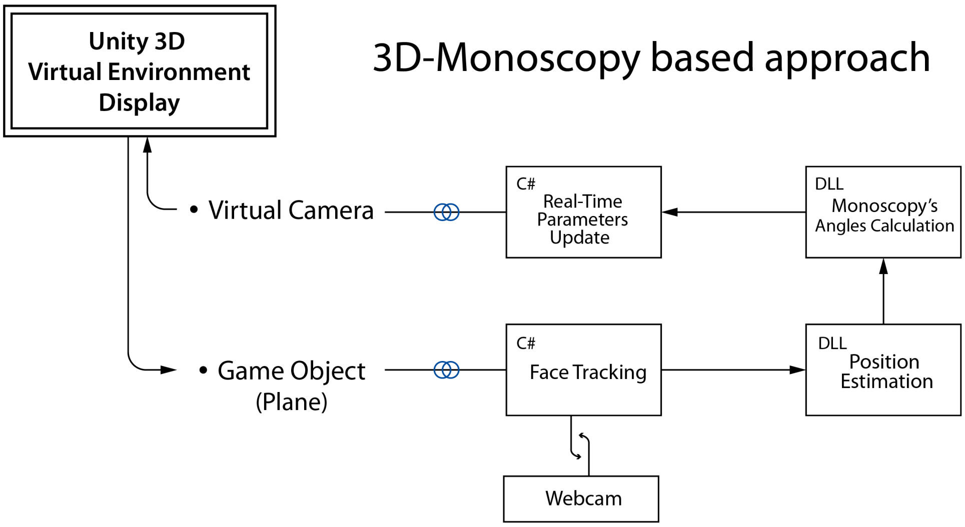


Figure 4.12: Schematization of the 3D Monoscopy based approach.

Within this approach (Figure 4.12), an attempt was made to implement the three methods on a Server/Client configuration established between *MATLAB* and Unity 3D. This attempt was aimed at making constant communication between the two software. *MATLAB* was responsible for detecting and tracking the user's face, as well as estimating the coordinates of the point of view, relying on the work developed during the Line of Sight-based approach. The parameters to be updated in Unity 3D's virtual camera properties would be determined in *MATLAB* too. *MATLAB* was defined as client, and Unity 3D as server, to achieve this goal.

The attempt was eventually discarded because it was a demanding process at a computational level. This fact was evidenced by the inability to update more than one parameter in the camera properties. This communication solution would not be viable in a system operating in real time due to the response delay.

4.3 Results

In this section, the results obtained with both approaches for the implementation of the methods in the development of the SST system are presented and discussed. When appropriate, comparisons will be made between the approaches. Note that no tests were carried out with other participants to assess the prototype's acceptance and performance. The results presented here are based on theoretical evaluations of the SST system, according to the methods that were used. The evaluations aim to predict performance problems associated with errors and deviations in the presentation of the content on the screen in relation to the environment and objects that are behind the transparent screen. Additionally, results from observations and interactions with the system that allowed relevant conclusions to be drawn will be discussed.

Regarding speed of face detection and tracking, in comparison between *MATLAB* and OpenCV+Unity, for brief moments of interaction with both it is possible to draw conclusions. The implementation in *MATLAB*, as foreseen in the previous sections, presents a slower response time. There is an obvious delay between the user's movement and the result on the screen. In face tracking implemented with OpenCv, this lag is subtle, almost imperceptible.

Due to their low complexity compared to Face Tracking, Depth Estimation and Screen Output methods in both approaches are not considered computationally demanding. Accordingly, it is reasonable to assume that the lag referred to in the above sentences primarily reflects the time it takes to retrieve and process information from the webcam.

According to Goyal et al. [140], these observations are also justified by the fact that *MATLAB*'s code is derived from Java, which is itself derived from C. For this reason the computer interprets the *MATLAB* code and converts it to Java before executing it. While OpenCV+Unity relies on C# library functions. The computer is directly provided with machine language code through these library functions. As a result, less time is spent on interpreting the information captured by the webcam, but rather on processing it.

When it comes to the stability of facial detection, it is possible to observe a flicker that has an influence on what is displayed on the screen. Flicker causes a constant, small and quick movement in the image projected on the screen. This effect was more evident in the Unity 3D implementation than in *MATLAB*, even when the user's head remained still. Instability was more noticeable for complex background environments, where OpenCV occasionally oscillated between identifying the user's face and an object in the background. The lag present in *MATLAB* was enough to avoid most of the flicker.

The visual instability in images that are presented to the user is responsible for generating any distractions or dizziness. This leads to another relevant discussion associated with cybersickness. Cybersickness is a term used to describe the symptoms that affect users who continually operate with augmented reality and virtual reality systems. Although the SST system has not been tested for long periods of time, it is possible that some symptoms may appear for those who intend to interact with the system continuously. Headaches, eye fatigue, nausea are highlighted and even epileptic seizures may eventually occur.

The indicators that may contribute to these symptoms, in addition to those already mentioned, are associated with the particularity of the screen's transparency, where the images are presented. There is a visual accommodation problem in the SST system. When focusing at a particular depth the eye needs to exert muscle tension to change its focal length [132]. In this prototype it is necessary for the user to alternately focus between the depth at which the object of interest is, behind the screen, and the depth at which the screen itself is. As the eyes have to constantly struggle to focus on elements of varying depths, some of the above symptoms may arise.

There are also other common symptoms associated with cybersickness such as vertigo or disorientation. These examples do not fit in the present context, but rather in systems that involve stereoscopic displays or devices that have the screen close to the user's eyes, as is the case with HMDs.

Finally, the SST system is evaluated in relation to the level of restrictions that the user is subject to during the interaction with the prototype. This is a factor that has a very low, almost nil, influence on the project that has been developed. The user is not inhibited from his natural perception of the environment that surrounds him, nor is he restricted from performing any type of movement. This is because the only requirement for the proper functioning of the system is that the user stays within the limits of the webcam's field of view. Note that the webcam is located in relation to the transparent screen in such a way that, if the user leaves the field of view of the webcam, it will not be so easy to observe the content displayed on the screen.

The biggest intrusiveness identified in the SST system is screen transparency. The screen is not completely transparent due to the polarising filters present influencing the visibility of objects behind it. One way to counteract the slight opacity of the screen is to have the brightness higher behind the screen than in front. In the presence of well-lit objects of interest, the opacity of the screen does not have as much influence on the user's perception.

The colors used in the display of content on the screen have a significant impact on the screen's opacity, regardless of the brightness level. Colors in the darker range, such as black, tend to hide the transparency of the screen, while light colors tend to increase the transparency of the screen, like white.

The lightning adjusting is only required since the screen used was adapted for the project and not designed for the purpose. A transparent OLED screen does not present adversities associated with luminosity as accentuated as those that were witnessed during the evaluations carried out in the laboratory. Regarding the influence of the colors of the content projected on the screen, in a transparent OLED screen the color's interference with opacity is not as significant. It is also verified that in this type of screen, darker colors contribute to the screen's transparency being more noticeable than lighter colors, as opposed to what was observed with the adapted screen.

From the interactions carried out with the developed SST system, some errors were detected between the real objects behind the screen and the virtual model projected on the screen itself. These mismatches were mainly evidenced in the second approach since for the first approach the visible marker is independent of the object's dimension and its exact position on the screen is determined for any position of the user.

In the 3D-Monoscopy approach, the digital content is projected onto the screen for a ratio of linear similarity between what is captured by the field of view and the dimensions of the screen itself. In other words, in contrast to what happens in the Line of Sight approach, the coordinates of each pixel displayed on the screen are not determined to exactly match the respective element of the surrounding environment, from the user's perspective. Thus, for user positions not exactly centered on the screen, regardless of its longitudinal distance, there are deviations between the position of the real object and its three-dimensional model on the screen.

These deviations were evaluated. To proceed with the evaluation, the importance of interpreting the results obtained by the depth estimation method developed is highlighted. By interpreting the results that express the effectiveness of the Exponential Fit method, it is possible to ascertain its influence on the detected deviations. Hence, the Residual Standard Error (S) was chosen to be interpreted. It will be restricted to the method with the double exponential regression, since the simple exponential regression has not been implemented.

$$S = \ 0.081701\, px/cm \tag{4.6}$$

The residual standard deviation is the average difference between the real and predicted values. Concretely, this value, in the present context, represents that for a given longitudinal distance, in cm, estimated by this regression, it is predicted that the object in the image captured by the camera has a ± 0.082 pixel difference, on average, from what it actually has. Therefore, an error range with a dimension of 0.164 pixel, on average, is defined.

Even though there is no universally acceptable residual standard deviation threshold, this is a very satisfactory result. Considering that, for the screen used, 1 pixel corresponds to approximately 0.25 millimeters, it is an error practically undetectable to the naked eye, so it is plausible to consider itself irrelevant in the visual deviations presented above.

The residual Standard error refers to an average of the errors so it is important to investigate how many values determined by the regression are expected to have an error in the range of +- 0.082 px.

For exponential regressions it is legitimate to consider that the error terms are Normally distributed (Figure 4.13). For a Normal distribution, 68% of the data is within 1 standard deviation, therefore, 68% of the errors should be within a range of ± 1 residual standard deviation [149].

This implies that it is predicted that 68% the size of the objects of interest in the captured images, relative to the predicted values, will be within ± 0.082 pixel of the real values. This fact will only confirm that the depth estimation method has no impact on the location deviation of virtual content.

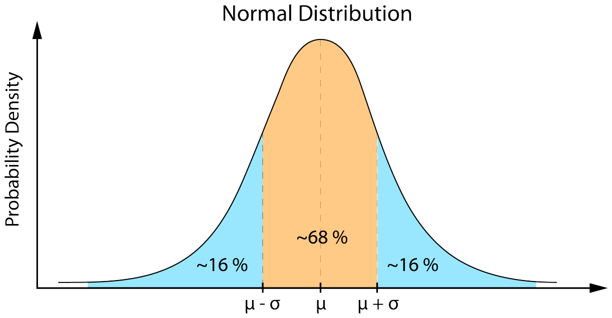


Figure 4.13: Normal Distribution.

As previously mentioned in the second approach, the digital content is projected on the screen for a ratio of linear similarity, disregarding the distorted perception of the screen by the user. Distorted perception results from the user's perspective for positions involving vertical and lateral displacements (Figure 4.14).

A theoretical evaluation of the error between the displayed position of virtual objects on the screen and the real and correct position obtained by the Line of Sight method was carried out. The influence of the user's lateral and vertical displacements in relation to the camera was individually analyzed. Only the area defined by the camera's field of view (79.42° and 50.27 ° for horizontal and vertical view angles, respectively) was considered for the calculation. For reasons of cohesion, the area of analysis was also limited to longitudinal distances belonging to the same interval applied in the elaboration of the exponential regression in Chapter 3, namely distances between 50 and 250 centimeters to the camera. The results are shown in Figure 4.15.

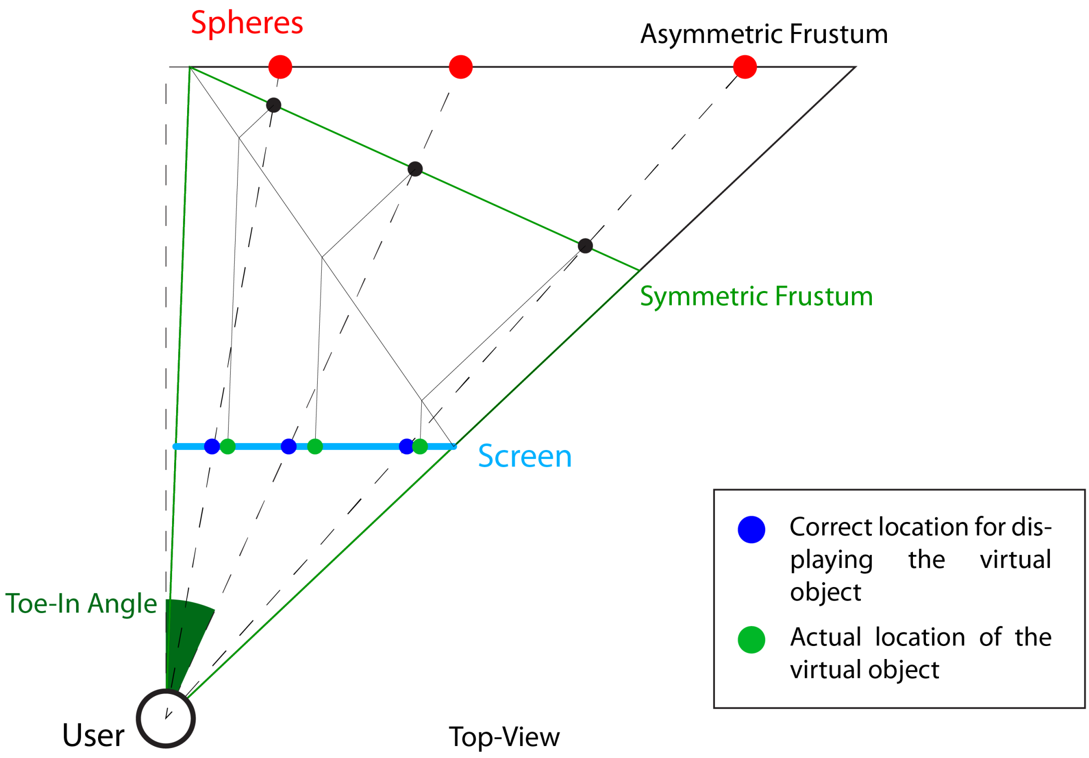

Spheres
Asymmetric Frustum
Symmetric Frustum
Screen
Toe-In Angle
User
Top-View
Correct location for displaying the virtual object
Actual location of the virtual object

Figure 4.14: Comparison between the Line of Sight Intersection Method and the Toe-in Method.

The results that are presented reflect the maximum deviation, in centimeters, present in the image for the respective position of the user in relation to the camera. Note that positive error values represent projected virtual images with a deviation to the right or upwards from the ideally correct position. Negative error values represent projected virtual images with a left or downward shift.

It can be seen from the outset that the results are symmetrical, and, as expected, a lateral shift of the user to the right will imply a shift in the perception of the image to the left. Consequently, a shift to the left will shift the image to the right. It is also concluded from the results that a longitudinal distancing contributes to lower errors.

Quantitatively interpreting the absolute errors, it appears that the maximum error, for the area of analysis, corresponds to just over 2.5 centimeters. Considering that the screen used has a width of 34.4 centimeters, it implies a maximum relative error of 7.27% in relation to the dimensions of the displayed image. This error is obtained for a longitudinal distance of 50.00 centimeters and a horizontal displacement of 41.53 centimeters. For this user position an error of 2.5 centimeters is noticeable and therefore potentially responsible for the failure of the system to identify the respective object.

Since there is no standard threshold value that defines the acceptable error for the deviation of the images presented, a study was carried out to assess the space available for the user to move so that the visual deviation is never greater than 1%, 0.5% or 0.25% of his/her longitudinal distance to the camera. For example, if the user is at a longitudinal distance of 1 meter, regardless of their lateral or vertical displacement, the visual deviation must be less than 10, 5 or 2.5 millimeters, respectively. Considering the value of 2.5 centimeters of deviation, for a longitudinal distance of 50 centimeters, it implies a deviation of 5%. The results obtained are shown in Figure 4.16.

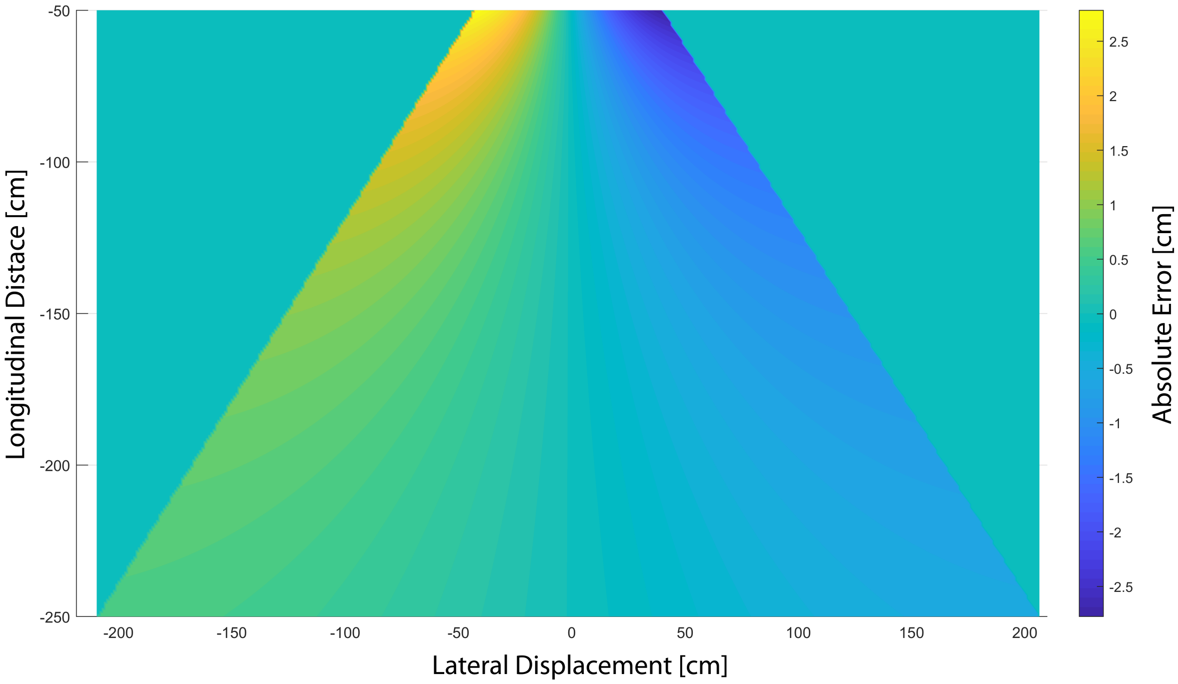


Figure 4.15: Absolute error of the virtual object position as a function of longitudinal distance and lateral displacement.

The results presented so far have been determined by always assuming the same fixed position of a particular object of interest. From this consideration arises the interest in studying the influence of the position of the object's position on the deviation of the image projected on the screen. The absolute error was verified, in the same way for objects that were at the longitudinal distance of the screen of 0.5, 1 and 2.5 meters (Figure 4.17). The results show that only the user's position has an influence on that deviation. This is a predictable conclusion as the shift is caused by the user's visual distortion of the screen due to their perception of perspective, of their current position.

It should be noted that the absolute error is not the same for all pixels in the image. To illustrate this fact, Figure 4.18 represents the deviation in a complete image to the user position defined by the coordinates $(-100, -50, -50)$.

It is possible to understand that in the vertical margins of the image, the absolute horizontal error is null and in the horizontal margins of the image, the absolute vertical error is null. It evolves to the maximum error value as it approaches the geometric center of the image. As mentioned above, the greater the lateral or vertical displacement, the greater the deviation of the image in the respective direction.

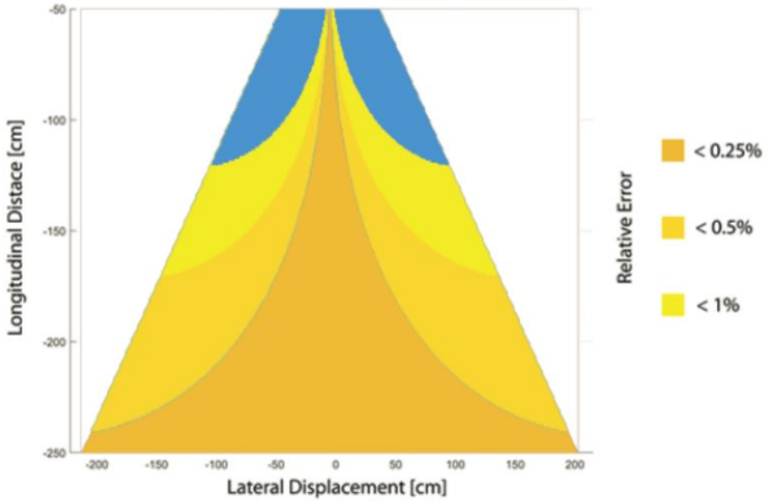
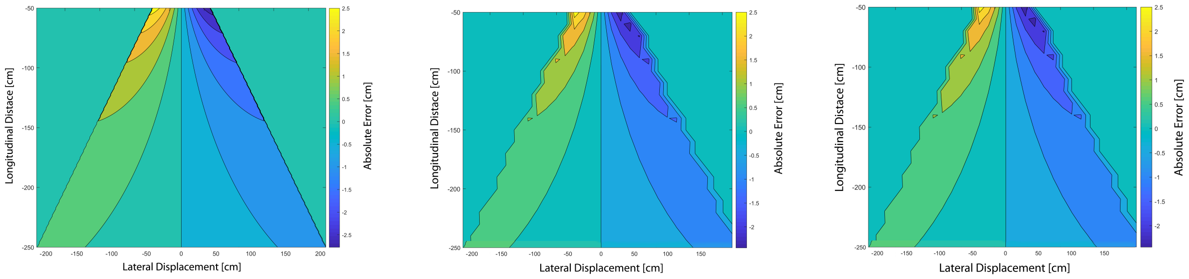


Figure 4.16: Region of space where visual deviation is less than 1%, 0.5% or 0.25%.

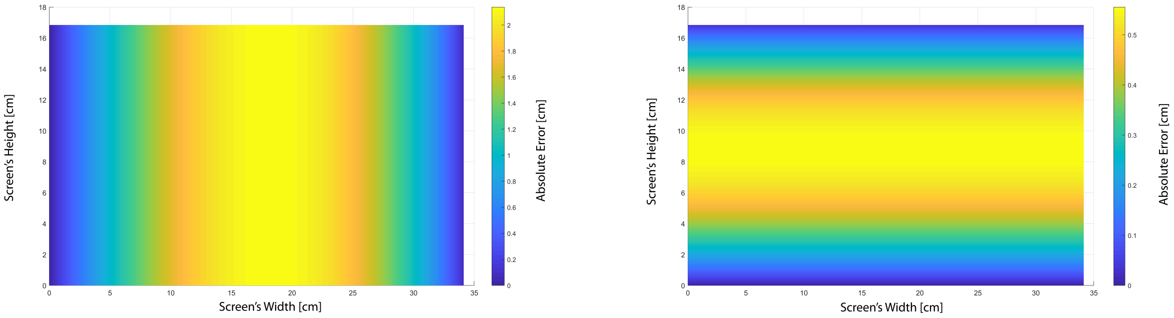


(a) Absolute error at 0.5 m

(b) Absolute error at 1.0 m

(c) Absolute error at 2.5 m

Figure 4.17: Absolute error for objects at a longitudinal distance = 0.5, 1 and 2.5 meters



(a) Absolute error along the horizontal axis

(b) Absolute error along the vertical axis

Figure 4.18: Absolute error of the virtual object position along the screen's horizontal and vertical axis.

In the same way that the presented position of the virtual objects presented deviations in relation to the correct position from the user's perspective, discrepancies in the dimension of the objects were also identified (Figure 4.19). Consequently, it became necessary to assess the influence of the longitudinal distance between the user and the object on the absolute error of the dimensions of virtual objects in relation to real ones. As it is a great challenge to assess the apparent size of objects that are far away, the study was considered using the method developed in the field of Depth Estimation. In other words, the objective is to find out if the apparent dimensions of the virtual objects, on the screen, coincide with the apparent dimensions of the respective real objects, behind the screen. By implementing the Exponential Fit method, it becomes possible to quantify the apparent dimensions of both, regardless of the longitudinal distance that separates the screen from the objects, and the objects from the user. From the results obtained, the absolute error between the real and virtual apparent dimensions is determined.

Illustrating what was explained in this paragraph, considering a sphere, if its real apparent diameter is equal to that of its virtual representation, it implies that the absolute error is null. Negative absolute errors represent an apparent virtual diameter smaller than expected and positive absolute errors represent a virtual diameter larger than expected. For this analysis it will be considered that the lateral and vertical displacements in relation to the geometric center of the screen are null, both for the user and for the object of interest. Thus, the effect of visual distortion shown in previous analyses is negligible.

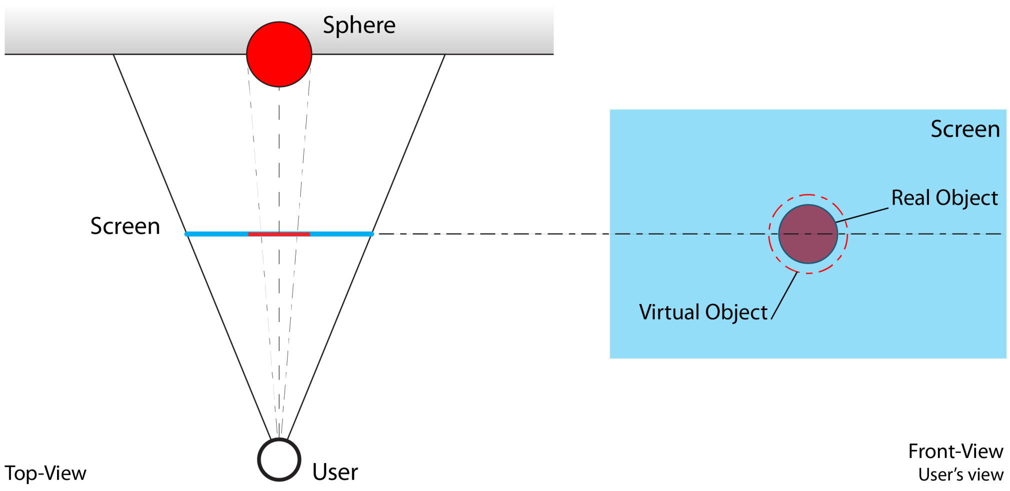


Figure 4.19: Discrepancies between the apparent dimensions of the real objects and their respective virtual representation.

Likewise, the range of longitudinal distances previously presented was considered. The results obtained are shown in Figure 4.20. The horizontal axis represents the longitudinal distance from the user to the screen, in meters, and the vertical axis, the longitudinal distance from the object to the screen, in meters. The study region is not based on an area, as horizontal and vertical displacements are not considered. The study is made regarding user and object movements restricted to a straight line perpendicular to the screen plane. Absolute errors are shown in centimeters.

Figure 4.20 presents theoretical results specific to an object used in the laboratory. Namely, was assumed a sphere with a diameter of 6.14 centimeters.

Counter-intuitively, the results conclude that for the study region, greater distances from the user and object to the camera contribute mainly to an increase in absolute errors. With the exception of the region defined by distance variations between 0.5 and 1.25 meters, approximately, where the referred monotony does not occur.

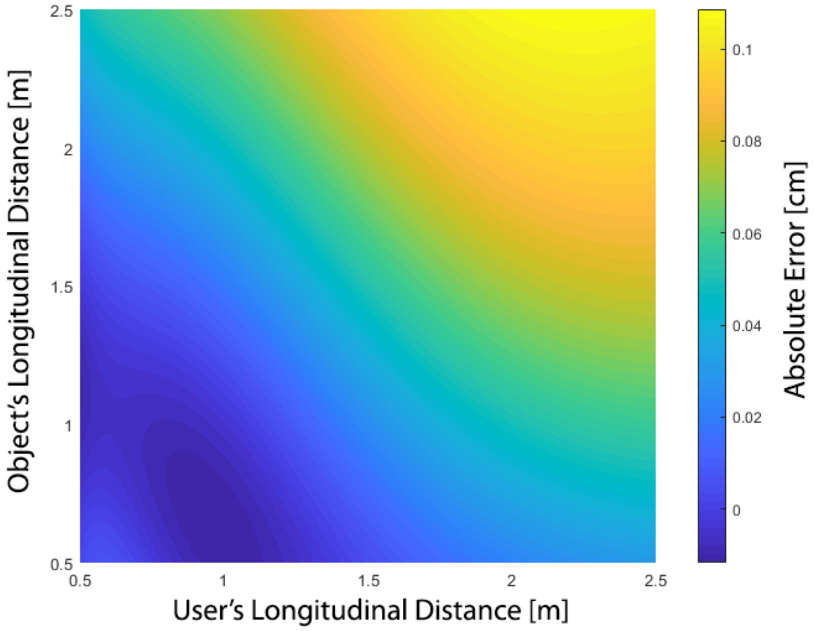
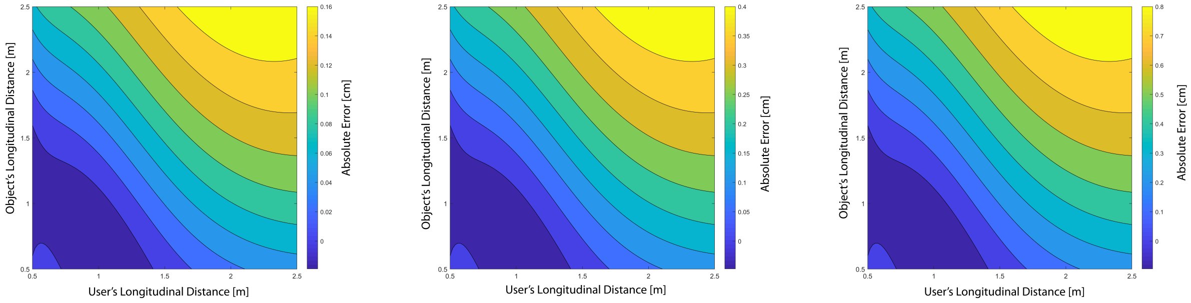


Figure 4.20: Absolute error for the apparent dimensions of the real objects compared their respective virtual representation.

However, the errors that appear are less than 1.3 millimeters, which corresponds to less than 0.05% of the distance from the user and the object to the screen. These results predict confidence in the SST system in the dimensions of virtual objects displayed, specifically for the 6.14 centimeter sphere. Therefore, there is a need to assess the influence of object size on errors. In this way, exactly the same study was carried out for spheres with diameters of 10, 25 and 50 centimeters (Figure 4.21).



(a) Absolute error for 10 cm objects

(b) Absolute error for 25 cm objects

(c) Absolute error for 50 cm objects

Figure 4.21: Absolute error for objects with real dimensions= 10, 25 and 50 cm.

The results establish a pattern between the absolute errors of the apparent dimensions for different diameters of the sphere under study. The error range is the same for any diameter, by a difference of a scale factor. If the maximum error value determined for each of the diameters is considered, it is possible to conclude that the maximum is constant and corresponds to 1.6% of the diameter of the sphere under analysis. Considering that this maximum value is only verified when the user is at a distance of 2.5 meters from the screen, it implies that in the case of the larger diameter sphere, the absolute error corresponds to 0.32% of the user's longitudinal distance.

Following the study carried out for image deviation and since again there is no established threshold value that defines the acceptable error for this context, it will also be opportune to evaluate the diameters of the spheres so that the maximum absolute errors are less than the relative error levels as compared to the user's longitudinal distance.

Therefore, the error in the apparent dimension only exceeds 1%, 0.5% or 0.25% of the longitudinal distance, for spheres with diameters 156,250 cm, 78,125 cm and 39,063 cm, respectively.

The error in the initial sphere was also analyzed for longitudinal distances between 0.5 and 10 meters with the aim of verifying if the absolute error maintained its tendency to increase with increasing distances (Figure 4.22).

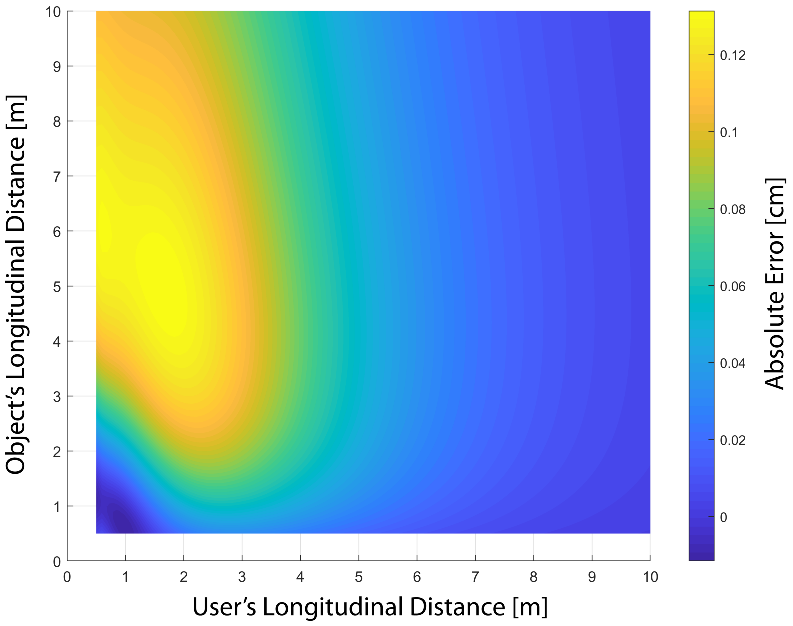


Figure 4.22: Absolute error for the apparent dimensions of the real objects compared their respective virtual representation until 10 meters of longitudinal distance.

It can be seen, therefore, that the error variation reverses the behavior shown above, following a more intuitive evolution, namely, decreasing the error with the increase of longitudinal distances to the screen, similarly to the study on visual deviations.

The apparent perception of objects as a function of their distance from the user is estimated by a non-linear regression, more specifically, by the composition of two exponentials. The absolute error, in turn, is obtained by the difference between two double exponential regressions, which implies the involvement of four distinct exponential functions, which justifies the fact that there is no constant monotony in the results obtained. However, it is possible to state that for the area between 0 and 2 meters of the longitudinal distance of the user and 0 and 7 meters of the longitudinal distance of the object, the error has a mostly positive variation with the increase of either of the two distances. However, outside this region, that is, for values greater than 2 meters for the user and 7 meters for the objects, a tendency is expected to decrease the absolute error in the apparent dimensions.

Chapter 5

Conclusions

For this last chapter the objectives achieved with the development of this project will be presented, in relation to what was initially stipulated to achieve in the first chapter. From the results obtained, there will be room to discuss the areas of application of the developed system, given its advantages and problems that were highlighted through the previous chapter. At the end of this chapter there will be some comments regarding future ideas for work that can be developed within the subject of this project.

The present research work was motivated by the objective of establishing the proof-of-concept of a proposed framework for a Spatial See-Through AR system supported by a transparent screen. The study was governed by several goals initially defined, which aimed to address issues highlighted in other research studies reviewed. The implementation of Computer Vision, Monocular Depth Estimation and 3D Monoscopy methods and algorithms aimed to achieve these goals.

Two specific approaches were presented to achieve the defined objectives through different strategies. The first approach proved to be faithful in presenting a digital marker in the correct position of the transparent screen depending on the user's spatial position. The second approach demonstrated the possibility of representing with great accuracy the apparent dimensions of the modeled objects virtually, relative to real objects. Thus, through separate approaches, it was proved that the reliability of a system that allows interacting with the surrounding environment, assuming the user's field of view. Allowing users to employ their natural spatial perception, a sense of presence in a real world enhanced by virtual content was achieved.

5.1 Achievements

In the area of Depth Estimation, a method specifically adapted to the project context was developed. By the results, the Exponential Fit method proved to be a viable solution with a high level of effectiveness, superior to the revised methods for similar purposes. The implementation of the method together with the application of Computer Vision algorithms contributed to make the user's interaction with the system possible with real-time response.

From the results discussed in Chapter 4, the reliability of the system was shown, but its direct use still has some issues associated with users' comfort and well-being after prolonged use. Symptoms such as headaches, eye fatigue, nausea and even epileptic seizures stand out. Although they have not been experienced, factors were identified that could contribute to their emergence.

This aspect suggests that the proposed framework for an AR system maintains its validity, however it requires the implementation of more refined systems or methods and solutions aimed at correcting and counteracting the factors that contribute to the listed symptoms. Possible examples that could be effective in combating the problems highlighted are based on technology developed by the automotive industry that integrates holographic augmented reality technology on the car's windshield.

Based on the results obtained, it is possible to predict the applicability of this type of technology. The main objective with the implementation of this solution is to allow the user to observe virtual information *in loco*. The vision of Industry 4.0 is coming through over the years so that virtual data is more constant in the industry and AR can enable it to be observed like no other technology. This system is a viable solution to do it with the least impact on the operation and the work environment, allowing it to be possible to produce and visualize a mental model of the user's work panorama, promoting easy data interpretation and decision making capabilities.

Explicitly, for the Manufacturing industry, the following applications are considered that may be viable:

- Monitoring of machines and production lines in real time from a transparent screen through which it is possible to observe the shop-floor. In this way, it is expected to be possible to control the production-rates of the machines and the production times, for example. It will also be a suitable solution to directly identify problems that may arise during production line processes and find out if there is already an operator on site to solve them.

- The previous example is still possible to expand for warehouse management activities where the technology appears to be promising. Namely, for order allocation, inventory management and order picking

- Support in the Machine-Tools area by installing a transparent screen in place of the machine window. By doing so, it may be possible to display real-time information directly superimposed on the operation, such as operating conditions or even virtual representations of the cutting tool movement. This solution may also be useful to guide the operator when the window is obstructed by the lubricant.

- Logistics management is anticipated to be an area of interest for SST system applications as it may allow faster decision-making and reduced error rates by simplifying the execution of natural logistics elements.

Additionally, it is foreseen the applicability of this project in areas that transcend manufacturing. Applications associated with exhibitions and museums, product displays and showcases, activities associated with tourism, culture and entertainment stand out, among which informative and didactic panels are particularly noteworthy. Sport will also be a possible application area, for information purposes in real time, during the course of events, which aim to monitor speeds and distances, among other information, live on the scene. This proposal can be easily extrapolated to any of the areas mentioned above. Applications of the SST system in the area of archeology may also be interesting for the virtual reconstruction of monuments, landscapes or scenery. Through the transparent screen, it is possible to visualize elements from certain eras superimposed on a modern environment, such as the reconstruction of ruins.

5.2 Future Work

For future developments within the scope of this project, the following are highlighted:

- It is foreseen the need to apply or develop methods to counteract or correct the errors highlighted in Chapter 4. In particular, prevent symptoms that jeopardize the user's comfort and reduce the error of visual distortion for approaches identical to the one implemented in 3D Monoscopy, possibly by merging the two developed approaches.

The project focused on evaluating the reliability of the system for the region bounded by the longitudinal distances involved in the Exponential Fit method, more specifically, between 0.5 and 2.5 meters.

- It would be relevant to evaluate in detail the remaining distances, since based on the results and the applicability that is anticipated for the system, the reliability of the SST in the other regions is promising.
- It will be important to carry out tests in real working conditions that aim to assess the acceptance and performance of the technology within the industry and the users themselves.
- The use of RGB-d cameras and transparent OLED screens should make it possible to expand the investigation work by obviating some of the highlighted drawbacks.

5.3 Final Remarks

The research work was carefully developed and structured with the aim of enhancing new projects. The exposed content aims to contextualize and provide the necessary bases for understanding the theme and the areas and scientific methods explored. Thus, it is expected that the project presented will enable the development of many applications in different areas of research.

Bibliography

[1] H. Tamura. Steady steps and giant leap toward practical mixed reality systems and applications. In *Proceedings of the International Status Conference on Virtual and Augmented Reality*, pages 3–12. Citeseer, 2002.

[2] Y. Itoh, T. Langlotz, D. Iwai, K. Kiyokawa, and T. Amano. Light attenuation display: Subtractive see-through near-eye display via spatial color filtering. *IEEE transactions on visualization and computer graphics*, 25(5):1951–1960, 2019.

[3] Azuma, Ronald T. “A Survey of Augmented Reality.” *Presence: Teleoperators and Virtual Environments* 6, 4 (1997):355-385.

[4] Bell, B., Feiner, S. and Hollerer, T., View management for virtual and augmented reality, in *ACM Symposium on User Interface Software and Technology*, CA, USA, 2001, pp. 101–110., .

[5] P. Fite-Georgel. Is there a reality in industrial augmented reality? In *2011 10th ieee international symposium on mixed and augmented reality*, pages 201–210. IEEE, 2011.

[6] Egger, J., Masood, T., Augmented Reality in Support of Intelligent Manufacturing – A Systematic Literature Review, *Computers & Industrial Engineering* (2019), doi: https://doi.org/10.1016/j.cie.2019.106195, .

[7] Seki, Hirosato, Toyokazu Nose, Young Hae Lee et al. ”Special issue on recent advances in Intelligent Manufacturing Systems.” *Computers Industrial Engineering* 65, 1 (2013):1.

[8] Oztemel, Ercan, and Samet Gursev. “Literature review of Industry 4.0 and related technologies.” *Journal of Intelligent Manufacturing* 47, 4 (2018):3.

[9] Pacaux-Lemoine, Marie-Pierre, Damien Trentesaux, Gabriel Zambrano et al. ”Designing intelligent manufacturing systems through Human-Machien Cooperation principles: A human-centred approach.” *Computers Industrial Engineering* 111 (2017):581- 595.

[10] Serrano, Veronica, and Thomas Fischer. “Collaborative innovation in ubiquitous systems.” *Journal of Intelligent Manufacturing* 18, 5 (2007):599–615.

[11] Smith, Shana, Gregory C. Smith, and Roger Jiao et al. “Mass customization in the product life cycle.” *Journal of Intelligent Manufacturing* 24, 5 (2013):877–85.

[12] O. Bimber and R. Raskar. *Spatial Augmented Reality: Merging Real and Virtual Worlds*. A. K. Peters, Wellesley, MA, USA, 2005. ISBN 1-56881-230-2.

[13] I. E. Sutherland, ”Sketchpad-A Man-Machine Graphical Communication System”, *Proceedings of the Spring Joint Computer Conference*, Detroit, Michigan, May 1963 (Washington, D.C.: Spartan, 1964). *Proceedings of IFIP Congress*, pp. 506-508, 1965.

[14] Van Krevelen, D.W.F. and Poelman, R. (2010) A survey of augmented reality technologies, applications and limitations. *The International Journal of Virtual Reality*, 9(2), 1–20.

[15] Eleonora Bottani & Giuseppe Vignali (2019) Augmented reality technology in the manufacturing industry: A review of the last decade, *IISE Transactions*, 51:3, 284-310, DOI: 10.1080/24725854.2018.1493244, .

[16] T. P. Caudell and D. W. Mizell. Augmented reality: An application of heads-up display technology to manual manufacturing processes. In *Proc. Hawaii Int'l Conf. on Systems Sciences*, pp. 659–669, Kauai, HI, USA, 1992. *IEEE CS Press*. ISBN 0-8186-2420-5.

[17] S. Benford, C. Greenhalgh, G. Reynard, C. Brown, and B. Koleva. Understanding and constructing shared spaces with mixed-reality boundaries. *ACM Trans. Computer-Human Interaction*, 5(3):185– 223, Sep. 1998.

[18] Milgram, P. and Kishino, A.F. (1994) Taxonomy of mixed reality visual displays. *IEICE Transactions on Information and Systems*, E77- D(12), 1321–1329, .

[19] Milgram, Paul, Karuo Takemura, and Akira Utsumi et al. “Augmented Reality: A class of displays on the reality virtuality continuum.” *Telemanipulator and Telepresence Technologies*, 2351 (1994):282-2802, .

[20] Santi, G.M.; Ceruti, A.; Liverani, A.; Osti, F. Augmented Reality in Industry 4.0 and Future Innovation Programs. *Technologies* 2021, 9, 33. https://doi.org/ 10.3390/technologies9020033.

[21] S. Feiner, B. MacIntyre, T. Höllerer, and A. Webster. A touring machine: Prototyping 3D mobile augmented reality systems for exploring the urban environment. *Personal and Ubiquitous Computing*, 1(4):74–81, Dec. 1997, .

[22] S. K. Ong , M. L. Yuan A. Y. C. Nee (2008) Augmented reality applications in manufacturing: a survey, *International Journal of Production Research*, 46:10, 2707-2742, DOI: 10.1080/00207540601064773, .

[23] Santi, G.M.; Ceruti, A.; Liverani, A.; Osti, F. Augmented Reality in Industry 4.0 and Future Innovation Programs. *Technologies* 2021, 9, 33. https://doi.org/ 10.3390/technologies9020033.

[24] Reif, R., Gunthner, W.A., Schwerdtfeger, B. and Klinker, G. (2010) Evaluation of an augmented reality supported picking system under practical conditions. *Computer Graphics Forum*, 29(1), 2–12.

[25] Jeon, J., Kim, S. and Lee, S. (2010) Considerations of generic framework for AR on the web, in *Proceedings of the W3C Workshop on Augmented Reality on the Web*. Available online at https://www.w3. org/2010/06/w3car/generic$_{f}ramework.pdf(accessedApril2021).$

[26] Wang, X., Ong, S.K. and Nee, A.Y.C. (2016a) A comprehensive survey of augmented reality assembly research. *Advances in Manufacturing*, 4(1), 1–22., .

[27] Wang, X., Ong, S.K. and Nee, A.Y.C. (2016b) Multi-modal augmentedreality assembly guidance based on bare-hand interface. *Advanced Engineering Informatics*, 30(3), 406–421, .

[28] Wang, X., Ong, S.K. and Nee, A.Y.C. (2016c) Real-virtual components interaction for assembly simulation and planning. *Robotics and Computer-Integrated Manufacturing*, 41, 102–114., .

[29] Craig, A.B. (2013). Understanding augmented reality: Concepts and applications. *Waltham, MA: Elsevier.*

[30] C. E. Hughes, C. B. Stapleton, D. E. Hughes, and E. M. Smith. Mixed reality in education, entertainment, and training. *IEEE Computer Graphics and Applications*, 2005.

[31] Shaaban, O.A., Che Mat, R. and Mahayudin, M.H. (2015) The development of mobile augmented reality for laptop maintenance (MAR4LM). *Jurnal Teknologi*, 77(29), 91–96.

[32] Moser, K., Itoh, Y., Oshima, K., Swan, J.E., Klinker, G. and Sandor, C. (2015) Subjective evaluation of a semi-automatic optical see-through head-mounted display calibration technique. *IEEE Transactions on Visualization and Computer Graphics*, 21(4), 491–500.

[33]]T. Ogi, T. Yamada, K. Yamamoto, and M. Hi-rose. Invisible interface for immersive virtual world. In *IPT'01: Proc. Immersive Projection Technology Workshop*, pp. 237–246, Stuttgart, Germany, 2001.

[34] T. Ohshima, K. Satoh, H. Yamamoto, and H. Tamura. RV-Border Guards: A multi-player mixed reality entertainment. *Trans. Virtual Reality Soc. Japan*, 4(4): 699–705, 1999.

[35] G. Goebbels, K. Troche, M. Braun, A. Ivanovic, A. Grab, K. von Löbtow, H. F. Zeilhofer, R. Sader, F. Thieringer, K. Albrecht, K. Praxmarer, E. Keeve, N. Hanssen, Z. Krol, and F. Hasenbrink. Development of an augmented reality system for intra-operative navigation in maxillo-facial surgery. In *Proc. BMBF Statustagung*, pp. 237–246, Stuttgart, Germany, 2002. I. Gordon and D. G. Lowe. What and where: 3D object recognition with accurate pose.

[36] Doshi, A., Smith, R.T., Thomas, B.H. and Bouras, C. (2017) Use of projector based augmented reality to improve manual spot-welding precision and accuracy for automotive manufacturing. *International Journal of Advanced Manufacturing Technology*, 89(5-8), 1279–1293.

[37] O. Bimber and R. Raskar. *Spatial Augmented Reality: Merging Real and Virtual Worlds*. A. K. Peters, Wellesley, MA, USA, 2005. ISBN 1-56881-230-2.

[38] N. Murauer, N. Pflanz, and C. von Hassel. Comparison of scan-mechanisms in augmented reality-supported order picking processes. In *SmartObjects@ CHI*, pages 69–76, 2018.

[39] F. Saxen, A. Köpsel, S. Adler, R. Mecke, A. Al-Hamadi, J. Tümler, and A. Huckauf. Investigation of an augmented reality-based machine operator assistance-system. In *Companion Technology*, pages 471–483. Springer, 2017.

[40] C. Wu and H. Wang. A multi-modal augmented reality based virtual assembly system. In *Proceedings of the International Conference on Human-centric Computing 2011 and Embedded and Multimedia Computing 2011*, pages 65–72. Springer, 2011.

[41] R. Radkowski and C. Stritzke. Interactive hand gesture-based assembly for augmented reality applications. In *Proceedings of the 2012 International Conference on Advances in Computer-Human Interactions*, pages 303–308. Citeseer, 2012.

[42] J.-M. Lee, K.-H. Lee, D.-S. Kim, and C.-H. Kim. Active insp ection supporting system based on mixed reality after design and manufacture in an offshore structure. *Journal of mechanical science and technology*, 24(1):197–202, 2010.

[43] X. Wang, S. Ong, and A. Y.-C. Nee. Multi-modal augmented-reality assembly guidance based on bare-hand interface. *Advanced Engineering Informatics*, 30(3):406–421, 2016.

[44] S. S. Tegeltija, M. M. Lazarević, S. V. Stankovski, I. P. Ćosić, V. V. Todorović, and G. M. Ostojić. Heating circulation pump disassembly process improved with augmented reality. *Thermal Science*, 20(suppl. 2):611–622, 2016.

[45] F. Ferrise, G. Caruso, and M. Bordegoni. Multimodal training and tele-assistance systems for the maintenance of industrial products: This paper presents a multimodal and remote training system for improvement of maintenance quality in the case study of washing machine. *Virtual and Physical Prototyping*, 8(2):113–126, 2013.

[46] S. Webel, U. Bockholt, T. Engelke, N. Gavish, M. Olbrich, and C. Preusche. An augmented reality training platform for assembly and maintenance skills. *Robotics and autonomous systems*, 61(4): 398–403, 2013.

[47] G. Kurillo, R. Bajcsy, K. Nahrsted, and O. Kreylos. Immersive 3d environment for remote collaboration and training of physical activities. In *2008 IEEE Virtual Reality Conference*, pages 269–270. IEEE, 2008.

[48] G. Caruso, M. Carulli, and M. Bordegoni. Augmented reality system for the visualization and interaction with 3d digital models in a wide environment. *Computer-Aided Design and Applications*, 12(1):86–95, 2015.

[49] M. Merten. Medicine report-awelterte realitat: Fusion of virtual and real expanses. *German ”A rzteblatt*, 104(13):840, 2007.

[50] F. Ahmad and P. Musilek. A keystroke and pointer control input interface for wearable computers. In *Fourth Annual IEEE International Conference on Pervasive Computing and Communications (PERCOM'06)*, pages 10–pp. IEEE, 2006.

[51] M. Kim and J. Y. Lee. Touch and hand gesture-based interactions for directly manipulating 3d virtual objects in mobile augmented reality. *Multimedia Tools and Applications*, 75(23):16529–16550, 2016.

[52] J. Lambrecht, M. Kleinsorge, M. Rosenstrauch, and J. Krüger. Spatial programming for industrial robots through task demonstration. *International Journal of Advanced Robotic Systems*, 10(5): 254, 2013.

[53] F. Ababsa, M. Maidi, J.-Y. Didier, and M. Mallem. Vision-based tracking for mobile augmented reality. In *Multimedia Services in Intelligent Environments*, pages 297–326. Springer, 2008.

[54] F. De Crescenzio, M. Fantini, F. Persiani, L. Di Stefano, P. Azzari, and S. Salti. Augmented reality for aircraft maintenance training and operations support. *IEEE Computer Graphics and Applications*, 31(1):96–101, 2010.

[55] J. Zhang, S.-K. Ong, and A. Y. Nee. Development of an ar system achieving in situ machining simulation on a 3-axis cnc machine. *Computer Animation and Virtual Worlds*, 21(2):103–115, 2010.

[56] C. Koch, M. Neges, M. König, and M. Abramovici. Natural markers for augmented reality-based indoor navigation and facility maintenance. *Automation in Construction*, 48:18–30, 2014.

[57] H. Flatt, N. Koch, C. Röcker, A. Günter, and J. Jasperneite. A context-aware assistance system for maintenance applications in smart factories based on augmented reality and indoor localization. In *2015 IEEE 20th Conference on Emerging Technologies & Factory Automation (ETFA)*, pages 1–4. IEEE, 2015.

[58] Y. Wang, S. Zhang, S. Yang, W. He, and X. Bai. Mechanical assembly assistance using markerless augmented reality system. *Assembly Automation*, 2018.

[59] J. Hu, H. Kim, Q. Cai, C. Peng, M. Chen, J. C. Prieto, A. J. Rosenbaum, J. S. Stringer, and X. Jiang. Fusion of ultrasonic tracking with an inertial measurement unit for high-accuracy 3d space localization. In *2020 IEEE International Ultrasonics Symposium (IUS)*, pages 1–4. IEEE, 2020.

[60] R. Raskar, J. Van Baar, P. Beardsley, T. Willwacher, S. Rao, and C. Forlines. ilamps: geometrically aware and self-configuring projectors. In *ACM SIGGRAPH 2006 Courses*, pages 7–es. 2006.

[61] G. Welch and E. Foxlin. Motion tracking: No silver bullet, but a respectable arsenal. *IEEE Computer graphics and Applications*, 22(6):24–38, 2002.

[62] K. Meyer, H. L. Applewhite, and F. A. Biocca. A survey of position trackers. *Presence: Teleoperators & Virtual Environments*, 1(2):173–200, 1992.

[63] S. You, U. Neumann, and R. Azuma. Hybrid inertial and vision tracking for augmented reality registration. In *Proceedings IEEE Virtual Reality (Cat. No. 99CB36316)*, pages 260–267. IEEE, 1999.

[64] R. Drath and A. Horch. Industrie 4.0: Hit or hype?[industry forum]. *IEEE industrial electronics magazine*, 8(2):56–58, 2014.

[65] A. Yew, S. Ong, and A. Y. Nee. Towards a griddable distributed manufacturing system with augmented reality interfaces. *Robotics and Computer-Integrated Manufacturing*, 39:43–55, 2016.

[66] Y. Liu and Y. Zhang. Controlling 3d weld pool surface by adjusting welding speed. *Welding Journal*, 94(4):125S–134S, 2015.

[67] M. Rohidatun, A. Faieza, M. Rosnah, S. Nor Hayati, and I. Rahinah. Development of virtual reality (vr) system with haptic controller and augmented reality (ar) system to enhance learning and training experience. *International Journal of Applied Engineering Research*, 11(16):8806–8809, 2016.

[68] M. Funk, A. Bächler, L. Bächler, T. Kosch, T. Heidenreich, and A. Schmidt. Working with augmented reality? a long-term analysis of in-situ instructions at the assembly workplace. In *Proceedings of the 10th International Conference on PErvasive Technologies Related to Assistive Environments*, pages 222–229, 2017.

[69] M. Holm, O. Danielsson, A. Syberfeldt, P. Moore, and L. Wang. Adaptive instructions to novice shop-floor operators using augmented reality. *Journal of Industrial and Production Engineering*, 34(5):362–374, 2017.

[70] T. Wójcicki. Supporting the diagnostics and the maintenance of technical devices with augmented reality. *Diagnostyka*, 15(1):43–47, 2014.

[71] L. Hou, X. Wang, L. Bernold, and P. E. Love. Using animated augmented reality to cognitively guide assembly. *Journal of Computing in Civil Engineering*, 27(5):439–451, 2013.

[72] M. Dalle Mura, G. Dini, and F. Failli. An integrated environment based on augmented reality and sensing device for manual assembly workstations. *Procedia Cirp*, 41:340–345, 2016.

[73] U. Jayaram, Y. Kim, S. Jayaram, V. K. Jandhyala, and T. Mitsui. Reorganizing cad assembly models (as-designed) for manufacturing simulations and planning (as-built). *J. Comput. Inf. Sci. Eng.*, 4(2):98–108, 2004.

[74] A. Jönsson, J. Wall, and G. Broman. A virtual machine concept for real-time simulation of machine tool dynamics. *International Journal of Machine Tools and Manufacture*, 45(7-8):795–801, 2005.

[75] H. Kagermann, J. Helbig, A. Hellinger, and W. Wahlster. *Recommendations for implementing the strategic initiative INDUSTRIE 4.0: Securing the future of German manufacturing industry; final report of the Industrie 4.0 Working Group*. Forschungsunion, 2013.

[76] F. Longo, L. Nicoletti, and A. Padovano. Smart operators in industry 4.0: A human-centered approach to enhance operators' capabilities and competencies within the new smart factory context. *Computers & industrial engineering*, 113:144–159, 2017.

[77] M. Peruzzini, F. Grandi, and M. Pellicciari. Exploring the potential of operator 4.0 interface and monitoring. *Computers & Industrial Engineering*, 139:105600, 2020.

[78] S. K. Ong and A. Y. C. Nee. *Virtual and augmented reality applications in manufacturing*. Springer Science & Business Media, 2013.

[79] P. Renner and T. Pfeiffer. Evaluation of attention guiding techniques for augmented reality-based assistance in picking and assembly tasks. In *Proceedings of the 22nd International Conference on Intelligent User Interfaces Companion*, pages 89–92, 2017.

[80] S. Lee and Ö. Akin. Augmented reality-based computational fieldwork support for equipment operations and maintenance. *Automation in Construction*, 20(4):338–352, 2011.

[81] M. Jayaweera, I. Wijesooriya, D. Wijewardana, T. De Silva, and C. Gamage. Enhanced real-time machine inspection with mobile augmented reality for maintenance and repair: Demo abstract. In *Proceedings of the Second International Conference on Internet-of-Things Design and Implementation*, pages 287–288, 2017.

[82] M. Kocisko, M. Teliskova, P. Baron, and J. Zajac. An integrated working environment using advanced augmented reality techniques. In *2017 4th International Conference on Industrial Engineering and Applications (ICIEA)*, pages 279–283. IEEE, 2017.

[83] R. Schlagowski, L. Merkel, and C. Meitinger. Design of an assistant system for industrial maintenance tasks and implementation of a prototype using augmented reality. In *2017 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*, pages 294–298. IEEE, 2017.

[84] J. Wolfartsberger, J. Zenisek, M. Silmbroth, and C. Sievi. Towards an augmented reality and sensor-based assistive system for assembly tasks. In *Proceedings of the 10th International Conference on Pervasive Technologies Related to Assistive Environments*, pages 230–231, 2017.

[85] A. E. Uva, M. Gattullo, V. M. Manghisi, D. Spagnulo, G. L. Cascella, and M. Fiorentino. Evaluating the effectiveness of spatial augmented reality in smart manufacturing: a solution for manual working stations. *The International Journal of Advanced Manufacturing Technology*, 94(1):509–521, 2018.

[86] Tlx @ nasa ames - home. URL `https://humansystems.arc.nasa.gov/groups/tlx/`.

[87] M.-H. Stoltz, V. Giannikas, D. McFarlane, J. Strachan, J. Um, and R. Srinivasan. Augmented reality in warehouse operations: opportunities and barriers. *IFAC-PapersOnLine*, 50(1):12979–12984, 2017.

[88] A. Syberfeldt, M. Holm, O. Danielsson, L. Wang, and R. L. Brewster. Support systems on the industrial shop-floors of the future–operators' perspective on augmented reality. *Procedia Cirp*, 44:108–113, 2016.

[89] L. Hou, X. Wang, and M. Truijens. Using augmented reality to facilitate piping assembly: an experiment-based evaluation. *Journal of Computing in Civil Engineering*, 29(1):05014007, 2015.

[90] J. Blattgerste, B. Strenge, P. Renner, T. Pfeiffer, and K. Essig. Comparing conventional and augmented reality instructions for manual assembly tasks. In *Proceedings of the 10th international conference on pervasive technologies related to assistive environments*, pages 75–82, 2017.

[91] N. Gavish, T. Gutiérrez, S. Webel, J. Rodríguez, M. Peveri, U. Bockholt, and F. Tecchia. Evaluating virtual reality and augmented reality training for industrial maintenance and assembly tasks. *Interactive Learning Environments*, 23(6):778–798, 2015.

[92] S. M. Desa and Q. A. Salih, "Image subtraction for real time moving object extraction," *Proceedings. International conference on computer graphics, imaging and visualization*, 2004. *CGIV 2004*. (pp. 41-45). Penang, Malaysia, Malaysia: IEEE., .

[93] Rosin, Paul L., and Tim J. Ellis. "Image difference threshold strategies and shadow detection." *BMVC*. Vol. 95. 1995., .

[94] Liu, Y., Ai, H., Xu, G. Y. (2001, September). Moving object detection and tracking based on background subtraction. *Proc. SPIE 4554, object detection, classification, and tracking technologies* (Vol. 4554, pp. 62-66). https://doi.org/10.1117/12.441618, .

[95] B.Prabhakar and Damodar V.Kadaba, "Automatic Detection and Matching of Moving Objects", *CRL Technical Journal*, Vo.3 No.3, pp.32-37, Dec 2001., .

[96] S.Y. Koay, A.R. Ramli, Y.P. Lew, V. Prakash and R. Ali, "A Motion Region Estimation Technique for Web Camera Application", *Student Conference on Research and Development Proceedings*, pp. 352-355, Shah Alam Malaysia, 2002., .

[97] C. Bertolini, "Sistema para medição de cores utilizando espectrofotômetro," Universidade Regional de Blumenau, Blumenau, 2010.

[98] Mihai, D. and Străjescu, E. (2007). From wavelength to rgb filter. *U.P.B. Sci.Bull. Series D*, Vol. 69, No. 2, 2007 ISSN1453-2358.

[99] MATLAB. *9.7.0.1190202 (R2019b)*. The MathWorks Inc., Natick, Massachusetts, 2018.

[100] M. K. Dabhi and B. K. Pancholi, "Face Detection System Based on Viola - Jones Algorithm," *International Journal of Science and Research*, vol. 5, no. 4, pp. 2015–2017, 2016., .

[101] R. Boda and M. J. P. Priyadarsini, "Face Detection and Tracking Using KLT and Viola Jones," *ARPN Journal of Engineering and Applied Sciences*, vol. 11, no. 23, pp. 13 472–13 476, 2016, .

[102] Yang, G., Huang, T.S.: Human face detection in complex background. *Pattern Recogn.* 27(1), 53–63 (1994), .

[103] Yow, K.C., Cipolla, R.: Feature-based human face detection. *Image Vis. Comput.* 15(9), 713–735 (1997), .

[104] Gong, S., McKenna, S., Raja, Y.: Modelling facial colour and identity with Gaussian mixtures. *Pattern Recogn.* 31(12), 1883–1892 (1998), .

[105] Craw, I., Tock, D., Bennett, A.: Finding face features. In: *Proceedings of Second European Conference on Computer Vision*, .

[106] Lanitis, A., Taylor, C.J., Cootes, T.F.: An automatic face identification system using flexible appearance models. *Image Vis. Comput.* 13(5), 393–401 (1995), .

[107] Turk, M., Pentland, A.: Eigenfaces for recognition. *J. Cogn. Neurosci.* 3(1), 71–86 (1991), .

[108] Sung, K.-K., Poggio, T.: Example-based learning for view-based human face detection. *IEEE Trans. Pattern Anal. Mach. Intell.* 20(1), 39–51 (1998), .

[109] I. Ishii, H. Ichida, and T. Takaki, “500-fps face tracking system,” *J. Real-Time Image Process.*, vol. 2, pp. 565–568, May 2012., .

[110] Yang, M.H., Kriegman, D.J., Ahuja, N.: Detecting faces in images: a survey. *IEEE Trans. Pattern Anal. Mach. Intell.* 24(1), 34–58 (2002), .

[111] *, Paul A. and Jones, Michael J. ”Rapid Object Detection using a Boosted Cascade of Simple Features”, IEEE CVPR, 2001*.

[112] K. Schwerdt, JL Crowely, Robust face tracking using color, *AFGR00*, 2000., .

[113] D. Meena and R. Sharan, ”An approach to face detection and recognition,” in *Proc. Int. Conf. Recent Adv. Innov. Eng. (ICRAIE)*, Dec. 2016, pp. 1–6., .

[114] Lucas, Bruce D., and Takeo Kanade. 1981. “An Iterative Image Registration Technique with an Application to Stereo Vision.” *Proceedings of the 7th international joint conference on Artificial Intelligence*, Vancouver, BC, Canada, Vol. 2, 674–679. Morgan Kaufmann Publishers.

[115] Tomasi, C. and Kanade, T. 1991. Detection and tracking of point features. *Technical Report CMU-CS-91-132*, Carnegie Mellon University.

[116] Nandita Sethi and Alankrita Aggarwal. 2011. Robust Face Detection and Tracking Using Pyramidal Lucas Kanade Tracker Algorithm. *IJCTA*, vol. 2.

[117] Shi, J., Tomasi, C. . Good features to track. *Technical report, Cornell University*,1993.

[118] Alphonse, P. J. A., and Sriharsha, K. V. Depth perception in single rgb camera system using lens aperture and object size: a geometrical approach for depth estimation. *SN Applied Sciences*, 2021, 3.6: 1-16., .

[119] Seal, Jonathan R.; Bailey, Donald G.; GUPTA, Gourab Sen. Depth perception with a single camera. In: *Proceedings of International Conference on Sensing Technology*. 2005. p. 96-101., .

[120] Ranftl, Rene, et al. Dense monocular depth estimation in complex dynamic scenes. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016. p. 4058-4066., .

[121] A. Joglekar, D. Joshi, R. Khemani, S. Nair, and S. Sahare. Depth estimation using monocular camera. *International journal of computer science and information technologies*, 2(4):1758–1763, 2011.

[122] S.-H. Lai, C.-W. Fu, and S. Chang. A generalized depth estimation algorithm with a single image. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 14(04):405–411, 1992.

[123] A. Rahman, A. Salam, M. Islam, and P. Sarker. An image based approach to compute object distance. *International Journal of Computational Intelligence Systems*, 1(4):304–312, 2008.

[124] A.-N. Spiess and N. Neumeyer. An evaluation of r 2 as an inadequate measure for nonlinear models in pharmacological and biochemical research: a monte carlo approach. *BMC pharmacology*, 10(1):1–11, 2010.

[125] J. Miles. R-squared, adjusted r-squared. *Encyclopedia of statistics in behavioral science*, 2005.

[126] J. Frost. *Introduction to Statistics: An Intuitive Guide for Analyzing Data and*. 2019.

[127] A. G. Barnston. Correspondence among the correlation, rmse, and heidke forecast verification measures; refinement of the heidke score. *Weather and Forecasting*, 7(4):699–709, 1992.

[128] J. F. Kenney. *Mathematics of statistics*. D. Van Nostrand, 1939.

[129] C. Cruz-Neira, D. J. Sandin, T. A. DeFanti, R. V. Kenyon, and J. C. Hart. The cave: audio visual experience automatic virtual environment. *Communications of the ACM*, 35(6):64–73, 1992.

[130] P. Bourke. Calculating stereo pairs. *Repéré à http://paulbourke. net/stereographics/stereorender*, 1999.

[131] L. Harrison, D. McAllister, and M. Dulberg. Stereo computer graphics for virtual reality. *SIGGRAPH'97, Course Notes*, 6, 1997.

[132] L. Meindl. *Omnidirectional stereo rendering of virtual environments*. PhD thesis, Wien, 2015.

[133] L. Lillakas, H. Ono, H. Ujike, and N. Wade. On the definition of motion parallax. *Vision*, 16(2): 83–92, 2004.

[134] D. Mackay. Generating synthetic stereo pairs and a depth map with povray. *Techn. rep., Defence Research and Development Canada, Suffield, CA*, 2006.

[135] R. Kooima. Generalized perspective projection. *J. Sch. Electron. Eng. Comput. Sci*, 6, 2009.

[136] K. Sung, P. Shirley, and S. Baer. *Essentials of interactive computer graphics: concepts and implementation*. CRC Press, 2008.

[137] O. Kreylos. Good stereo vs. bad stereo. URL http://doc-ok.org/?p%3D77.

[138] T. Dobbert. *Matchmoving: the invisible art of camera tracking*. John Wiley & Sons, 2006.

[139] H. Luijten. Basics of color based computer vision implemented in matlab. *TechnischeUniversiteit Eindhoven, Department Mechanical Engineering, Dynamics and Control Technology Group, Eindhoven*, pages 1–24, 2005.

[140] K. Goyal, K. Agarwal, and R. Kumar. Face detection and tracking: Using opencv. In *2017 International conference of Electronics, Communication and Aerospace Technology (ICECA)*, volume 1, pages 474–478. IEEE, 2017.

[141] K. Sharma, V. Gupta, S. Verma, and S. Avikal. Study and implementation of face detection algorithm using matlab. In *2018 International Conference on Recent Innovations in Electrical, Electronics & Communication Engineering (ICRIEECE)*, pages 2745–2748. IEEE, 2018.

[142] Standard body measurements sizing. URL https://www.craftyarncouncil.com/standards/body-sizing.

[143] A. Kompaniets, H. Chemerys, and I. Krasheninnik. Using 3d modelling in design training simulator with augmented reality. 2020.

[144] M. Smith, A. Maiti, A. D. Maxwell, and A. A. Kist. Using unity 3d as the augmented reality framework for remote access laboratories. In *International Conference on Remote Engineering and Virtual Instrumentation*, pages 581–590. Springer, 2018.

[145] X. Liu11, Y.-H. Sohn, and D.-W. Park. Application development with augmented reality technique using unity 3d and vuforia. *International Journal of Applied Engineering Research*, 13(21):15068–15071, 2018.

[146] P. Kumbhar, M. Attaullah, S. Dhere, and S. Hipparagi. Real time face detection and tracking using opencv. *International journal for research in emerging science and technology*, 4(4), 2017.

[147] Deland-Han. Dynamic link library (dll) - windows client. URL https://docs.microsoft.com/en-us/troubleshoot/windows-client/deployment/dynamic-link-library.

[148] U. Technologies. Preparing assets for unity. URL https://docs.unity3d.com/2019.3/Documentation/Manual/BestPracticeMakingBelievableVisuals1.html.

[149] R. W. on The DO Loop. Error distributions and exponential regression models, Sep 2015. URL https://blogs.sas.com/content/iml/2015/09/16/plot-distrib-exp.html.