



Universitat Politècnica de València
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA DEL DISEÑO

von Kármán Institute for Fluid Dynamics
AERONAUTICS AND AEROSPACE DEPARTMENT



Development of a Photogrammetric Measurement Technique for the VKI Plasmatron

Martínez González, Guillermo

Supervisor UPV:

José Luis Lerma García

Israel Quintanilla García

Supervisor VKI:

Olivier Chazot

Advisors VKI:

Bernd Helber

Francisco Torres Herrador

September 18, 2018

To Bernd, Olivier and the rest of the VKI staff, for giving me the great opportunity of
researching with them and be part of the VKI family.

A Francisco, por ser el gran apoyo de este proyecto y enseñarme las cualidades de un
investigador mediante su compromiso.

A Israel y José Luis, por conseguir enseñar fotogrametría a alguien que solo sabe de aviones.
A la universidad y Sergio Hoyas, por guiarme, hacerme crecer y apoyarme tanto durante estos
maravillosos años.

A mis amigos, mi familia y mi pareja, por su apoyo incondicional y su derroche de confianza.
Sois la verdadera recompensa a todo el esfuerzo.

...

A ti, tío Antonio. Siempre estarás con nosotros.

Si he logrado ver más lejos, ha sido porque he subido a hombros de gigantes
Isaac Newton.

Contents

Contents	I
List of Figures	IV
List of Tables	VII
1 Introduction	1
2 Literature review	6
2.1 Hypersonic flow and TPS	6
2.1.1 Characteristics of the hypersonic flow	6
2.1.1.1 Thin shock layers	7
2.1.1.2 Entropy layer	7
2.1.1.3 Viscous interaction	7
2.1.1.4 High-temperature flow	8
2.1.1.5 Low-density flow	9
2.1.2 Ablation study: modeling and experimental research	10
2.2 Digital cameras: history and configuration	11
2.2.1 History and evolution of the cameras	12
2.2.2 Camera parameters and configuration	13
2.2.2.1 Camera exposure	13
2.2.2.2 Digital camera image noise	14
2.2.2.3 Diffraction	15
2.2.2.4 Digital sensor size	16
2.2.2.5 Depth of field	17
2.3 Photogrammetry	19
2.3.1 Camera model	19
2.3.2 Lens distortion	20
2.3.2.1 Radial distortion	20
2.3.2.2 Tangential distortion	21
2.3.3 Projective geometry	21
2.3.4 Camera calibration	22
2.3.4.1 Camera calibration methods	23
2.3.4.2 Calibration procedure	23
2.3.4.3 Calibration tools	24
2.3.4.4 Calibration calculation	24
2.3.5 Stereo camera imaging	25
2.3.5.1 Triangulation	25

2.3.5.2	Rectification	26
2.3.5.3	Epipolar geometry	27
2.3.5.4	The essential and fundamental matrices	29
2.3.5.5	Summary	31
3	Experimental tools	32
3.1	Experimental facilities: VKI Plasmatron	32
3.1.1	Reentry flow simulation	33
3.2	Surface recession experimental investigation	34
3.2.1	Current methodology: High-Speed Camera (HSC) imaging	34
3.2.2	Experimental methodology proposal: Stereophotogrammetry 3D reconstruction	36
4	Software development: <i>Stereo 3D Reconstruction</i>	39
4.1	Software development tools	39
4.1.1	Programming language: Python	39
4.1.2	Stereophotogrammetry libraries: OpenCV	40
4.2	<i>Stereo 3D Reconstruction</i> description	40
4.2.1	Software overview	40
4.2.2	<i>Stereo 3D Reconstruction</i> applications	41
4.2.2.1	Camera calibration	42
4.2.2.2	Stereo camera calibration	49
4.2.2.3	3D reconstruction with calibrated stereo camera	52
4.2.2.4	3D reconstruction with uncalibrated stereo camera	54
5	Experiments and results	57
5.1	Camera Depth of Field calculator	57
5.2	<i>Stereo 3D Reconstruction</i> validation with OpenCV images	60
5.2.1	Individual calibration	60
5.2.1.1	Parallel processing performance boost	61
5.2.1.2	Distortion modeling control tool	63
5.2.1.3	Calibration parameters optimization	65
5.2.2	Stereo camera calibration	68
5.2.3	3D reconstruction	72
5.2.3.1	Disparity map calculation	72
5.2.4	3D point cloud calculation	73
5.3	<i>Stereo 3D Reconstruction</i> experiment with VKI equipment	74
5.3.1	Experimental setup	75
5.3.2	Guide for accurate calibration image capturing	77
5.3.2.1	Camera configuration	77
5.3.2.2	Stereo camera position	77
5.3.2.3	Calibration pattern	78
5.3.2.4	Capturing process	80

5.3.3	Results	81
5.3.3.1	Individual camera calibration	81
5.3.3.2	Stereo camera calibration	87
5.3.3.3	3D reconstruction	89
6	Conclusions	94
6.1	Conclusions and future work on <i>Stereo 3D Reconstruction</i>	94
6.2	Conclusions and future work on experiments	95
6.3	UPV Photogrammetry collaboration	95
7	Budget	96
	References	98

List of Figures

1.1	Vostok I, in which Russian Major Yuri Gagarin became the first human to fly at hypersonic speed, during the world’s first manned, orbital flight. 12 April 1961 (National Air and Space Museum).	2
1.2	Ablative TPS reentry environments of previous missions illustrate the wide range of applications; values in parentheses are the fraction of the entry probe mass devoted to TPS [2].	3
1.3	Galileo Probe Heat shield [2]. TPM distribution before and after reentry.	4
2.1	Physical effects characteristic of hypersonic flow [7].	6
2.2	High-temperature shock layer [7].	9
2.3	Camera obscura working principle[R2].	12
2.4	The exposure triangle [18].	13
2.5	Diffraction effect depending on the aperture [18].	15
2.6	Definition of airy disk and example of image resolution [18].	16
2.7	DSLR camera sensor size [18].	16
2.8	Diagram depicting depth of focus versus camera aperture [18].	18
2.9	Pinhole camera model [20].	19
2.10	Radial distortion effect [21].	21
2.11	Tangential distortion effect [21].	21
2.12	Change from world to pixel coordinates process [21].	22
2.13	With a perfectly undistorted, aligned stereo rig and known correspondence, the depth Z can be found by similar triangles [20].	26
2.14	Depth and disparity are inversely related, so fine-grain depth measurement is restricted to nearby objects [20].	27
2.15	A mathematical alignment of the two cameras into one viewing plane so that pixel rows between the cameras are exactly aligned is desired [20].	27
2.16	Stereo coordinate system used for undistorted rectified cameras: the pixel coordinates are relative to the upper left corner of the image, and the two planes are row-aligned [20].	28
2.17	The epipolar plane is defined by the observed point P and the two centers of projection, O_l and O_r ; the epipoles are located at the point of intersection of the line joining the centers of projection and the two projective planes [20].	28
2.18	The essential geometry of stereo imaging is captured by the essential matrix E, which contains all of the information about the translation T and the rotation R [20].	29
3.1	Schematic of the VKI Plasmatron system [13].	32

3.2	Use of ICP jet for TPM testing: Simplified hypersonic flow in real flight condition and subsonic plasma jet following the LHTS methodology, reproducing the stagnation point region [13].	33
3.3	Surface recession optical measurement limitation.	35
3.4	Surface recession optical measurement limitation.	35
3.5	Surface recession optical measurement limitation.	36
3.6	Surface recession optical measurement limitation.	36
3.7	Optical accesses in VKI Plasmatron (side view).	37
3.8	Optical accesses in VKI Plasmatron (back view).	38
3.9	Schematic of experimental setup in VKI Plasmatron [13].	38
4.1	Schematic of <i>Stereo 3D Reconstruction</i> software: characteristics, initialization and results.	41
4.2	Example of boxplot and probability density function of a normal set. Courtesy of Chen-Pan Liao.	48
4.3	Definition of maximum allowable error α in a boxplot.	48
4.4	Schematic of the optimization algorithm implemented in <i>Stereo 3D Reconstruction</i>	48
4.5	Schematic of <i>Stereo 3D Reconstruction</i> software application for calibrated stereo camera.	52
4.6	Comparison of images with a low and large number of potential features to be detected.	53
4.7	Schematic of <i>Stereo 3D Reconstruction</i> software application for uncalibrated stereo camera.	54
5.1	Total DoF in meters with respect to f in mm.	59
5.2	Total DoF in meters with respect to aperture number N	59
5.3	Total DoF in meters with respect to aperture number N and focal length in mm. The red cross of the diagram represents a possible configuration in VKI Plasmatron.	60
5.4	Example of three stereo image pairs provided by OpenCV [tutorials].	61
5.5	Example of initialization file for individual calibration.	62
5.6	Example of initialization file for individual calibration.	63
5.7	Distribution of areas in the image for the distortion modeling control tool.	64
5.8	Undistortion effect in an image. Courtesy of OpenCV.	65
5.9	Optimization of f_x showed in Results folder: histogram, pdf and boxplot of the initial and optimum subsets.	66
5.10	Optimization of the focal length parameters.	67
5.11	Optimization of the principal point coordinates.	67
5.12	Optimization of the radial distortion coefficients.	68
5.13	Optimization of the tangential distortion coefficients.	68
5.14	Example of initialization file for stereo camera calibration.	69
5.15	Optimization of the translation vector norm and its components showed in Results folder.	71
5.16	Optimization of the translation rotation vector norm and its components showed in Results folder.	71

5.17 Stereo pair of images of an aloe vera plant used in the 3D reconstruction. Courtesy of OpenCV.	72
5.18 Stereo pair of images of an aloe vera plant used in the 3D reconstruction. Courtesy of OpenCV.	73
5.19 3D Reconstruction of the aloe vera plant visualized in MeshLab.	74
5.20 3D Reconstruction of the aloe vera plant visualized in MeshLab.	74
5.21 SP-12000-CX4 camera model by Stemmer Imaging.	75
5.22 Picture of the calibration pattern.	76
5.23 Picture of a VKI cork sample.	76
5.24 Example of one of the calibration patterns used along the experiments of the project.	79
5.25 Example of corner density for a 6 x 4 grid calibration pattern.	79
5.26 Example of corner density for a 24 x 16 grid calibration pattern.	79
5.27 Example of one of the calibration patterns used along the experiments of the project.	80
5.28 Maximum recommended angle of the calibration pattern during image acquisition. Courtesy of MathWorks.	81
5.29 Example of images used for left camera calibration.	81
5.30 Example of initialization file for individual calibration.	82
5.31 Optimization of the camera matrix calibration parameters for the Left camera. . . .	84
5.32 Optimization of the camera matrix calibration parameters for the Right camera. . .	85
5.33 Optimization of the radial distortion coefficients for the Left camera.	85
5.34 Optimization of the radial distortion coefficients for the Right camera.	86
5.35 Optimization of the tangential distortion coefficients for the Left camera.	86
5.36 Optimization of the tangential distortion coefficients for the Right camera.	86
5.37 Example of initialization file for istereo camera calibration.	88
5.38 Optimization of the translation vector norm and components for the stereo camera.	89
5.39 Optimization of the rotation vector norm and components for the stereo camera. . .	90
5.40 Images used for the first reconstruction of the VKI Plasmatron sample.	90
5.41 Images used for the second reconstruction of the VKI Plasmatron sample.	91
5.42 Normalized disparity map of the VKI Plasmatron sample.	91
5.43 3D point cloud result for the first reconstruction of the VKI Plasmatron sample. . .	92
5.44 3D point cloud result for the second reconstruction of the VKI Plasmatron sample. .	92
5.45 3D mesh result for the first reconstruction of the VKI Plasmatron sample.	93

List of Tables

4.1	Example of number of subsets C obtained with 20 images N and certain combinations of images per set m	46
5.1	Example with the diffraction calculator.	58
5.2	Inputs of the DoF calculator.	58
5.3	Results of the DoF calculator.	58
5.4	Image points distribution along the image area.	64
5.5	Focal length parameters before and after optimization.	67
5.6	Central point coordinates before and after optimization.	67
5.7	Translation vector norm and components before and after calibration.	70
5.8	Rotation vector norm and components before and after calibration.	70
5.9	OpenCV StereoSGBMcreate inputs for disparity calculation.	73
5.10	SP-12000-CX4 remarkable specifications.	75
5.11	Initialization file for the calibration optimization.	83
5.12	Camera matrix parameters optimization.	84
5.13	Translation vector norm and components optimization.	87
5.14	Rotation vector norm and components optimization.	89
5.15	OpenCV StereoSGBMcreate inputs for disparity calculation.	91
7.1	Human and material resources.	96
7.2	Partial and total cost of the human resources of the project.	96
7.3	Breakdown of the personal cost of the project.	97
7.4	Breakdown of the material resources of the project.	97

1 Introduction

The aim of this chapter is to provide the reader with general knowledge about hypersonic flow and thermal protection systems applied in space exploration and atmospheric reentry. The motivation of this project is to develop a photogrammetric measurement technique and to apply it to the experimental research developed in a field where it has been rarely used, the aerospace sector.

Almost everyone has their own definition of the term hypersonic. If we were to conduct something like a public opinion poll [...] to name a Mach number above which the flow of a gas should properly be described as hypersonic there would be a majority of answers round about five or six, but it would be quite possible for someone to advocate, and defend, numbers as small as three, or as high as 12.

*P. L. Roe, comment made in a lecture at the Von Kármán Institute.
Belgium, January 1970.*

V-2 rocket is one of the models taken by the USA after the World War II from Germany. A program called “Bumper” by the U.S Army consisted on mounting a slenderer rocket called WAC Corporal on top of the V-2 in order to demonstrate the use of multi-stage rockets in order to achieve high speeds and altitudes.

The 24th of February 1949, after the ignition of the V-2, the WAC Corporal is fired and reaches a maximum speed of 8288 km/h and an altitude of 393 km. This multi-stage rocket combination was the first object of human origin to achieve hypersonic flight (the first time that any vehicle has flown faster than five times the speed of sound).

Some years later, on 12th April 1961, the scene changes to Russia. This day the Major Yuri Gagarin became the first human in history to fly in space, to orbit the Earth and safely return. Major’s Gagarin orbital craft, Vostok I, achieved speeds exceeding 25 times the speed of sound. Then, this was the first time that a human experienced hypersonic flight.

A few months later, on 23rd June 1961, the pilot Major Robert White from the US Air force flies the X-15 airplane at Mach 5.3. The same year White extended his own record to Mach 6 in the same airplane. This was the first hypersonic flight inside of an airplane.

The last milestone exposed in this section will be the historic Apollo spacecraft, which returned the first and only men reaching the Moon on 1969. The Apollo command module’s reentry was performed at a maximum velocity of Mach 36.

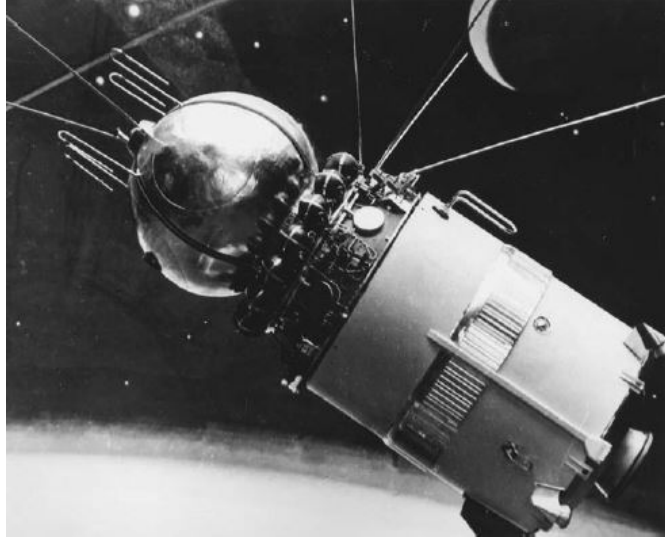


Figure 1.1: Vostok I, in which Russian Major Yuri Gagarin became the first human to fly at hypersonic speed, during the world's first manned, orbital flight. 12 April 1961 (National Air and Space Museum).

As it has been commented by Prof. Chazot at VKI lectures: *hypersonics is when you start focusing on heat transfer rather than aerodynamics*. The huge amount of energy produced under these conditions made necessary to research new ways to maintain the integrity of the spacecraft while performing critical tasks like orbiting around Earth and reentering the atmosphere. The first progresses were made in the design, as hypersonic vehicles tend to follow the blunt body theory and acquire a blunt shape so that a strong shock wave is produced in front of the vehicle. In this case, the hot gases created in the bow shock are deflected away from the surface [1]. This development meant a revolution in the design concepts regarding high velocity vehicles, as supersonic ones had slender bodies. This slender shape cannot be applied in atmospheric entry as the oblique shock waves result in extreme heat loads in the spacecraft surface.

However, even using a blunt shape is not enough if the reentry speed is high enough. For example, a speed of 11 km/s (Apollo 10 reentry) can lead the flow to reach several thousand Kelvin. This temperature is prohibitive for any known material. Therefore it is necessary to dissipate the thermal energy reaching the spacecraft by means of **Thermal Protection Systems (TPS)**.

Ablative heat shields were the first option developed in order to handle the high thermal energy. This thermal protection materials are transformed and destroyed to dissipate heat. On the early starts of the space race between the Soviet Union and the United States this ablators were used as heat shields in many famous spacecraft as Vostok I and Apollo's command module.

When Apollo program was completed the study of the heat shields shifted from ablative TPS to reusable ones. In the late 70s the Space Shuttle orbiter had to face lower heat fluxes during reentry as its main missions were delivering payload and crew to the ISS, put satellites in orbit and maintain the Hubble Space Telescope and the ISS. These reusable materials are able to

resist high temperature, have a low heat conductivity and a high heat capacity. Therefore in the reentry there is an equilibrium where the incoming heat flux is balanced with the re-radiated energy.

Nowadays, the research is shifting again towards the ablative heat shields. Focusing mainly in obtaining lightweight ablators. This materials can withstand higher temperatures than the reusable ones and are the only solution to perform high speed reentries (more than 10 km/s). This is something mandatory if it is intended to carry astronauts beyond LEO (round trip to Mars, asteroids, etc.). In Figure 1.2 it can be seen how the TPS mass increases with the heat flux received by the spacecraft.

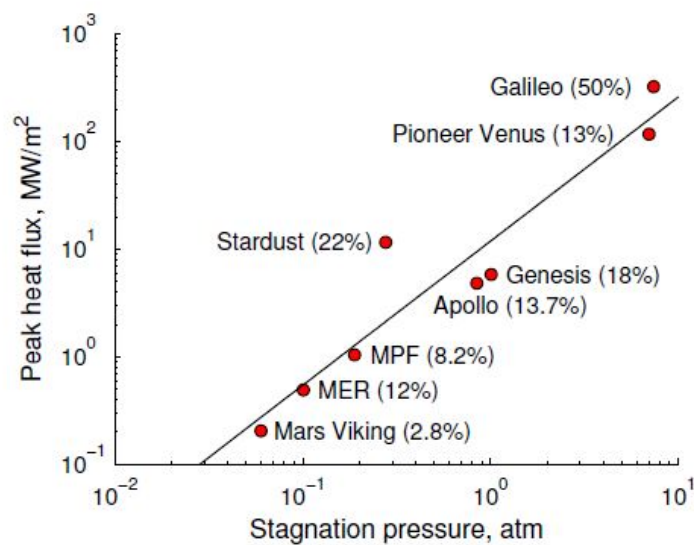


Figure 1.2: Ablative TPS reentry environments of previous missions illustrate the wide range of applications; values in parentheses are the fraction of the entry probe mass devoted to TPS [2].

As it is showed in Figure 1.2, Galileo probe had a heat shield corresponding to half the total mass of the probe. This mission has been the most challenging atmospheric entry ever performed. The probe entered Jupiter’s atmosphere without previous brake and at speed of >47 km/s, decelerating at 228 g from Mach 50 to subsonic speeds in less than two minutes and loosing 80 kg of TPM during the process [3],[4]. However, it is important to remark that the estimation of the TPM distribution was not accurate. As it can be observed in Figure 1.3, the ablation of the TPM in the stagnation point was overestimated, increasing the mass of the probe and adding extra complexity to the mission. Therefore, it is important to model and study the behavior of the ablators as well as be able to control the surface recession rate. Here is where this project tries to contribute by means of a highly accurate measurement technique for the surface recession in three dimensions.

It is also important to remark that the Thermal Protection Materials (TPM) can be classified following different criteria, but the degradation mechanism will be the one used in this introduction:

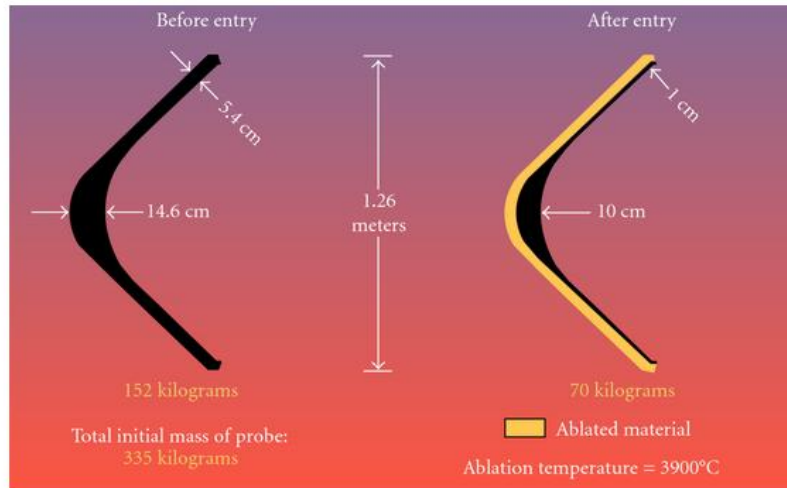


Figure 1.3: Galileo Probe Heat shield [2]. TPM distribution before and after reentry.

- **Reusable heat shields:** These materials dissipate heat by re-radiating it to the atmosphere. They need to resist high temperatures and have a high emissivity at these conditions. They are used for mild Earth reentries and can also be used for orbital reentries at moderate heat fluxes. They are made of dense composites (carbon/carbon, carbon/silicon carbide, etc.) and their maintenance is quite expensive.
- **Ablative heat shields:** Ablators can also be classified by their pyrolyzing/charring properties. Moreover, pyrolyzing ablaters are generally categorized by their density, ranging from low density ($<350\text{kg/m}^3$) to high density ($>1000\text{kg/m}^3$) ablaters.

Therefore, depending on the mission and the atmospheric environment during the reentry, there will be certain conditions that will determine the material used (Peak heat flux, total heat load, peak pressure, and gas composition). These conditions will influence the trade-off that has to be made between material strength and thermal conductivity, both increasing with density.

Most of charring, lightweight ablative materials are composed of a rigid precursor (i.e. carbon) and a filling matrix (i.e. silicon), to serve as a pyrolyzing, ablating, and insulating material at low weight. In this way, they are able to dissipate high heat fluxes through chemical and physical decomposition, transforming the thermal energy into mass loss and surface recession, while the remaining solid material insulates the vehicle substructure.

At this point, the physical nature of the conditions faced by a spacecraft at hypersonic velocities has been presented, as well as it has been comprehended that the use of TPS is mandatory in the most challenging missions. However, to use these TPSs it is necessary to study

their materials in detail and simulate such conditions in ground-test facilities. The objective of this project is the development of a photogrammetric measurement technique that allows the study of the surface recession rate of the TPMs in these facilities. However, now is when a question may arise: What is photogrammetry?

Photogrammetry is the science of making measurements from photographs [5]

- '*PHOTO*': light
- '*GRAM*': drawing
- '*METRY*': measurement

Photogrammetry is the technique that defines precisely the shape, dimensions and position of an object in the space, using basically measurements made in one or more pictures from this object. If one picture is used, it is possible to find information about the two-dimensional geometry of the object. If two images are taken, it is possible to obtain three-dimensional information from the overlapping area of the pictures (stereoscopic vision). By taking more images it will be feasible to reconstruct the whole geometry of a real world object.

Along this project the main technique employed will be stereophotogrammetry, which employs two or more pictures taken from different positions to estimate the three-dimensional coordinates of the object points. There are many methodologies that can be employed to reconstruct objects in 3-D using this technique depending on the characteristics of the experiment. For example, taking many pictures of a fixed object while moving the camera from one position to another and then reconstructing it (this is widely extended in aerial photogrammetry and in cultural heritage reconstruction). However, as the target object of this project changes with time, another methodology should be used.

The main constrains in this project is given by the geometry of plasma tunnels; the optical access are few due to the extreme inner conditions, and by the fact that the probe shape changes with time. Therefore, it is necessary to have several cameras working at the same time in some defined positions. In photogrammetry it is possible to use as many cameras as you want, but in order to make a flexible technique adaptable to any ground-test facility and to reduce the complexity of the triangulation, a set-up formed by two equal cameras is proposed.

Then, the aim is to reproduce binocular vision, which is the one used by humans and many more animals. As human eyes are located in different positions of the head, binocular vision provides two different images. Therefore, the position of a certain real world point will have a different position in the images projected (there will be a mainly horizontal displacement in human case). This positional difference is called disparity, and is what creates the perception of depth for the real world object, commonly referred as "stereoscopic depth" [6].

2 Literature review

2.1 Hypersonic flow and TPS

There is a rule of thumb that defines hypersonic aerodynamics as a flow with a Mach number exceeding 5. However, the flow does not change dramatically its conditions at this Mach number. Therefore, hypersonic flow is best defined as that regime where certain physical flow phenomena become progressively more important as the Mach number is increased to higher values. In addition, it is known that under this conditions the use of Thermal Protection Systems is mandatory to maintain the safety of the spacecraft systems.

2.1.1 Characteristics of the hypersonic flow

The physical phenomena changing in the hypersonic flow can be seen in Figure 2.1. A detailed description of the most important characteristics of this flow will be made below following the explanations of the widely extended book by John David Anderson: [7].

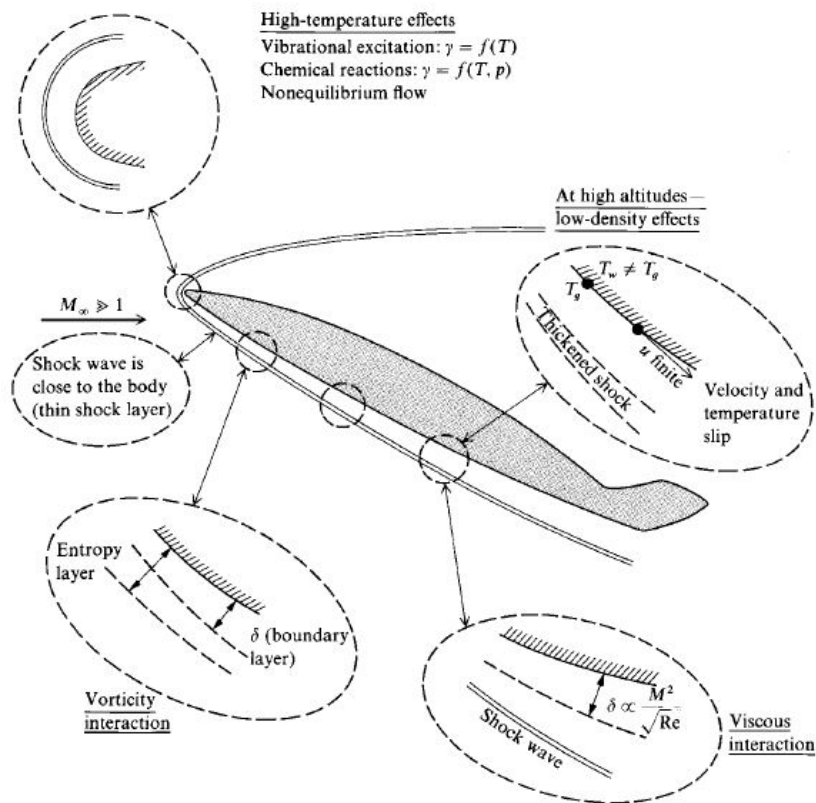


Figure 2.1: Physical effects characteristic of hypersonic flow [7].

2.1.1.1 Thin shock layers

It is known from oblique shock theory that the density increases across the shock wave for a given deflection angle. This increment becomes progressively higher when Mach number grows. Moreover, at higher densities it is easier for the flow to pass through smaller area. Then, in the case of hypersonic flow the distance between the shock wave and the body tends to be small. This distance defines the shock layer, a flowfield that can be quite thin for hypersonic speeds. In addition, high-temperatures chemical reacting effects made the shock-wave angle even smaller.

The fact that shock waves lie close to the body and that the shock layer is thin for hypersonic flow can induce some physical changes. At low Reynolds numbers the merging of the shock wave with a thick, viscous boundary layer can create physical complications. On the other hand, at high Reynolds numbers the shock layer can be considered inviscid and its thinness can be used as an advantage using a general analytical approach called thin shock-layer theory.

2.1.1.2 Entropy layer

The entropy of the flow increases across a shock wave, and the stronger the shock wave, the larger the entropy increase. Moreover, in the nose region of the spacecraft the shock wave is highly curved. When the flow passes through the strong, normal portion of the shock near the stagnation point, it will suffer a larger entropy increase than a flow passing through a weaker portion of the shock. Then, there is a high entropy gradient generated in the nose region; this entropy layer flows downstream and wets the body for large distances from the nose.

The boundary layer grows along the surface inside of this entropy layer and is affected by it. The entropy layer is also a region of strong vorticity as related through Crocco's theorem from classical compressible flow, this interaction is sometimes called a vorticity interaction. The entropy layer causes analytical problems when we wish to perform a standard boundary-layer calculation.

2.1.1.3 Viscous interaction

Hypersonic flow contains a huge amount of kinetic energy due to the high-velocities. Hence, placing a flat plate in a hypersonic flow produces its deceleration due to the viscous effects within the boundary layer. In this case, a considerable amount of the kinetic energy is transformed into internal energy due to the viscous dissipation and temperature increases within the boundary layer.

The characteristics of the hypersonic boundary layers are dominated by this temperature increases. For example, viscosity coefficients increases with temperature, what will produce a thicker boundary layer. Moreover, as pressure p is constant in the normal direction, the increase of temperature T results in a decrease of density ρ through the equation of state $\rho = p/RT$. In

this case, the boundary layer becomes thicker as more area is needed by a lower density flow to pass through.

The thick boundary layer in hypersonic flow can exert a major displacement effect on the inviscid flow outside the boundary layer, causing a given body shape to appear much thicker than it really is. The outer inviscid flow is greatly changed; the changes in the inviscid flow in turn feed back to affect the growth of the boundary layer. This major interaction between the boundary layer and the outer inviscid flow is called viscous interaction. This interaction can have considerable effects on the surface-pressure distribution. Moreover, skin friction and heat transfer are increased by viscous interaction too.

The boundary layer on a hypersonic vehicle can become so thick that it essentially merges with the shock wave (merged shock layer). When this happens, the shock layer must be treated as fully viscous, and the conventional boundary layer analysis must be completely abandoned.

2.1.1.4 High-temperature flow

As discussed previously, the strong viscous dissipation taking place in the hypersonic boundary layer can create extremely high temperatures. This energy excites the vibrational energy internally within the molecules and can lead to dissociation and ionization in the gas. Moreover, in the case of the ablative thermal protection systems, the chemical products derived from the ablation and pyrolysis are also present in the boundary layer. This produces complex hydrocarbon chemical reactions. In consequence, a hypersonic spacecraft can be wetted by a chemically reacting high-energy boundary layer.

The temperature can also be extremely high behind the bow shock in the nose region, where it is nearly normal to the flow. For example, during the Apollo reentry a temperature of 11000 K was reached on the nose region at a Mach number of 36.

The inclusion of the chemically reacting effects is very important to calculate the shock-layer temperature. In addition, in this flow conditions, not only can the boundary layer be chemically reacting, but the entire shock layer can be dominated by chemically reacting flow.

Regarding the chemical nature of this process, if the vibrational excitation and chemical reactions take place very rapidly in comparison to the time it takes for a fluid element to move through the flowfield, we have vibrational and chemical equilibrium flow. If the opposite is true, we have nonequilibrium flow, which is considerably more difficult to analyze.

The most dominant aspect of high temperatures in hypersonics is the resultant high heat-transfer rates to the surface. Aerodynamic heating dominates the design of all hypersonic vehicles. This aerodynamic heating takes the form of heat transfer from the hot boundary layer to the cooler surface (convective heating). Moreover, if the shock-layer temperature is high

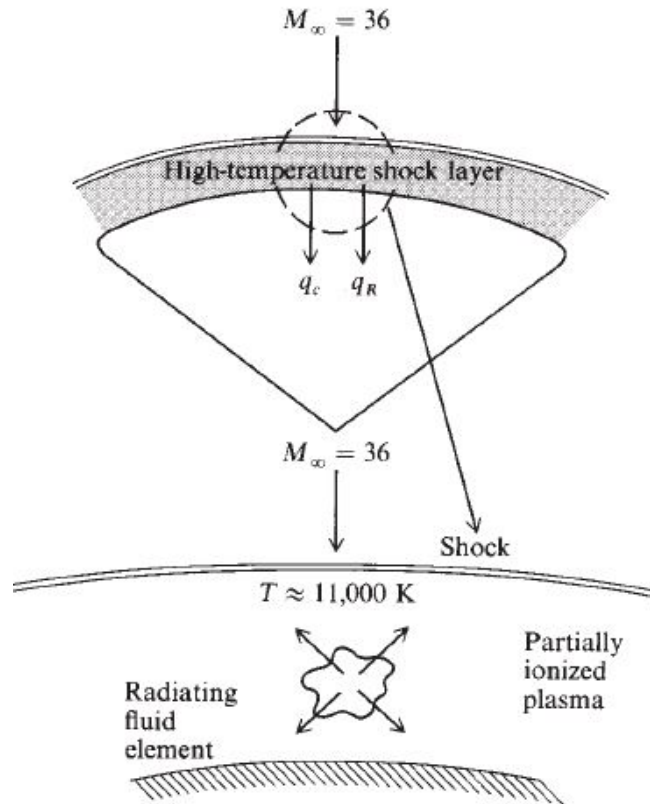


Figure 2.2: High-temperature shock layer [7].

enough the thermal radiation emitted by the gas itself can become important, giving rise to a radiative flux to the surface (radiative heating).

Another consequence of high-temperature flow over hypersonic vehicles is the "communications blackout" experienced at certain altitudes and velocities during atmospheric entry, where it is impossible to transmit radio waves either to or from the vehicle. This is caused by ionization in the chemically reacting flow, producing free electrons that absorb radio-frequency radiation.

2.1.1.5 Low-density flow

Most aerodynamic problems are calculated assuming the flow as a continuum medium. However, as the altitude increase, the density of the air becomes lower. At sea-level conditions the mean free path (average distance between successive molecular collisions of the medium) for air is $\lambda = 6.63 \cdot 10^{-8}m$, while at an altitude of 104.24 km $\lambda = 0.3m$. In this last case the air does not feel as a continuous substance and aerodynamics have to be calculated using concepts from kinetic theory. This is a regime called low-density flow.

There are hypersonic applications at high-altitudes that involve low-density flow. At a certain altitude viscous flow no-slip conditions cannot be assumed any more. Specifically, at low

densities the flow velocity at the surface, which is normally assumed to be zero because of friction, takes on a finite value. This is called the velocity-slip condition. In the same way, the flow temperature at the surface, which is normally taken as equal to the surface temperature of the material, now acquires a different value. This is called the temperature slip condition. In this case the equations remain as the continuum-flow ones but using the velocity and temperature-slip conditions.

If the altitude continues to increase, continuum-flow equations become no longer valid. In this case, the kinetic theory must be applied in order to predict the aerodynamic behavior. Finally, the air density can become low enough that only a few molecules impact the surface per unit time, and after these molecules reflect from the surface they do not interact with the incoming molecules. This is the regime of free molecule flow. For the space shuttle, the free molecular regime begins about 150 km. Therefore, in a simplified sense, we visualize that a hypersonic vehicle moving from a very rarified atmosphere to a denser atmosphere will shift from the free molecular regime, where individual molecular impacts on the surface are important, to the transition regime, where slip effects are important, and then to the continuum regime.

2.1.2 Ablation study: modeling and experimental research

The previously explained conditions taking place in the hypersonic flow and the fact that space missions are more challenging each day has rapidly increased the demand on the new lightweight ablative heat shields. Numerical simulation, ground-testing and material qualification is necessary to determine if a new ablator is valid for a certain mission.

There are many investigations currently studying the behavior of ablative thermal shields. Among them, it is important to highlight those made by the VKI Aeronautics and Aerospace Department, which are carried out in many cases with prestigious entities such as NASA, ESA, Airbus Defense, etc. In addition, it is also interesting to comment the new spacecraft in development by Sierra Nevada Corporation and operated by NASA: the Dream Chaser Cargo System. The design of this new space vehicle may be familiar to that of the Space Shuttle. However, it uses an ablative thermal shield created by NASA's Ames Research Center that would be replaced as a large group rather than tile by tile, and would need to be replaced only after several flights [8], [9].

Ablation is actually studied by means of material characterization in ground-test facilities and then by using numerical models and codes developed to predict the material response at certain condition. Experimental ablation studies will be explained in this section to understand the characteristics of the facilities in which the photogrammetric measurement technique will be used.

The experiments carried on for TPMs are mainly focused on the characterization of the material at mission conditions. The data obtained in the experiments is then used in reduced

engineering models so that the material's behavior during the whole mission can be predicted. However, the conditions to be simulated would require such a huge amount of energy that make complete ground-test experiments prohibitively expensive. Nonetheless, there is a way to solve this problem and have realistic costs: having a experimental facility to simulate hypersonic speeds at low temperatures and a facility to reproduce high temperature flows but at subsonic speeds.

Hypersonic impulse facilities create a strong shock wave impinging on a test model, reproducing aerodynamic forces and compressible flow phenomena during several milliseconds. The different types of hypersonic facilities are well detailed by Chazot [10] and Pope [11]. In addition, the test times of this facilities are not enough to reproduce the chemical processes taking part in the atmospheric reentrys and plasma wind-tunnels are used to simulate them. On the other hand, plasma wind-tunnels can create continuous high temperatures for several minutes by means of electric power [12]. This facilities can use pure air at high enthalpies as test gas, what is really useful in order to simulate the areothermochemistry of an atmospheric entry. There are two main kinds of plasma wind-tunnels:

- **Arc-jets:** As the name of the facility suggests, the gas is heated by an arc generated between the cathode and the anode. The gas can be expanded to supersonic conditions and bigger models and higher heat rates are accomplished with respect to inductively coupled plasma generators. However, the erosion of the electrodes pollutes the flow and makes the characterization of the flow difficult.
- **Induction-coupled plasma (ICP):** In this facility the plasma torch is generated by means of electromagnetic induction thanks to a high-frequency induction coil. The flow speeds are generally subsonic, what allow the gases to relax towards thermochemical equilibrium and simplifying the computation of the boundary layer edge condition. The contribution of Helber in the reproduction of reentry plasmas in the von Karman Institute for Fluid Dynamics [13] is the main influence of this project .

2.2 Digital cameras: history and configuration

The data analyzed during the project comes from pictures taken with digital cameras. Therefore, it is necessary to learn about them and obtain an extensive knowledge about their configuration in order to obtain valuable images that allow an accurate calibration of the stereo camera and reconstruction of the probe along the experiment.

2.2.1 History and evolution of the cameras

Camera obscura is an optical instrument used to project an external image onto the inner surface of the device. The methodology used in these devices was developed on early stages in closed dark rooms with a small hole in one of the walls. The light rays enter the room through this hole, reflecting the external objects on the opposite wall. The projected image is vertically and horizontally inverted because the hole works as a converging lens [14].

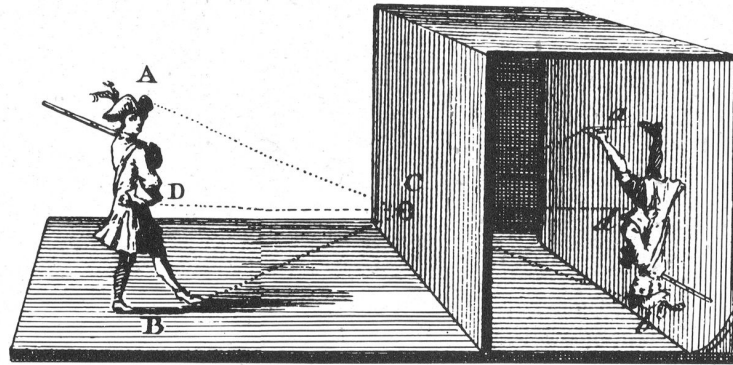


Figure 2.3: Camera obscura working principle[R2].

At the beginning, the only way to conserve the image projected was drawing it in the projected surface. However, it was discovered that some substances darkened when exposed to sunlight [15]. In fact, Johann Heinrich Schulze published in 1727 that the darkening of the salts was due to light alone, and not influenced by exposure to air or heat [16]. Thomas Wedgwood was the first person to use this chemical procedure to capture images. However, these images were not permanent as he did not use a fixing mechanism and the whole surface of the images darkened when exposed to light [15].

The first permanent photograph of history was taken by Joseph Nicéphore Niépce in 1826, using a wooden box camera made in Paris. This image shows the view through his window, and he needed 8 hours of exposure to obtain it [16].

After many years of research about fixing images, the photographic film was invented by George Eastman in 1885. His first camera, the “Kodak”, was released in 1888. Films also allowed to capture the movement, which led to the creation of cinematography and the movie industry [17].

The single-lens reflex camera (SLR) was not invented until the middle of the 20th century. This device uses a mirror to redirect the light taken by the lens to the viewfinder in order to focus and place the image. When the shutter is released, the mirror swings up and all the light is exposed towards the film in order to capture the photograph [16].

An engineer of Eastman Kodak named Steven Sasson was the inventor of the first electronic

camera in 1975. This self-contained device was used for science, defense and medicine in early stages. In 1986 Nikon company created the first digital single-lens reflex (DSLR) camera. Along the 2000s, full-frame DSLR cameras became the most popular option, replacing photographic cameras and expanding to the average consumer.

2.2.2 Camera parameters and configuration

In this section the most important camera parameters will be explained, as well as it will be learned how to set up a camera properly in order to have the best quality of image for the experiments. The reference for this subsection is the webpage *Cambridge in Colour* [18], which contains a lot of tutorials and deeply explained information for beginners and experts in photography.

2.2.2.1 Camera exposure

Camera exposure will determine the light level of a picture when taken by the camera. Exposure depends on three camera settings: aperture, ISO and shutter speed. These parameters determine the "exposure triangle".

- **Aperture:** controls the area over which light can enter your camera.
- **Shutter speed:** controls the duration of the exposure.
- **ISO speed:** controls the sensitivity of your camera's sensor to a given amount of light.

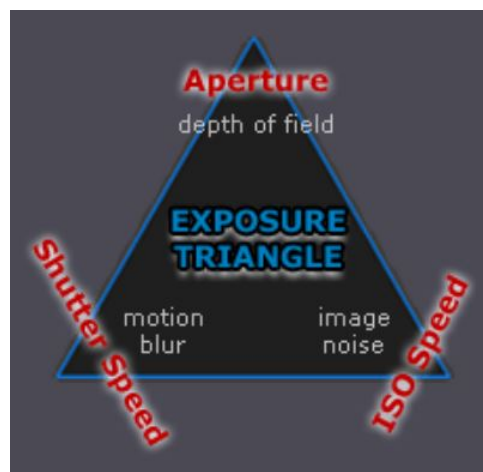


Figure 2.4: The exposure triangle [18].

As exposure depends on the combinations of all the three parameters, one can use many combinations of the above three settings to achieve the same exposure. Moreover, each setting influences other image properties. Therefore, knowing which trade-offs have to be made is

essential in order to obtain a quality image.

Aperture

Aperture is the setting that determines the area over which light will enter through the camera lens. The light-collecting area quadruples each time the f-stop value halves. The aperture setting determines a photo's depth of field (the range of distance over which objects appear in sharp focus). Lower f-stop values correlate with a shallower depth of field.

In the project it is desirable to have a sharp image of the object that will be reconstructed as well as the calibration pattern in every position. Therefore, a considerable depth of field is needed.

Shutter speed

The shutter speed specifically refers to how long light is permitted to enter the camera. When the exposure time doubles the amount of light entering the camera doubles.

Regarding our calibration, both the cameras and the calibration pattern will always be fixed in position and time. Therefore, shutter speed can acquire any desirable value in order to adjust aperture and ISO sensitivity and obtain the best image possible. In other words, shutter time is not a constraint during calibration.

Regarding the application in the Plasmatron test, taking pictures of the sample with a high frequency implies low shutter times. However, the luminosity level is extremely high, what makes sure that the aperture and shutter time needed will be very low.

ISO speed

ISO speed determines how sensitive the camera is to incoming light. Similar to shutter speed, it also correlates 1:1 with how much the exposure increases or decreases. A lower ISO speed is almost always desirable, since higher ISO speeds dramatically increase image noise. As a result, ISO speed is usually only increased from its minimum value if the desired aperture and shutter speed aren't otherwise obtainable.

2.2.2.2 Digital camera image noise

"ISO speed" is a standard which describes its absolute sensitivity to light. Higher numbers represent greater sensitivity and the ratio of two ISO numbers represents their relative sensitivity, meaning a photo at ISO 200 will take half as long to reach the same level of exposure as one taken at ISO 100.

A higher ISO setting makes your sensor more sensitive to light, meaning that you can take photos in darker conditions without the need to use a flash or tripod. However, a high ISO also creates more noise, reducing the image quality. Noise is the digital equivalent of film grain, and

it shows up as lots of tiny colored dots, which are particularly noticeable in the darker areas of your photo.

In the calibration of our project using a fixed camera or a tripod may be a better option than increasing ISO because both will help you to shoot in low-light conditions, but without deteriorating image quality.

2.2.2.3 Diffraction

Diffraction is an optical effect which limits the total resolution of your photography independently on how many megapixels your camera may have. It happens because light begins to disperse or "diffract" when passing through a small opening (such as your camera's aperture).

Diffraction becomes more significant as the size of the aperture decreases relative to the wavelength of light passing through, but occurs to some extent for any aperture or concentrated light source.

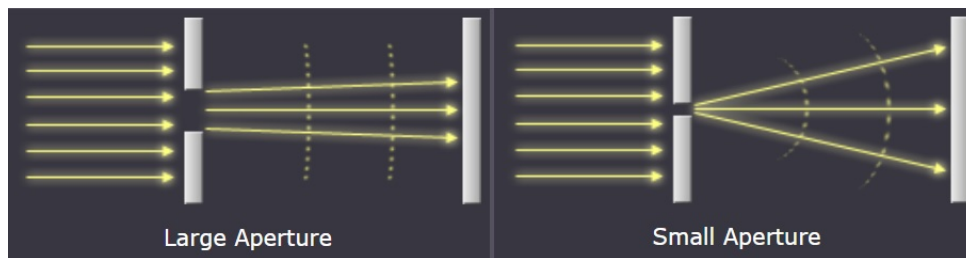


Figure 2.5: Diffraction effect depending on the aperture [18].

For an ideal circular aperture, the 2-D diffraction pattern is called an "airy disk," after its discovering by George Airy. The width of the airy disk is used to define the theoretical maximum resolution for an optical system (defined as the diameter of the first dark circle).

When the diameter of the airy disk's central peak becomes large relative to the pixel size in the camera (or maximum tolerable circle of confusion), it begins to have a visual impact on the image. Once two airy disks become any closer than half their width, they are also no longer resolvable (Rayleigh criterion).

The calculation of the Airy Disk can be performed via the following equation [R9].

$$C_{\text{Airy}} = 2.44 \cdot N \cdot \omega \quad (2.1)$$

Being ω the light wavelength (assuming 0.00055 mm as a nominal value) and N the aperture value of the camera.

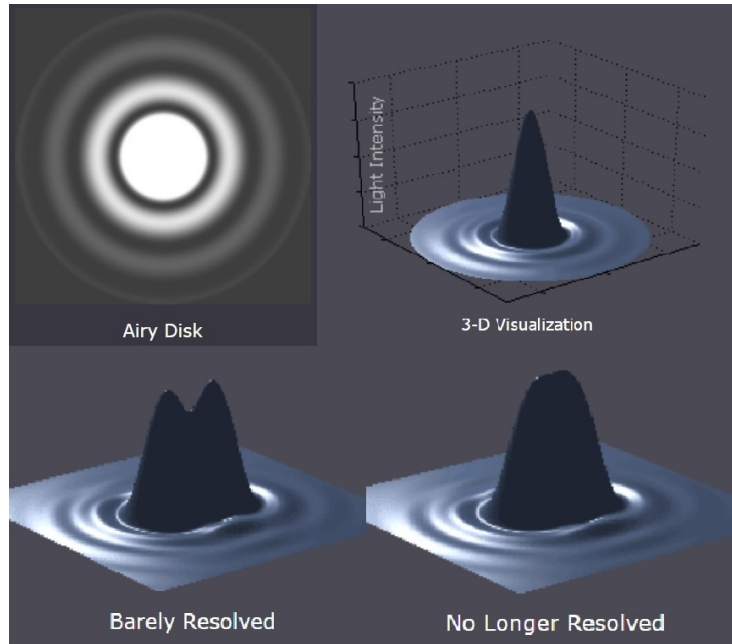


Figure 2.6: Definition of airy disk and example of image resolution [18].

As a result of the sensor's anti-aliasing filter (and the Rayleigh criterion above), an airy disk can have a diameter of about 2-3 pixels before diffraction limits resolution (assuming an otherwise perfect lens). However, diffraction will likely have a visual impact prior to reaching this diameter.

A limit in which diffraction is prohibitive can be established in order to have high quality photographs. The Airy disk must be smaller than the circle of confusion and 3 times the pixel size. Moreover, having an Airy disk smaller than one pixel is desirable.

2.2.2.4 Digital sensor size

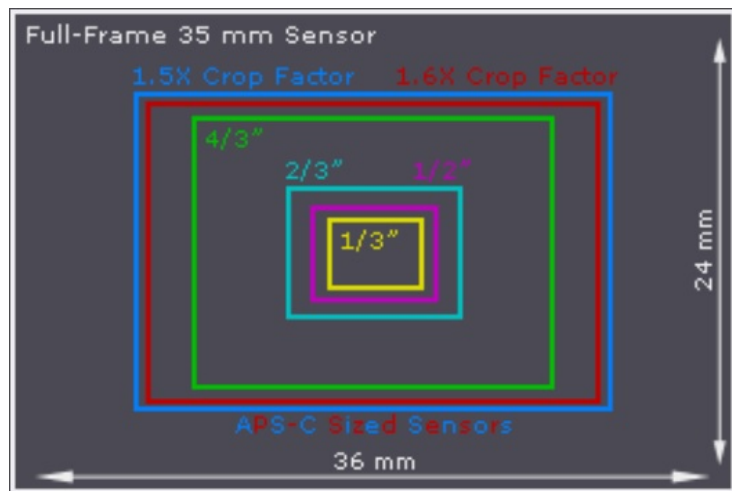


Figure 2.7: DSLR camera sensor size [18].

The crop factor is the sensor's diagonal size compared to a full-frame 35 mm sensor. One might initially think that throwing away image information is never ideal, however it has some advantages. Nearly all lenses are sharpest at their centers, while quality degrades progressively toward to the edges. This means that a cropped sensor effectively discards the lowest quality portions of the image, which is quite useful when using low quality lenses.

$$\text{Focal length}_{\text{effective}} = \text{Crop factor} \cdot \text{Focal length}_{\text{theoretical}} \quad (2.2)$$

The crop factor can be calculated from the sensor size:

$$\text{Crop factor} = \frac{\sqrt{36^2 + 24^2}}{\sqrt{\text{width}^2 + \text{height}^2}} \quad (2.3)$$

2.2.2.5 Depth of field

Depth of field refers to the range of distance that appears acceptably sharp. It varies depending on camera type, aperture and focusing distance. The depth of field does not abruptly change from sharp to unsharp, but instead occurs as a gradual transition. In fact, everything immediately in front of or in back of the focusing distance begins to lose sharpness — even if this is not perceived by our eyes or by the resolution of the camera.

Note that focal length has not been listed as influencing depth of field, contrary to popular belief. Even though the total depth of field is virtually constant, the fraction of the depth of field which is in front of and behind the focus distance does change with focal length.

Circle of confusion

Since there is no critical point of transition, a more rigorous term called the "circle of confusion" is used to define how much a point needs to be blurred in order to be perceived as unsharp. When the circle of confusion becomes perceptible to our eyes, this region is said to be outside the depth of field and thus no longer "acceptably sharp".

An acceptably sharp circle of confusion is loosely defined as one which would go unnoticed when enlarged to a standard 8 x 10 inch print (0.25 x 0.2 m), and observed from a standard viewing distance of about 1 foot. In this standard the CoC diameter is 0.25mm, which is the maximum size for a circle to be seen as a point by the human eye.

The circle of confusion is normally provided as a specification of the camera and some webpages provide its value for the most popular camera models [19]. The CoC depends on the sensor size and is calculated using the following methodology:

- An image with a size of 0.25 x 0.2 m has a diagonal of size:

$$d_i = \sqrt{25^2 + 20^2} = 32.015\text{cm} \quad (2.4)$$

- The diagonal of the sensor is calculated for a full frame camera:

$$d_s = \sqrt{3.6^2 + 2.4^2} = 4.327\text{cm} \quad (2.5)$$

- Taking into account that the size of the CoC in the printed 0.25 x 0.2 m is 0.25 mm, we can calculate the CoC in our full frame sensor:

$$CoC_s = CoC_i \cdot \frac{d_s}{d_i} = 0.25 \cdot \frac{43.27}{320.015} = 0.034\text{mm} \quad (2.6)$$

Depth of focus and aperture

Another implication of the circle of confusion is the concept of depth of focus (also called the "focus spread"). It differs from depth of field because it describes the distance over which light is focused at the camera's sensor, as opposed to the subject:

Using the smallest aperture in order to have the larger depth of field seems to be the best option. However, other than the fact that this may require prohibitively long shutter speeds without a camera tripod, small apertures softens the image by creating a larger circle of confusion (or "Airy disk") due to diffraction (even within the plane of focus). Diffraction quickly becomes more of a limiting factor than depth of field as the aperture gets smaller.

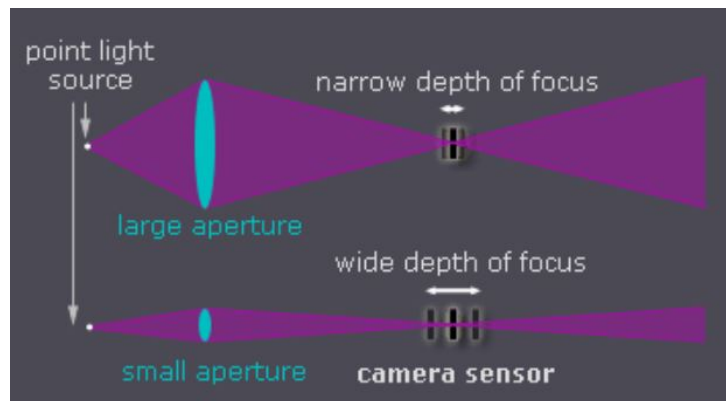


Figure 2.8: Diagram depicting depth of focus versus camera aperture [18].

Hyperfocal distance

Definition 1: The hyperfocal distance is the closest distance at which a lens can be focused while keeping objects at infinity acceptably sharp. When the lens is focused at this distance, all objects at distances from half of the hyperfocal distance out to infinity will be acceptably sharp.

Definition 2: The hyperfocal distance is the distance beyond which all objects are acceptably sharp, for a lens focused at infinity.

$$H = \frac{f^2}{N \cdot c} + f \quad (2.7)$$

Where H is the hyperfocal distance, f is the focal length, N is f-number and c is the circle of confusion limit. Therefore, if f and N are calculated in order to place the target object at the hyperfocal distance and focus it, sharpness from half the distance of the object until infinity will be ensured.

2.3 Photogrammetry

Once the physical nature of the experiments is understood it is necessary to study how to implement a stereophotogrammetric measurement technique. Computer vision will be the principal tool for processing the information and obtain three-dimensional data. There are some theory about the physics of the problem that should be addressed in order to program a stereophotogrammetry software.

2.3.1 Camera model

As said in section "Digital cameras history and configuration" pinhole camera model is the most widely extended one. Now it is the moment to explain the physics of this model in detail. As it can be seen in Figure 2.9, a pinhole (the pinhole aperture) lets through only those light rays that intersect a particular point in space; these rays then form an image by "projecting" onto an image plane.

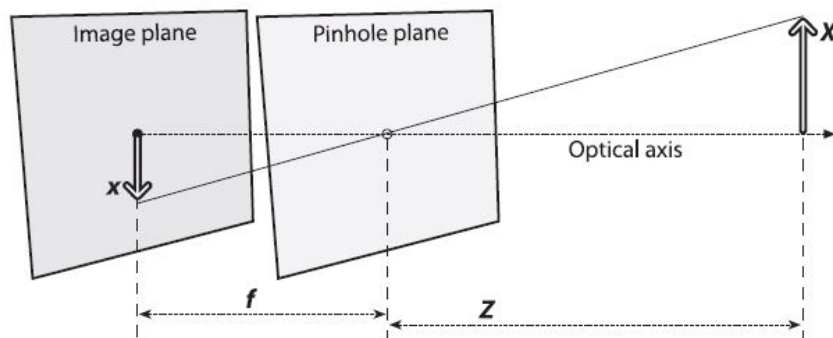


Figure 2.9: Pinhole camera model [20].

The image projected onto the image plane (can be also called projective plane) is always in focus. The size of the projected image is determined by the focal length f (distance from the pinhole aperture to the image plane). As it can be seen in figure 2.9 the size x can be determined by similar triangles by the following equation being f the focal length, x the object's size on the projective plane, Z the distance from the camera to the object and X the real size of the object:

$$-x = f \cdot \frac{X}{Z} \quad (2.8)$$

In the ideal camera model it is obvious that the sensor of the camera will be in the image plane and that its center will lie in the optical axis. However, it is known that manufacturing is not perfect, and it is possible to have a certain displacement between the center of the sensor and the intersection of the sensor with the optical axis. Therefore, it is necessary to define this intersection as the principal point c , which will have two components: c_x and c_y .

$$x_{\text{sensor}} = f_x \cdot \left(\frac{X}{Z}\right) + c_x \quad , \quad y_{\text{sensor}} = f_y \cdot \left(\frac{X}{Z}\right) + c_y \quad (2.9)$$

In addition, it is necessary to consider that a camera may not have perfect squared pixels. In order to take this effect into account f_x and f_y will be defined as the product of the physical focal length of the lens F in mm and the size s_x and s_y respectively of the individual imager elements in pixels per mm. It is important to remark that neither the physical focal length F nor the size of the pixels s_x - s_y can be measured in real life. Only the combinations f_x - f_y can be obtained without measuring the components directly by dismantling the camera.

$$f_x = F \cdot s_x \quad , \quad f_y = F \cdot s_y \quad (2.10)$$

2.3.2 Lens distortion

As it has been remarked previously, real world manufacturing cannot be considered equal to the ideal model. For example, it is easier to create a spherical lens than an ideal parabolic one and is it difficult to align perfectly the sensor and the lens. These are the reasons of the radial distortion, due to the shape of the lens, and tangential distortion, due to the assembly process.

2.3.2.1 Radial distortion

When the light rays bend more near the edges of a lens than in its optical center is because of the effect of radial distortion. In most of the cases the distortion is greater when the lens is smaller. There are two main types of radial distortion effects; the negative radial distortion (pincushion) and the positive radial distortion (barrel).

Knowing that radial distortion is 0 at the optical center and grows towards the corners it is possible to characterize it by using terms of a Taylor series expansion around $r = 0$. The first two terms are called k_1 and k_2 and are normally used to model normal cameras. If the camera induces a higher distortion it is also possible to use a third term k_3 . The equations of the mathematical model used to characterize the distortion are the following:

$$x_{\text{corrected}} = x \cdot \left(1 + k_1 r^2 + k_2 r^4 + k_3 r^6\right) \quad (2.11)$$

$$y_{\text{corrected}} = y \cdot \left(1 + k_1 r^2 + k_2 r^4 + k_3 r^6\right) \quad (2.12)$$

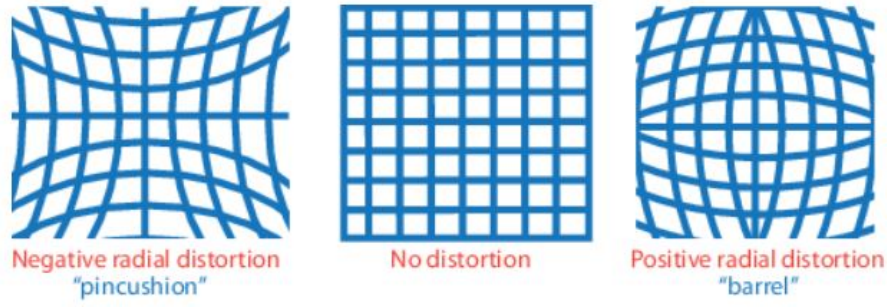


Figure 2.10: Radial distortion effect [21].

Where x and y are the undistorted pixel locations and r is the norm of the \vec{xy} vector.

2.3.2.2 Tangential distortion

The second distortion is the tangential one, which is induced by the non-parallelism of the lens with the sensor due to a lack of accuracy in the manufacturing process. This distortion is characterized by two coefficients; p_1 and p_2 .

$$x_{\text{corrected}} = x + \left[2p_1y + p_2 \left(r^2 + 2x^2 \right) \right] \quad (2.13)$$

$$y_{\text{corrected}} = y + \left[p_1 \left(r^2 + 2y^2 \right) + 2p_2x \right] \quad (2.14)$$

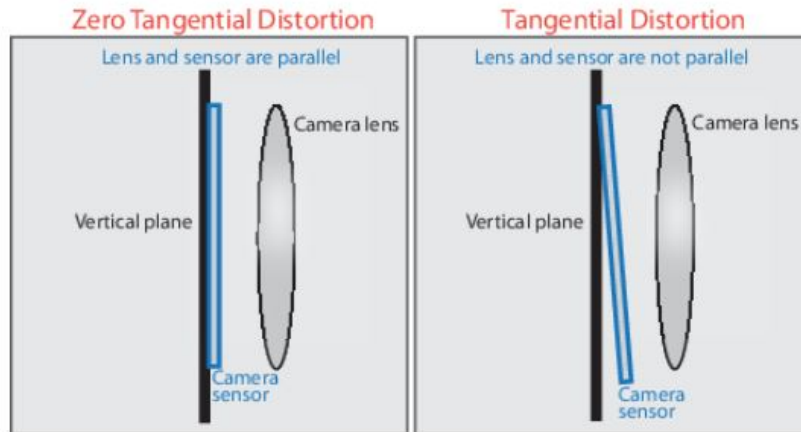


Figure 2.11: Tangential distortion effect [21].

2.3.3 Projective geometry

The camera model parameters and the distortion correction can now be used in order to match physical world points (X, Y, Z) with the points on the projection screen with coordinates (x, y) . This relation is a projective transform, and in such transforms it is useful to use homogeneous

coordinates. The projection of physical world points into the camera is made as follows:

$$q = MQ, \text{ where } q = \begin{bmatrix} x \\ y \\ w \end{bmatrix}, \quad M = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}, \quad Q = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \quad (2.15)$$

Where q is the homogeneous vector of camera coordinates, M is the **camera intrinsics matrix**, and Q is the homogeneous vector of physical world points in camera coordinates.

2.3.4 Camera calibration

Once the intrinsic and distortion properties of the cameras have been mathematically understood it is time to explain how they are calculated in computer vision. The world points are transformed to camera coordinates using the extrinsics parameters. The camera coordinates are mapped into the image plane using the intrinsics parameters. The extrinsic parameters represent the location of the camera in the space and consist on a rotation, R , and a translation, t . The center of the camera's coordinates system is the principal point. The main reference of this subsection and for the implementation of the future code is the book *Learning OpenCV: Computer Vision with the OpenCV* written by Bradski and Kaehler [20].

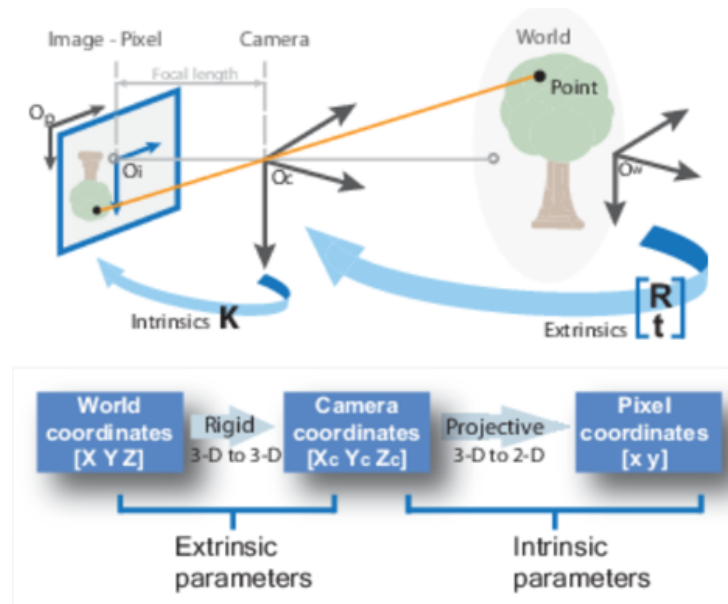


Figure 2.12: Change from world to pixel coordinates process [21].

$$\begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} = \begin{matrix} \text{Intrinsics (Camera matrix)} \\ \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \end{matrix} \begin{matrix} \text{Extrinsics} \\ \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \end{matrix} \begin{bmatrix} X_o \\ Y_o \\ Z_o \\ 1 \end{bmatrix} \quad (2.16)$$

2.3.4.1 Camera calibration methods

There are different methodologies in order to calibrate a camera. In the paper made by Guerchouche [22] a comparison of three of them is made.

- Faugeras and Toscani [23], [24] method proposal consist on an estimation of the Perspective Projection Matrix (PPM) using an image of a non-planar pattern. One image and at least six noncoplanar feature 3D points, manually selected or automatically detected on the acquired image, are needed.
- The algorithm detailed in [25] for camera calibration is based on a robust estimation of the PPM. The target used is a 3D cube with different colored faces. The six vertices associated to the faces of the cube are calculated and the the PPM is calculated using the Faugeras-Toscani algorithm.
- Zhang [26], [27] described an algorithm which requires at least two different views of a planar pattern. An even more accurate calibration is obtained using a large number of views (twenty or more). The displacements of the pattern between the views are not necessarily known.

In this project it has been decided to use Zhang technique as it is more flexible and can be easily implemented. A 2D calibration pattern is much more easily manufactured and needs less space than a 3D one. A detailed description of Zhang's technique and the algorithms used by it can be found in [27]. The proposed procedure consists of a closed-form solution, followed by a nonlinear refinement based on the maximum likelihood criterion. Both computer simulation and real data have been used to test the proposed technique, and very good results have been obtained.

2.3.4.2 Calibration procedure

Now that the technique that will be used has been explained, it is time to detail the process taking part for each one of the pictures taken to the 2D calibration pattern. Regarding the extrinsic parameters, the rotation and translation will need three parameters for each one to be solved. As the intrinsic matrix for a camera has four parameters (f_x, f_y, c_x, c_y), there is a total of ten parameters to be solved in each view (but note that the camera intrinsic parameters stay the same between views).

Using a planar object implies that just eight parameters are fixed each view. Because the six parameters of rotation and translation change between views, for each view we have constraints on two additional parameters that we use to resolve the camera intrinsic matrix. Therefore, at least two views will be needed in order to solve all the geometric parameters [20].

2.3.4.3 Calibration tools

The calibration object used in this project is a chessboard. In principle, any characterized object could be used, but the regular shape of the chessboard and the fact that its corners can be easily detected in computer vision makes the process easier. Moreover, the shape of the squares is regular, and then the position of each one of the corners is previously known. These corners will be positioned in the image and then they will be used to obtain the calibration parameters and the model the distortion (as the chessboard lines are straight it is easier to compute distortion using this patterns). It is important to have a considerable amount of points covering the whole field of view of the cameras in order to model distortion and obtain calibration parameters correctly.

2.3.4.4 Calibration calculation

Now that all the parameters have been understood it is time to show how they are calculated. It has been explained that there are four intrinsic parameters (f_x, f_y, c_x, c_y), five distortion parameters: two tangential (p_1, p_2) and three radial (k_1, k_2, k_3). Intrinsic parameters are directly tied to the 3D geometry (and hence the extrinsic parameters) of where the chessboard is in space; distortion parameters are tied to the 2D geometry of how the pattern of points gets distorted, so we deal with the constraints on these two classes of parameters separately.

For the extrinsic parameters it is mandatory to know where the chessboard is. This will require three rotation parameters and three translation parameters for a total of six per view of the chessboard, because in each image the chessboard will move. Three corner points in a known pattern yielding six pieces of information are (in principle) all that is needed to solve for our five distortion parameters (of course, much more are used for robustness). Thus, one view of a chessboard is all that we need to compute our distortion parameters. Together, the four intrinsic and six extrinsic parameters make for ten altogether that we must solve for each view.

More than one image is needed as a 2D chessboard is used for calibration, which implies that a homography can yield at most eight parameters from four (x, y) pairs. This is because only four points are needed to express everything that a planar perspective view can do. So, no matter how many corners are detected on a plane, only four of them provide useful information. Per chessboard view, then, the equation can give us only four corners of information. This implies that two views of a 3-by-3 chessboard (counting only internal corners) are the minimum that could solve our calibration problem. Consideration for noise and numerical stability is typically what requires the collection of more images of a larger chessboard. In practice, for high-quality results, at least ten images of a 7-by-8 or larger chessboard are needed (and that's only if the chessboard is moved enough between images to obtain a "rich" set of views). In order to get deeper information about the algorithms used to calculate the calibration parameters [R2] pages 384-396 detail the whole process.

2.3.5 Stereo camera imaging

Once the cameras are calibrated it is time to address the problem of using two of them to emulate stereo imaging capability, which is the one that allows humans to detect 3D objects. By using computer vision it is possible to detect correspondences between points seen by one camera and the same points seen by the other one. Knowing this correspondences and the relative orientation of one camera with respect to the other it is possible to compute the 3D location of the points. As well as in the previous subsection, [20] is the base of the explanations made below. Stereo imaging is performed by following these four basic steps when using two cameras:

1. **Undistortion** of the images by removing radial and tangential distortion.
2. Obtain the relative position of one camera with respect to the other. **Rectification** provides images that are now row-aligned (two image planes are coplanar and that the image rows are exactly aligned) and rectified.
3. Find the same features in both camera views (in the case of our project the typical notation for the imagers will be left and right camera). The output is a **disparity map**, where the disparities are the differences in x-coordinates on the image planes of the same feature viewed in the left and right cameras: $d = x_l - x_r$.
4. Once the geometric position of the cameras is known and the disparity map has been obtained it is possible to obtain the 3D position of the points by triangulation. The output of this process called reprojection is the **depth map**.

The last three points will be detailed along this point as undistortion has already been explained in the previous one. Moreover, in order to motivate the second and third step the last one will be explained first.

2.3.5.1 Triangulation

Assume that the experimental set is the one defined in Figure 2.13 undistorted, aligned and with a measured baseline T . Also, assume for now that the principal points c_x left and c_x right have been calibrated to have the same pixel coordinates in their respective left and right images and that focal lengths are equal. Moreover, it should be assumed that the images are coplanar and row-aligned, with exactly parallel optical axis.

In this simplified case, taking x_l and x_r to be the horizontal positions of the points in the left and right imager (respectively) allows us to define disparity simply by $d = x_l - x_r$. The depth Z can now be derived from disparity by using similar triangles.

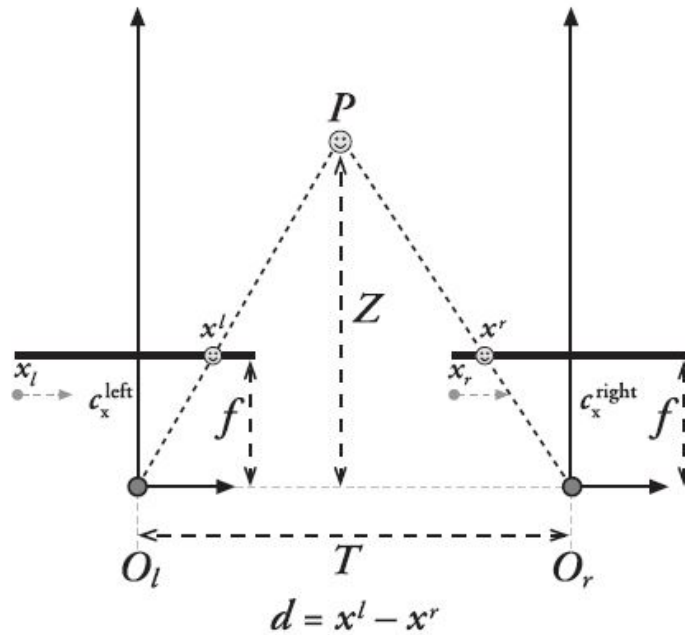


Figure 2.13: With a perfectly undistorted, aligned stereo rig and known correspondence, the depth Z can be found by similar triangles [20].

$$\frac{T - (x_l - x_r)}{Z - f} = \frac{T}{Z} \Rightarrow Z = \frac{fT}{x_l - x_r} \tag{2.17}$$

Looking deeper at these equations it can be seen that depth is inversely proportional to disparity, which implies that there is a nonlinear relationship between these two terms. This can be easily seen in Figure 2.14. The consequence of this effect is that stereo vision systems have high depth resolution only for objects relatively near the camera.

2.3.5.2 Rectification

The main objective of these process is to obtain an ideal arrangement from a real-world camera setup. As the imagers will almost never be exactly aligned as in the ideal case, it is necessary to mathematically find image projections and distortion maps in order to rectify the left and right camera into a frontal parallel arrangement. Of course, if the physical alignment is close to the ideal one, the mathematical transformations will be more tractable. Figure 2.15 shows the transformation process to be done.

Once the mathematical transform has been performed, the stereo coordinate system obtained will be the one for undistorted rectified cameras showed in Figure 2.16. It can be observed that it is a right-handed coordinate system, while in other references it is expressed as a left-handed one. The cameras have the origin of the pixels in the upper left corner, the center

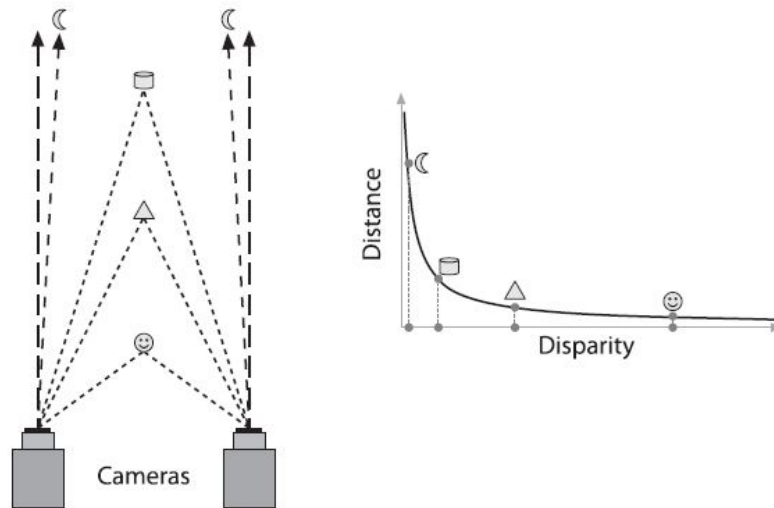


Figure 2.14: Depth and disparity are inversely related, so fine-grain depth measurement is restricted to nearby objects [20].

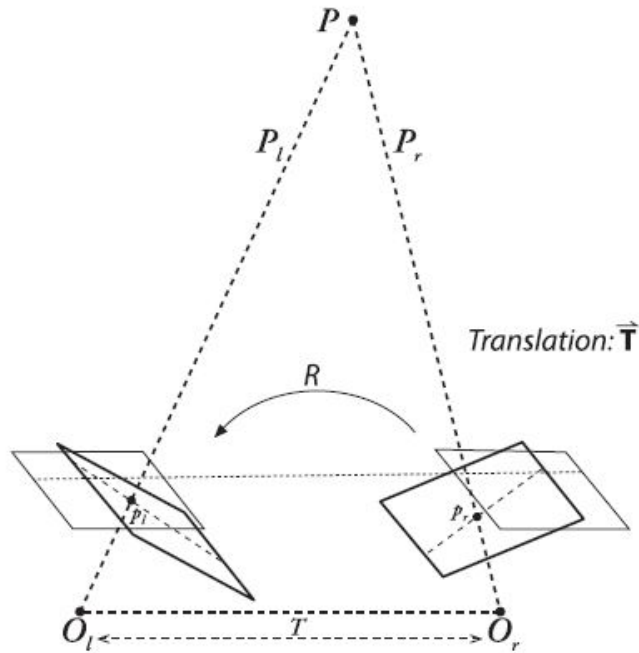


Figure 2.15: A mathematical alignment of the two cameras into one viewing plane so that pixel rows between the cameras are exactly aligned is desired [20].

of projection are at O_l and O_r with principal rays intersecting the image plane at the principal point (c_x, c_y) . After mathematical rectification, the cameras are row-aligned (coplanar and horizontally aligned), displaced from one another by T , and of the same focal length f .

2.3.5.3 Epipolar geometry

In order to perform the previously described mathematical alignment it is necessary to understand even more about the geometry of a stereo camera. The basic geometry of a stereo

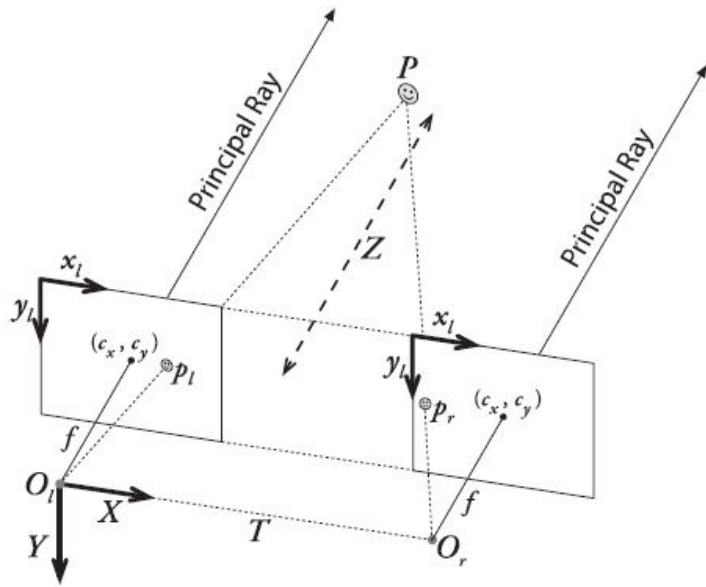


Figure 2.16: Stereo coordinate system used for undistorted rectified cameras: the pixel coordinates are relative to the upper left corner of the image, and the two planes are row-aligned [20].

imaging system is called epipolar geometry and it can be observed in Figure 2.17. Each camera has its center of projection (O_l and O_r) and projective plane. A real-world point P has a projection onto each projective plane (p_l and p_r). The epipoles (e_l and e_r) are the projection of the center of projection of the other camera (e_r is the projection of O_l in the right camera projective plane and then the right camera epipole). The lines $\overrightarrow{p_l e_l}$ and $\overrightarrow{p_r e_r}$ are called epipolar lines and the plane formed by P , e_l and e_r is called the epipolar plane.

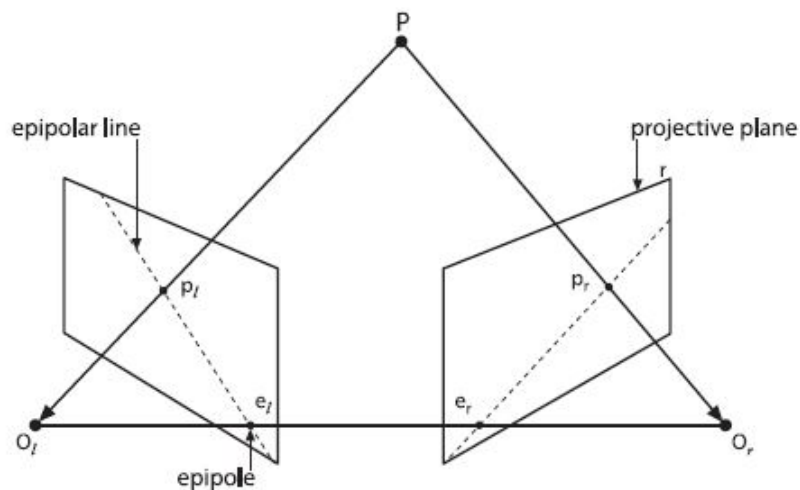


Figure 2.17: The epipolar plane is defined by the observed point P and the two centers of projection, O_l and O_r ; the epipoles are located at the point of intersection of the line joining the centers of projection and the two projective planes [20].

Now that epipolar geometry has been understood it is time to show why it is so important in

photogrammetry. It can be observed that every real-world point viewed by the two cameras will lie in an epipolar plane that intersects each image in an epipolar line. Therefore, given a feature in one image, its corresponding view in the other image must lie along the corresponding epipolar line. This is known as the epipolar constraint, and it implies that what was going to be a two-dimensional search for matching features has become a one-dimensional search along the epipolar lines. This is extremely computational saving and useful to reject incorrect matches. Of course, it is mandatory to know the epipolar geometry and the spatial distribution of the cameras in order to perform the one-dimensional search.

2.3.5.4 The essential and fundamental matrices

The essential matrix E and the fundamental matrix F will provide the information about the position of the cameras in the physical space. Matrix E contains the information about the rotation and translation of one imager with respect to the other, and matrix F contains the same information but also embedding information about the intrinsics of both cameras. Therefore, matrix E relates the camera position in physical coordinates and does not provide information about the cameras, while matrix F gives the camera position taking into account their intrinsics in pixel coordinates.

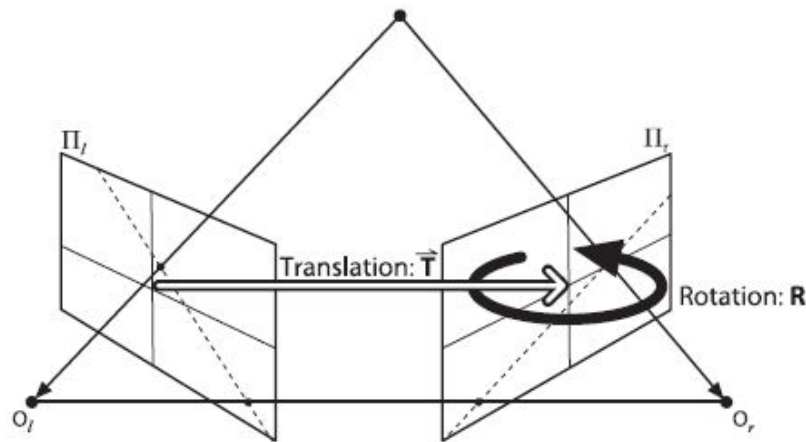


Figure 2.18: The essential geometry of stereo imaging is captured by the essential matrix E , which contains all of the information about the translation T and the rotation R [20].

Essential matrix

The essential matrix E is defined as the one that relates points p_l and p_r (projection of real world point P in both cameras). As it has been already seen, this points are related by using epipolar geometry. Then, it is time to derive the expression of the essential matrix in order to show its mathematical explanation.

The left camera will be chosen as the center of coordinates, but it works completely equal with the right camera. Taking O_l as the center of coordinates, the location of the observed point P is P_l , and the origin of the right camera O_r is known thanks to the translation and rotation

vectors T and R (Figure 2.15). Therefore, the location of the point P_r (P seen by the right camera) can be defined as $P_r = R(P_l - T)$.

Now it is time to introduce the epipolar constraint. There are several ways to represent a plane in mathematics, but in this case the following will be used:

$$(x - a)n = 0 \quad (2.18)$$

Where all points c on a plane with a normal vector n and passing through point a follow the previous constraint. Then, as the epipolar plane contains vectors T and P_l , this mathematical constraint can be used for our purpose as following:

$$(P_l - T)^T (T \times P_l) = 0 \quad (2.19)$$

Knowing that $R_t = R^{-1}$, the equality that defines P_r can be rewritten:

$$P_r = R(P_l - T) \Rightarrow (P_l - T) = R^{-1}P_r \quad (2.20)$$

This substitution yields to the following expression for equation 2.19:

$$(R^T P_r)^T (T \times P_l) = 0 \quad (2.21)$$

$$T \times P_l = SP_l \Rightarrow S = \begin{bmatrix} 0 & -T_z & T_y \\ T_z & 0 & -T_x \\ -T_y & T_x & 0 \end{bmatrix} \quad (2.22)$$

This leads to the first result and the definition of the E matrix:

$$(P_r)^T R S P_l = 0 \Rightarrow (P_r)^T E P_l \quad (2.23)$$

Finally, substituting by the projection equations $p_l = f_l P_l / Z_l$ and $p_r = f_r P_r / Z_r$ and then divide the whole thing by $Z_l Z_r / f_l f_r$ it is possible to obtain the final result:

$$p_r^T E p_l = 0 \quad (2.24)$$

Fundamental matrix

When applying photogrammetry theory to computer vision applications it is much more interesting to use pixel coordinates instead of physical ones. In order to transform the physical information into pixel coordinates it is necessary to include the data about the intrinsics of the cameras. It should be reminded that a point in physical coordinates p is expressed in pixel coordinates q by means of the camera matrix M . Therefore, knowing that $q = M \cdot p$ it is possible to include this information in equation 2.24 in order to rewrite it:

$$q_r^T (M_r^{-1})^T E M_l^{-1} q_l = 0 \quad (2.25)$$

Then, the definition of the fundamental matrix F is the following:

$$F = (M_r^{-1})^T E M_l^{-1} \Rightarrow q_r^T F q_l = 0 \quad (2.26)$$

2.3.5.5 Summary

Now that all the theory has been explained a brief summary of its application in common stereophotogrammetry software will be done:

1. The images of both cameras are undistorted.
2. The camera matrix of both imagers is obtained by camera calibration.
3. The rotation and translation vectors of one camera with respect to the other is obtained by performing a stereo camera calibration. The essential and fundamental matrices are also calculated in this process.
4. The stereo camera is rectified, providing row-aligned images.
5. By using the epipolar constraint and the fundamental matrix it is possible to match the features seen by one camera with its corresponding features in the other image.
6. The disparity can be calculated: $d = x_l - x_r$.
7. Finally the three-dimensional information of the points can be calculated by triangulation, obtaining the depth map.

3 Experimental tools

This chapter details the experimental tools used in this project in order to characterize the ablation process. Firstly, a description of the Plasmatron is made, presenting the applications of the facility and how it is used to characterize the flow field. Then, the experimental tools used to measure the surface recession are analyzed, defining the motivation of this project and the need of creating a new measurement technique in order to increase accuracy.

3.1 Experimental facilities: VKI Plasmatron

The Plasmatron is a VKI facility used for the reproduction of the aerothermodynamic environment of reentry flows. The recreation of this conditions implies a high-enthalpy, highly dissociated subsonic gas flow. A full scheme of the Plasmatron can be seen in Figure 3.1.

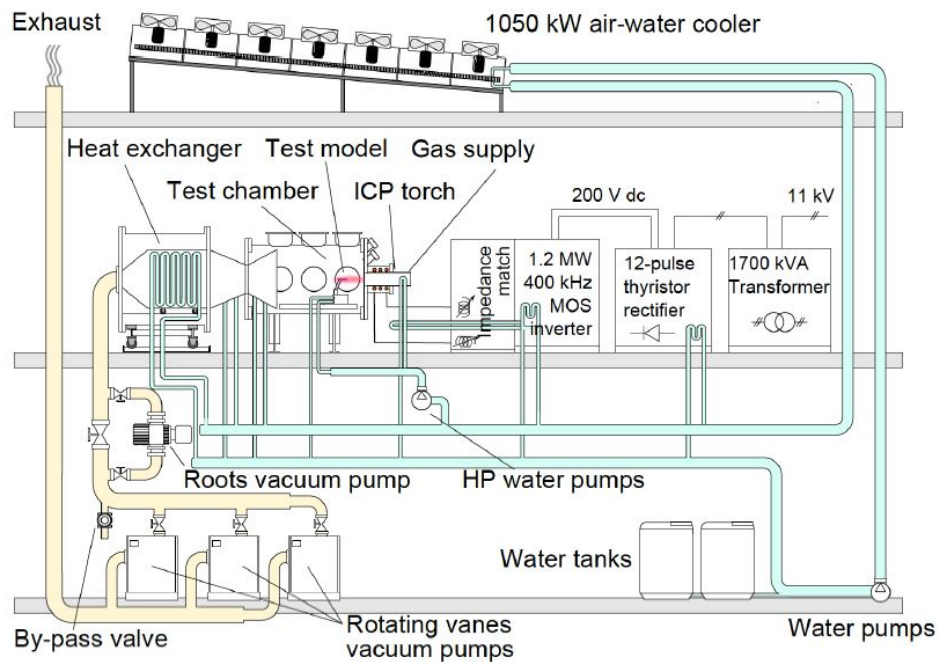


Figure 3.1: Schematic of the VKI Plasmatron system [13].

This facility is an inductively coupled plasma (ICP) wind tunnel, which means that the energy used to generate the flow conditions comes from electric currents created by electromagnetic induction. In the case of the Plasmatron, a high-power, high frequency (1.2MW, 400kHz) radio-frequency generator with MOS technology powers the inductor, which is a single turn, flat coil. The plasma torch starts with an annular gas injection into a 160mm inner diameter cold cage, that is placed inside of a 200mm diameter quartz tube surrounded by the

above mentioned coil. The initial injection is performed with argon in order to facilitate the ignition due to its low ionization energy. An initial spark created by the strong electric currents introduce free electrons into the torch that are accelerated by the electromagnetic field, colliding with the argon atoms and leading to further ionization. Once the hot gas is accelerated out of the torch in form of a plasma jet it is time to switch to the test gas (N_2 , CO_2 , air). When the new plasma jet reaches equilibrium the process runs uninterruptedly as long as enough supply gas, electricity and cooling are provided. One of the main advantages of this facility is the high purity of the plasma jet due to the absence of eroding electrodes as the system heats gas by electromagnetic induction.

The cooling system (1050kW) of this facility supplies a closed water loop and fan-driven air coolers to maintain the torch, test chamber, sample retention system and holding arms at a feasible temperature. The vacuum created in the test chamber reaches a minimum of 2Pa thanks to three rotatory-vane vacuum pumps. The Plasmatron systems are controlled using a 719 I/O lines PLC connected to a computer. This facility is further described by Bottin in his PhD thesis [28], as well as it is in Helber PhD [13], which is the main theoretical base of this project.

3.1.1 Reentry flow simulation

The aim of the VKI Plasmatron is to reproduce reentry flow conditions in order to perform thermal protection material (TPM) and debris decomposition tests, as it can be observed in Figure 3.2.

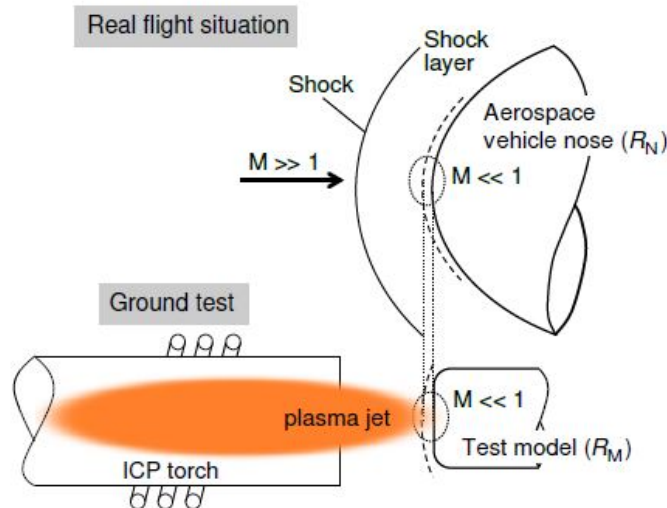


Figure 3.2: Use of ICP jet for TPM testing: Simplified hypersonic flow in real flight condition and subsonic plasma jet following the LHTS methodology, reproducing the stagnation point region [13].

A hypersonic reentry capsule has a stagnation point region characterized by a subsonic boundary layer reaching extremely high temperatures. These reentry flight conditions can be

recreated by ground-test facilities such as the Plasmatron, based on the Local Heat Transfer Simulation methodology developed by Kolesnikov [29] and adapted by Barbante et al. to the Plasmatron [30]. Kolesnikov stated that the stagnation point heat flux will be the same in flight and on ground if the pressure p_e , the boundary layer edge total enthalpy H_e , and the radial velocity gradient in radial direction at the wall are the same.

3.2 Surface recession experimental investigation

This section will describe the measurement technique that is actually being applied in order to measure surface recession in Plasmatron ablation tests. Once the existing tools are understood, a new way to measure surface recession will be proposed and discussed.

3.2.1 Current methodology: High-Speed Camera (HSC) imaging

The most simple way to measure the surface recession along the experiment is with caliper measurements. This methodology will only provide absolute, averaged recession values but will not give information about the steadiness of the process. The surface that wants to be evaluated is a brittle char layer, composed of small carbon fibers and carbonized phenolic resin. Due to the nature of the sample, the surface recession is difficult to evaluate and the use of intrusive tools may alter the surface, what is prohibitive if a microscopy test wants to be performed afterwards.

Therefore, optical methods are chosen to measure the surface recession in-situ without affecting the experiment, as it is a non-intrusive technique. A Vision Research Phantom 7.1 CMOS HSC was elected due to its short exposure times ($2\mu s$) and CMOS sensor that prevents CCD sensor saturation. A mm-resolved chessboard is placed at the sample location in order to have a mm:pixel calibration. The recession rate is retrieved as the slope of a linear regression, fitted to the illuminated pixels over time. Acquisition of the HSC was triggered using a Digital Delay Generator, synchronized with the OES measurements, before injection of the test sample. This allowed for precise determination of the total injected time [13].

This measurement technique has certain limitations that should be commented. If a 2D sample is being used and the camera is able to capture the whole surface, a two-dimensional surface recession measurement can be performed. However, studying three-dimensional samples is the most common and interesting case, and in these cases using one camera to measure surface recession can lead to incorrect results. In Figure 3.3 a hypothetical case with a 3D sample is exposed: the camera is placed with the lens and sensor perpendicular to the y-axis and the sample is a three-dimensional piece (the z-axis dimension is omitted in the sketch for simplicity). As it can be observed in Figure 3.4, when the surface moves in the -x direction there is an underestimation of the stagnation point movement as another point is being recorder by the camera instead of it. This effect can be also observed in Figures 3.5 and 3.6: in this case if Point 1 wants to be tracked along the experiment, the camera will show the position of Point 2 in reality. This effect becomes larger when the sample moves lot and when the f/d coefficient

grows, being f the focal length and d the distance from the camera to the sample.

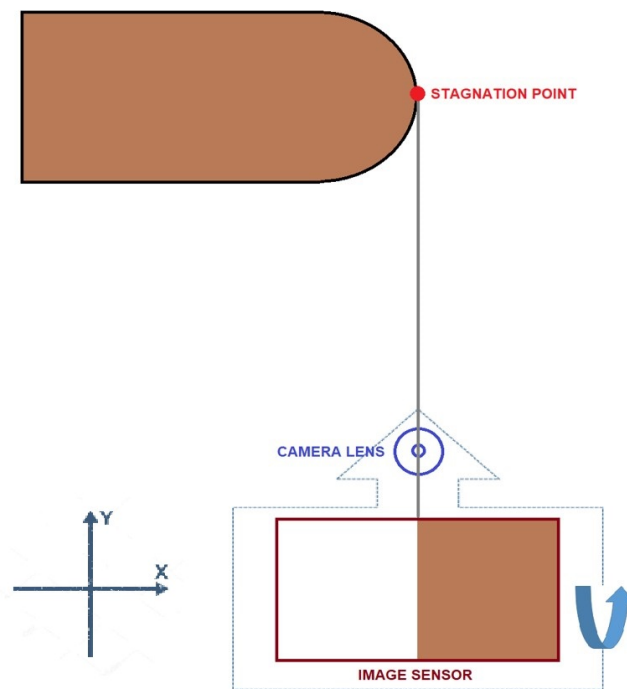


Figure 3.3: Surface recession optical measurement limitation.

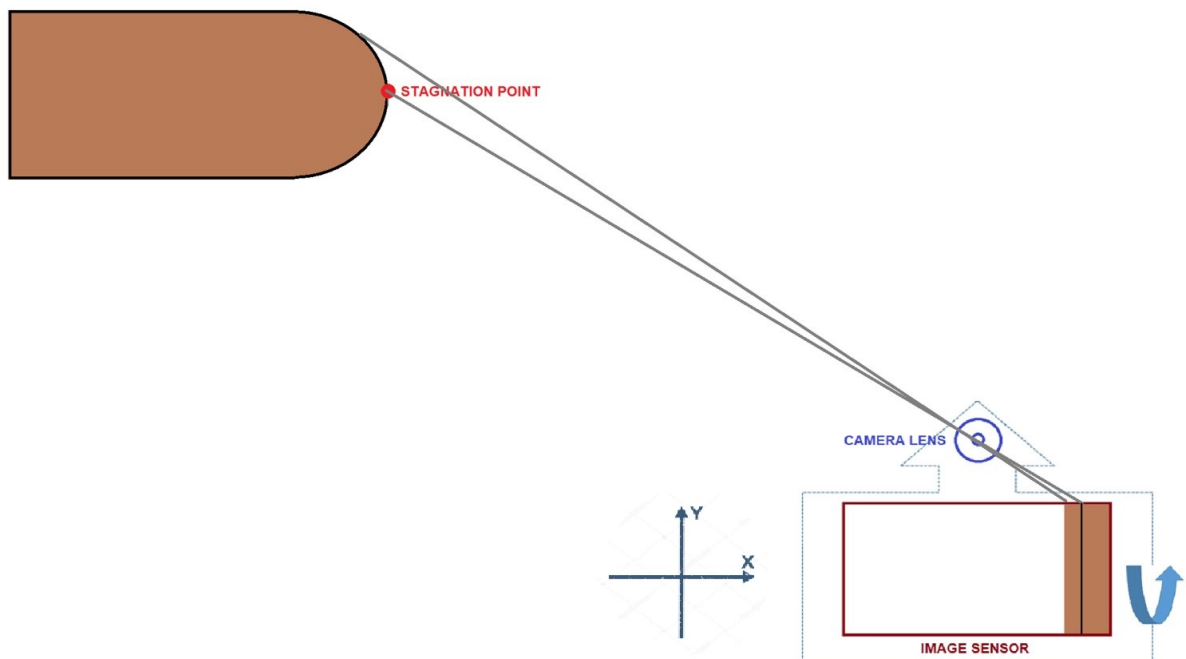


Figure 3.4: Surface recession optical measurement limitation.

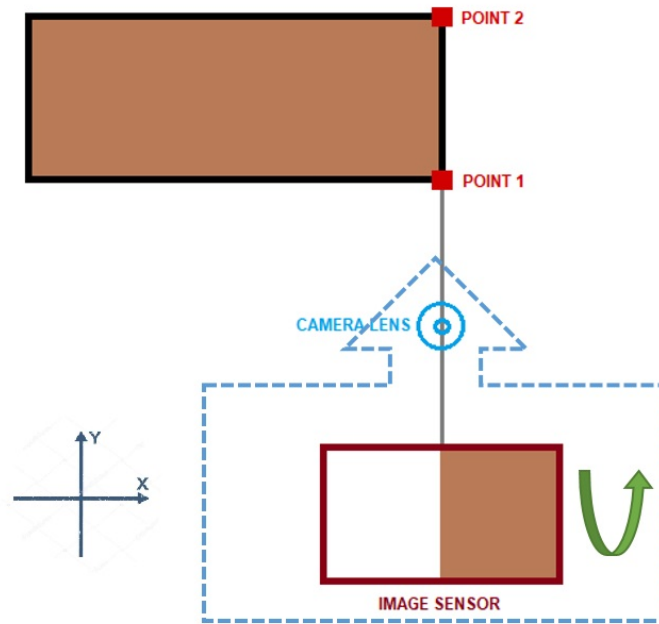


Figure 3.5: Surface recession optical measurement limitation.

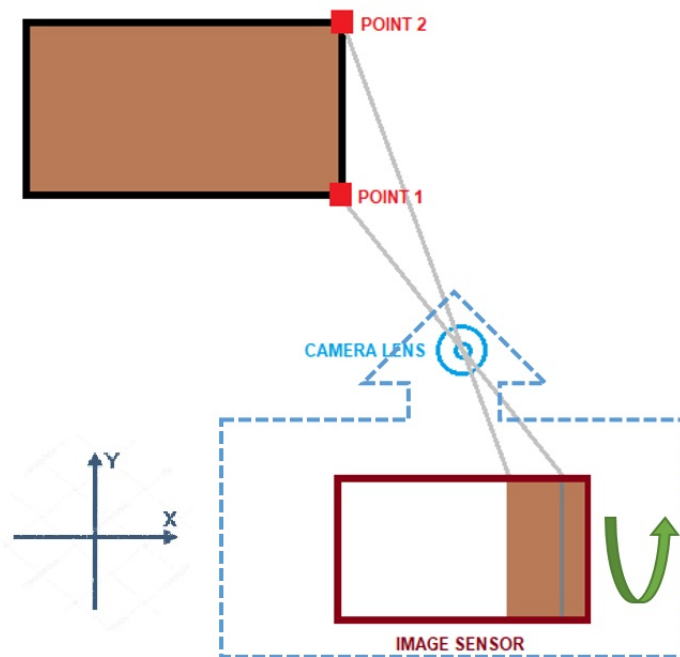


Figure 3.6: Surface recession optical measurement limitation.

3.2.2 Experimental methodology proposal: Stereophotogrammetry 3D reconstruction

It has been explained that using just one camera to measure the surface recession along the experiment has certain limitations (just 2D measurements). Then, the motivation of this project lies in the implementation of a new methodology based on photogrammetry to accurately measure the whole surface recession.

First of all, it is necessary to analyze the possibilities that the VKI Plasmatron has to implement this technique. As it can be observed in Figures 3.7 and 3.8, there are three optical accesses in the back of the Plasmatron (two on the top and one a little below) and one window in each side. However, it has to be taken into account that the optical accesses and windows have to be shared with the rest of measurement tools, as seen in Figure 3.9. This is the main reason to investigate a stereophotogrammetry system using two cameras instead of a multi-camera one. However, there is something important that has to be reminded from the theory of stereophotogrammetry: only the points seen by both cameras can be placed in space. This means that it is not feasible to reconstruct the whole sample, but it is not something completely necessary taking into account that the samples are almost always symmetric. The high brightness of the plasma torch and electromagnetic interference are some of the factors that can complicate the experiment. Moreover, as the main objective of the project is to follow the recession of the sample's surface, special attention should be paid to the possibility of tracking surface features properly.

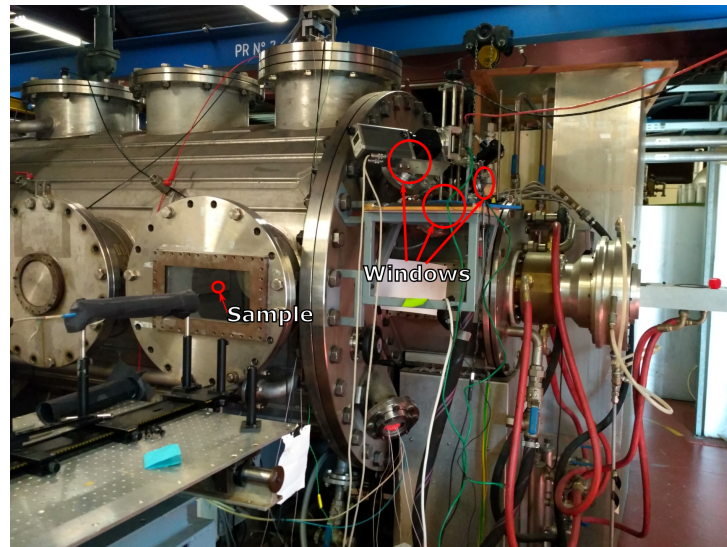


Figure 3.7: Optical accesses in VKI Plasmatron (side view).

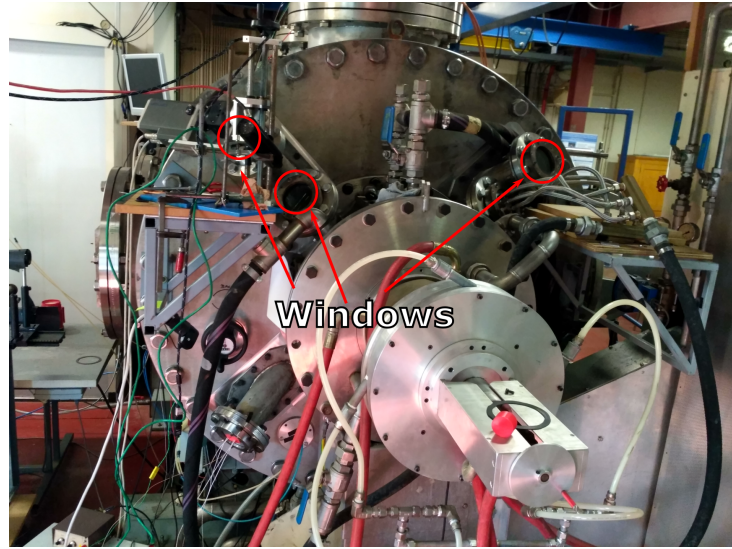


Figure 3.8: Optical accesses in VKI Plasmatron (back view).

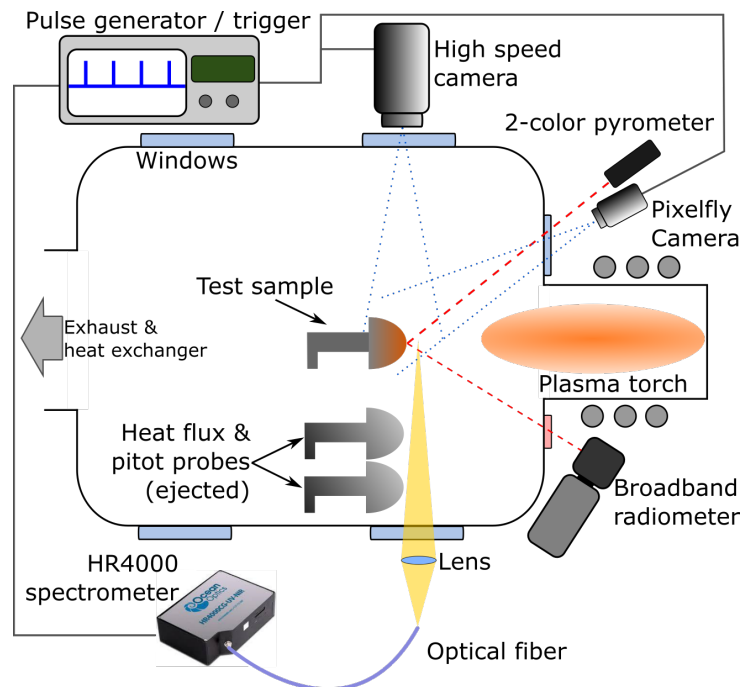


Figure 3.9: Schematic of experimental setup in VKI Plasmatron [13].

4 Software development: *Stereo 3D Reconstruction*

The final objective of this project is to develop and validate a stereophotogrammetry software to create 3D models of real world objects: *Stereo 3D Reconstruction*. As it has been explained previously, the application that motivates this project is to reconstruct TPM samples in the VKI Plasmatron in order to study their surface recession rates. This chapter details the programming language and libraries employed to implement the software, as well as description of the software itself and its applications.

4.1 Software development tools

Software development is the process of designing, coding, testing, documenting and maintaining a certain software program. The motivation of the project provides the conception of the program, but the rest of the process remains uncompleted. First, this section is devoted to the description of the tools used to implement the software. The design of the program will be explained in section 4.2 and the testing and validation in chapter 5.

4.1.1 Programming language: Python

A programming language is a formal language that provides a set of instructions to a computer in order to produce a certain output. These languages are used to create programs and code algorithms in the software environment.

The programming language chosen for this project is **Python**. This language was released in 1991 by Guido van Rossum, but it has been in the last years when its use has grown more rapidly until becoming one of the leaders of the programming language community. Python is a language with a strong abstraction from the details of the computer (high-level programming language), which comes from the philosophy of creating a clear code with high readability for general-purpose programming. Moreover, Python supports object-oriented programming, has a wide standard library and features a dynamic type system and automatic memory management. Nowadays, Python is one of the most demanded languages in the job market and it is being used in world-wide companies like Google and Youtube and NASA [31], [32], [33].

The reasons to choose Python as the programming language of this project are simple: it is an open source language with a wide range of applications that allows intuitive and clear coding. Being a simple and readable language, in addition to the current high demand for Python and the large amount of free documentation, makes learning relatively simple considering the

duration of the project.

4.1.2 Stereophotogrammetry libraries: OpenCV

Another reason to choose Python to develop the code is that it allows the implementation of a large amount of libraries, being one of them the base of the stereophotogrammetry algorithms used by the program: **OpenCV**.

OpenCV (Open Source Computer Vision) is a library of programming functions that implement Computer Vision and Machine Learning algorithms. It was originally developed by Intel and later supported by Willow Garage. It is implemented on multiple computing platforms and is free under the open-source BSD license for both academic and commercial use. OpenCV supports a wide variety of programming languages like C++, Python, Java etc and is available on different platforms including Windows, Linux, OS X, Android, iOS, etc.

OpenCV-Python is the Python API (application programming interface) of OpenCV. It combines the best qualities of OpenCV, originally coded in C++, with the simplicity of Python language. Python is slower than other languages like C++, but it can be easily extended with C/C++. This property has allowed the developers to write computationally intensive codes in C/C++ and create a Python wrapper for it so that these wrappers can be used as Python modules. This has two main advantages: first, the code is as fast as original C/C++ code (since it is the actual C++ code working in background) and second, it is very easy to code in Python. This is how OpenCV-Python works, it is a Python wrapper around original C++ implementation [34].

4.2 *Stereo 3D Reconstruction* description

Stereo 3D Reconstruction is an in-house software developed at VKI and based on the implementation of OpenCV functions in Python. In this section an overview of the code will be done, describing its characteristics, initialization and performance. Once the operation of the program is understood, it is time to detail how it works in deep, describing the internal behavior in order to show its structure and potential.

4.2.1 Software overview

The programming paradigm used in the code has been OOP (Object Oriented Programming), which is based in the concept of an "object" that contains data (attributes) and code (methods). The program has the capability to run certain tasks in parallel by using bash scripting and parsing files. Moreover, making a user friendly software was one of the goals of the project, leading to the development of a program that does not require the knowledge of the internal codes to be used.

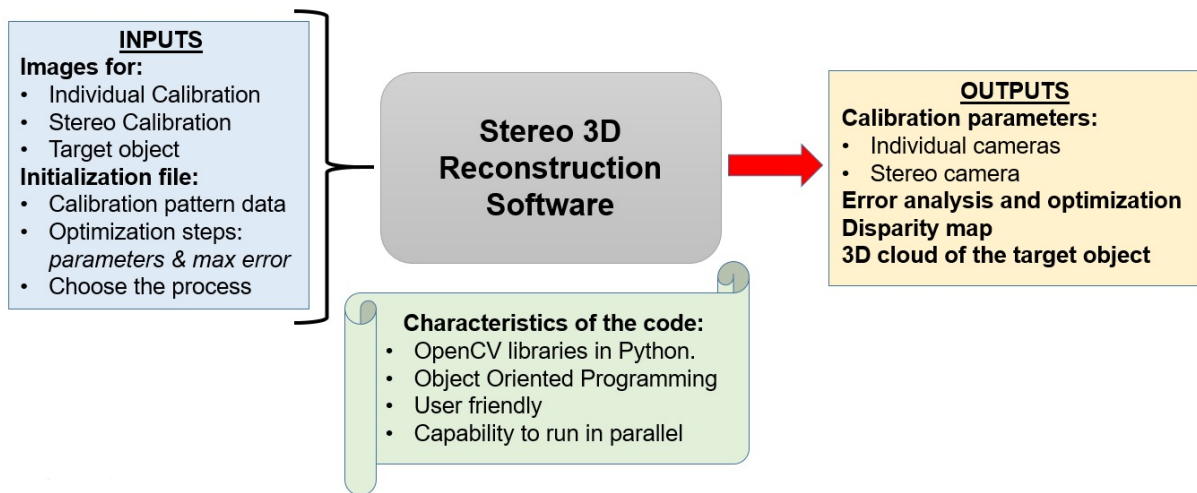


Figure 4.1: Schematic of *Stereo 3D Reconstruction* software: characteristics, initialization and results.

Figure 4.1 shows a scheme of the principal features of *Stereo 3D Reconstruction* software. From the already described characteristics it has been remarked that making a user friendly program is really important, this leads to the creation of an initialization file and folder and to obtain a folder with results. This methodology allows the user to avoid the contact with the code. Furthermore, it is also important to describe the nature of the initialization and the results:

- Initialization: the inputs of this project are a folder with images and an initialization file that has to be filled. The way in which the folders are created, named and distributed is described in the initialization file depending on the task to be performed. The initialization file contains the data about the size and shape of the calibration pattern, the optimization steps and the application that will be performed. The optimization steps have three main inputs: the number of steps to be done, the intrinsic parameter to be optimized and the maximum error allowed for each parameter in each step.
- Results: the outputs of this project depend on the application that has been chosen by the user. The calibration parameters of the individual cameras and the stereo camera, as well as the error analysis and optimization of these parameters are the outputs of the calibration applications. From two pictures of a target object taken from different positions the program provides the disparity map as a result. Finally, with the disparity map and knowing the relative position of one camera with the other, the program creates a three-dimensional point cloud of the target object in a .PLY file (this files can be loaded in programs like MeshLab).

4.2.2 *Stereo 3D Reconstruction* applications

One of the most interesting features of this program is the possibility of choosing between several tasks depending on the needs of the user. All of these applications are of course related

with photogrammetry and allow users to perform a whole project step by step and validating each procedure. The applications of the software will be detailed in deep in this subsection:

4.2.2.1 Camera calibration

This application of the software was the first one to be developed and allows us to calibrate one camera. This means that the program will provide data about the focal length (f_x , f_y), the coordinates of the principal point (c_x , c_y) and a mathematical model for image distortion. This data can be considered valid until the configuration of the camera changes (i.e changing the lens may result in a different distortion and a new value for the focal length).

The first step is to initialize the application. To do this it is necessary to include in the folder of the project another folder called "images" with the pictures of the calibration pattern and also to complete the initialization file. The images folder must contain more than two images of a 2D calibration pattern (chessboard) in different positions. A detailed description of a correct methodology to take the pictures used in the calibration will be described in Chapter 5. In the initialization file it is necessary to indicate the process (individual calibration), the images format (.tif, .jpg, .png, etc) and the physical data of the calibration pattern (square size, mesh width and mesh height).

It has to be commented that the camera matrix coefficients (f_x , f_y , c_x , c_y) are always calculated by the program, but the distortion coefficients may change depending on the mathematical model. The reason is that the camera lens distortion may be very different between models (i.e a GoPro with fish-eye lens does not have the same distortion as a normal DSLR camera). A model with three coefficients for radial distortion (k_1 , k_2 , k_3) and two coefficients for tangential distortion (p_1 , p_2) is set as default (See Subsection 2.3.2). However, the data of the coefficients that have been selected is parsed from the initialization file and included as the attribute of an object inside of the code, which makes straightforward to change the coefficients and perform a calibration with another mathematical model for image distortion. Most professional cameras have a negligible tangential distortion, so OpenCV functions allow to choose between modeling it or not. Moreover, in these functions it is possible to choose the mathematical model for the radial distortion by selecting the coefficients that will be modeled and the ones that will be considered zero. For example, the default model sets k_4 , k_5 and k_6 from Equations 4.1 and 4.2 as zero and calculates k_1 , k_2 and k_3 . Another common model for lenses with low radial distortion is using just k_1 and k_2 . An interesting sensitivity analysis to perform in order to obtain a complete analysis of camera distortion modeling would be using the maximum combination of models, check the results, compare them and choose the most suitable for each camera.

$$x_{\text{corrected}} = x \cdot \left(\frac{1 + k_1 r^2 + k_2 r^4 + k_3 r^6}{1 + k_4 r^2 + k_5 r^4 + k_6 r^6} \right) \quad (4.1)$$

$$y_{\text{corrected}} = y \cdot \left(\frac{1 + k_1 r^2 + k_2 r^4 + k_3 r^6}{1 + k_4 r^2 + k_5 r^4 + k_6 r^6} \right) \quad (4.2)$$

Once this point has been reached, it is time to press "Run" and wait the result. However, it is also important to describe **what is happening inside of the code**:

- **Parsing data from initialization file:** all the data described above is read from the initialization file and included as attributes of an object (this is why OOP is the programming paradigm used in this project).
- **Set object points:** The chessboard of the calibration images has some points (the inner corners) that have certain coordinates. The key of using a 2D calibration pattern is that the Z coordinate of every point will be always 0, and the X and Y coordinates can be obtained knowing the width and length of the mesh and the size of the squares. Therefore, if a 20 x 10 chessboard with a square size of 20mm is used the object points will be passed as (0,0,0),(0,20,0),(0,40,0),...(20,0,0),(20,20,0),..., (400,200,0).
- **Obtain image points:** OpenCV has a function that allows to find the inner corners of a chessboard: *cv2.findChessboardCorners*. By doing this it is possible to obtain the position in pixel coordinates of the corners of the chessboard in each picture.
- **Calibration:** OpenCV has a function that implements the Zhang's camera calibration algorithm [27] previously commented in Subsection 2.3.4.1: *cv2.calibrateCamera*. This function uses as input the size of the image, the image points (x_i, y_i) of each picture and the object points (X_o, Y_o, Z_o). In order to avoid a common misconception it is useful to clarify that the object points will be the same for all the pictures, as the calibration pattern is always the same. The function returns the camera matrix (f_x, f_y, c_x, c_y), the distortion coefficients (by default k_1, k_2, k_3, p_1, p_2), the rotation and translation vectors (Subsection 2.3.4) and the reprojection error.

$$\begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} = \overbrace{\begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}}^{\text{Intrinsics (Camera matrix)}} \overbrace{\begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix}}^{\text{Extrinsics}} \begin{bmatrix} X_o \\ Y_o \\ Z_o \\ 1 \end{bmatrix} \quad (4.3)$$

The program has provided a result and now it is time to evaluate it. In order to do that there are two main tools to **check the quality of the results**:

- **Visual inspection:** visual inspection is used to check if the distortion modeling has been done correctly. Once the distortion coefficients are calculated, it is possible to correct distortion in any picture taken with the camera by using OpenCV function `cv2.undistort`. Therefore, taking one picture and comparing with the undistorted one is enough to check if the correction is accurate enough. It has to be commented that the correct methodology to see this is by checking real-world straight lines in the image. Moreover, it has been remarked that radial distortion becomes more important in the corners, then it is essential to check this undistortion effect in these areas. A new image of an object with a lot of straight lines covering the whole field of view can be done in order to see this effect, but using the same images of the calibration is also valid and faster.
- **Reprojection error:** the reprojection error is the root mean square (RMS) of the distances, in pixels, between the detected and the reprojected points. Its calculation is done by projecting each 3D chessboard point (object points) into the image plane using the final set of calibration parameters (camera matrix, distortion coefficients and rotation and translation vectors) and comparing with the known position of the chessboard corners (image points). Another way to see it is by looking at Equation 4.3, the reprojection error is the RMS of the differences between the LHS and the RHS for every point in pixels. The RMS error is calculated for each image and the value provided by the function is nothing but the average of each image's error. A reprojection error of 1 pixels means that, on average, each of these projected points is 1.0 pixel away from its actual position. There are many values that can be used as maximum allowable error, but the important part is to understand the concept and adjust a value which fits the possibilities of the project. For example, in high-quality photogrammetry laboratories a the reprojection error should be less than 0.1 pixel. However, it is a common assumption to use 1 pixel as the maximum allowable deviation. It is also important to consider the size of the pixel of the camera, as it can provide an idea of the reprojection error in mm.

Sometimes it happens that one image is incorrect or induces a high reprojection error (see Subsection 5.3.2 *metodologia de como tomar las fotos*), which leads to a polluted result and to lose accuracy. This is frustrating, as the influence of one incorrect image can result in a prohibitive reprojection error coming from an erroneous calculation of the intrinsics. Then, arrived at this point some questions may arise: How do I know that I am not losing accuracy because of some risky images? How do I know if I have taken the correct images and obtained a good result? How do I know that my result is the optimum?

The principal strength of program code lies the possibility of answering this question with numbers. In the initialization there is the option to configure an **optimization** for the results based on the concept of **robust statistics**. This statistical methods have a good performance when analyzing data with a wide range of probability distributions, especially if these distributions are not normal. The strong point of robust statistics is the capability to obtain

results resistant to the influence of the outliers (experimental observations distant from the rest of the observations due to measurement or experimental errors). For instance, the mean is a non-robust measure which has a breakdown point of 0% (one single outlier can completely pollute the result), while the median is a robust measure of central tendency with a breakdown point of 50%. Another example is the comparison of the standard deviation, which is not a robust measure, with the interquartile range (IQR), which is a robust measurement as it avoids the effect of outliers.

In order to apply robust statistics to the project it is important to follow a methodology that allows to obtain consistent results, which means that the values obtained in the camera calibration can be considered correct and independent of the images and process used in the calibration. The methodology followed by [35] is a good example of how to apply robust statistics to optimize the camera calibration results. However, in this project the idea is to create a more user-defined flexible optimization, which is implemented as follows:

- **Creation of subsets of images:** instead of using all the images in the camera calibration, a certain amount of subsets of images is created by using the maximum number of possible combinations depending on the user needs and computational resources. In the initialization file it is necessary to define the number of images per set. In order to clarify this concept it is useful to look carefully at Equation 4.4

$$C_N^m = \frac{N!}{m!(N-m)!} \quad (4.4)$$

Where C is the number of subsets, N is the total number of images and m is the number of images per set. If the number of images per set m is the same as the total number of images N , just one set of images will be created and the optimization will not be possible. Once the number of images per subset starts being less than the total number of images an amount of C subsets of images will be created. Therefore, the number of subsets C will be defined by the user depending on the total number of images N and the number of images settled per set m . However, there are some tips to be considered when determining the number of subsets:

- There should be a minimum of images per subset. Increasing the number of subsets does not always means increasing the accuracy. This is because Zhang’s calibration algorithm (Subsection 2.3.4.1) becomes robust when there is a minimum number of images in each analyzed set.
- The total number of subsets C has to be considered as it will affect the computational time. However, this software is able to run the camera calibration of the subsets in parallel. This feature can also be configured depending on the number of processors available in each computer in order to avoid the program to crash. This does not imply that the number of subsets can be increased indiscriminately. Table 4.1 shows how the number of subsets C grows depending on N and m .

N	m	C
20	19	20
20	18	760
20	17	41040
20	16	2790720

Table 4.1: Example of number of subsets C obtained with 20 images N and certain combinations of images per set m .

In conclusion, there is a minimum number of images that has to be considered prohibite, as having less will induce incorrect results in the subsets. However, once a certain point of images is reached, the result is more likely to be polluted by the effect of incorrect images than by having a low number of images per set. Of course, there is not a fixed rule, as the results are completely case dependent. The minimum number of images per set and the number of sets have to be determined depending on the nature of each project and the computational resources available.

- **Determination of the optimization steps:** as it has been remarked above, the strength of this software is the possibility of implementing the optimization depending on the characteristics of the project. The reason is that the maximum allowable error and the parameters with more deviation will depend on the nature of the project, the methodology, the accuracy requirements and the equipment. Therefore, in the initialization file the user can define as much optimization steps as needed in order to do a serial optimization. In each optimization step it is necessary to set the calibration parameter and the maximum allowable error. Each step can optimize as much parameters as decided, but in this case the optimization will be done in parallel. Here an example of how to initialize an optimization is presented:

```

1  [[ Calibration steps ]]
2
3  number steps = 4
4
5  [[[ step 1 ]]]
6  parameters = re ,
7  max error = 1.0 ,
8
9  [[[ step 2 ]]]
10 parameters = p1 , p2
11 max error = 1.5 , 1.5
12
13 [[[ step 3 ]]]
14 parameters = k1 , k2 , k3
15 max error = 1.5 , 1.5 , 1.5

```

```

16     [[[ step 4 ]]]
17     parameters = fx , fy , cx , cy
18     max error = 0.75 , 0.75 , 0.75 , 0.75
19

```

In this case the optimization will start with a first step to reach an average reprojection error equal or less than one pixel. Later, the subsets will be optimized to obtain a maximum error of 1.5 in the tangential distortion coefficients. Then, the same optimization step will be performed, but with the radial distortion coefficients. Finally, the coefficients of the camera matrix will be also optimized in parallel with a maximum error of 0.75. It is interesting to comment that there is a constraint that can be activated in order to ensure a minimum number of subsets of images. This has been added to avoid deleting a large number of subsets in a very restrictive optimization step, which will ensure not finishing with a prohibitive number of subsets (if the number is too low, statistics methods are not valid anymore).

- **Calculation of the optimization:** now it is the time to explain how robust statistics methods have been implemented in the code. First of all, it is necessary to define the interquartile range (IQR), which is a measure equal to the difference between the 75th (upper quartile) and 25th (lower quartile) percentiles. This measure allows to study the dispersion and variability by dividing a data set into four equal parts thanks to the first (Q1), second (Q2) and third (Q3) quartiles. The quartiles are calculated using the median, being Q2 the median of the whole data set n , Q1 the median of the $n/2$ smallest data and Q3 the median of the $n/2$ largest data. The IQR is used to create boxplots, as in Figure 4.2, which allow to study the probability distribution [36].

The principal reason to use the IQR is the possibility of finding outliers in the data set. The definition of outlier will depend on the definition of the maximum allowable error α made in the initialization. As seen in Figure 4.3, α will define as outlier all the entries that fall above $Q3 + \alpha \cdot \text{IQR}$ or below $Q1 - \alpha \cdot \text{IQR}$. This definition of α will determine the size of the whiskers of the boxplot, leaving the outliers out of them and facilitating its identification.

The optimization algorithm is implemented as showed in Figure 4.4. For the optimization of a certain parameter X with a determined maximum error α , X_N is defined as the vector storing the value of this parameter in each subset (Equation 4.5, being C the total number of subsets). Once X_N is known it is possible to calculate the IQR, Q1 and Q3 of the data set, which allows to define $X_{min}(\alpha) = Q1 - \alpha \cdot \text{IQR}$ and $X_{MAX}(\alpha) = Q3 + \alpha \cdot \text{IQR}$. Then, each value x_i stored in X_N is checked and if all the values are inside of the imposed range, the optimization is completed and the average is assumed as the final value of the parameter. If there are sets outside of the range, they are deleted and the calculation of IQR, Q1, Q3, $X_{min}(\alpha)$ and $X_{MAX}(\alpha)$ is done again, repeating the optimization process

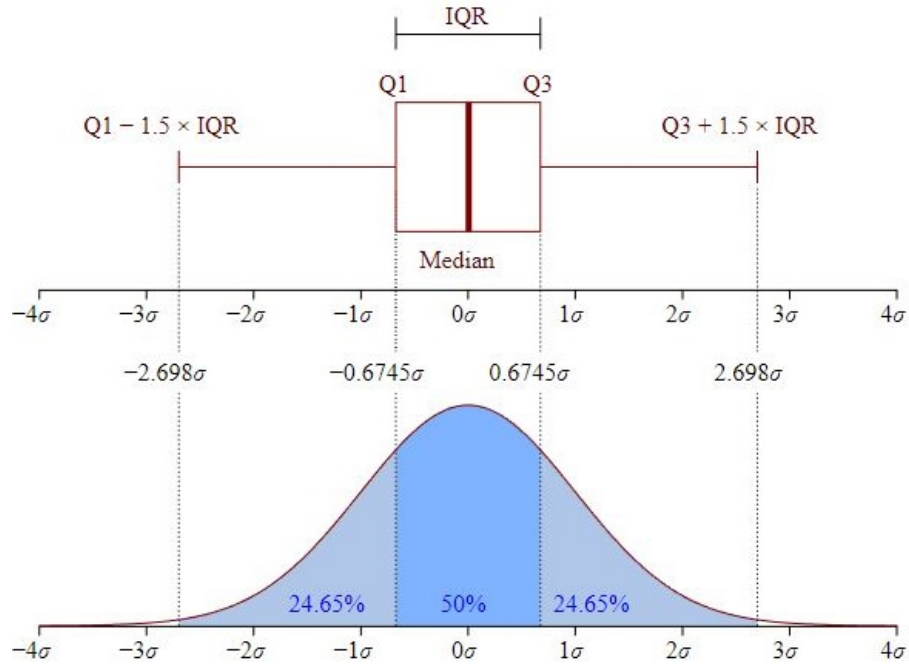


Figure 4.2: Example of boxplot and probability density function of a normal set. Courtesy of Chen-Pan Liao.

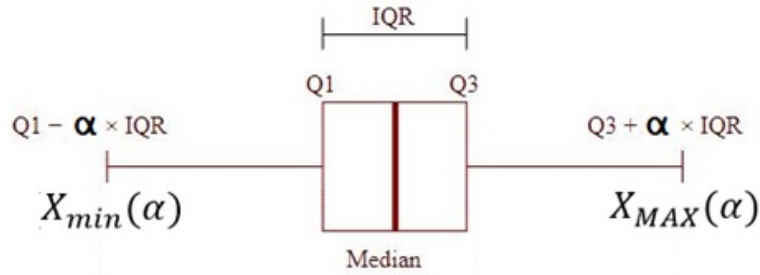


Figure 4.3: Definition of maximum allowable error α in a boxplot.

until having all the subsets inside of the imposed range.

$$X_N = [x_1, x_2, \dots, x_C] \quad (4.5)$$

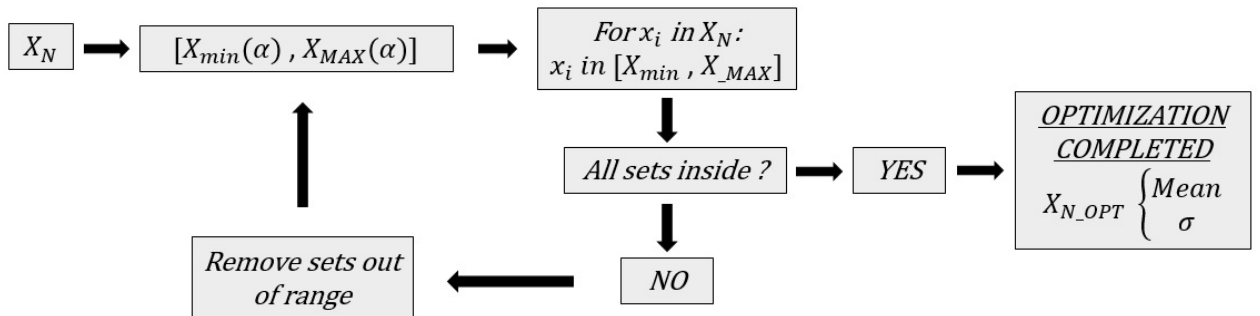


Figure 4.4: Schematic of the optimization algorithm implemented in *Stereo 3D Reconstruction*.

If the imposed range is very restrictive or there is a huge deviation in the optimized parameter, a lot of subsets will be deleted. Therefore, it is more useful to study first the dispersion of the data and then implementing a realistic optimization in several steps. Moreover, if the optimization is done in parallel (several parameters at the same time), all the sets that does not fulfill the conditions for all the parameters will be deleted. So it is better to optimize in parallel parameters with similar variability behavior, like f_x and f_y .

- **Interpretation of the results:** once the program finishes all the processes it creates a folder called results in which an analysis of the optimization can be done. This folder contains text files with the average (μ), standard deviation (σ) and error (σ/μ in %) of each calibration parameter and the average, minimum and maximum of the reprojection error. There is a text file for the initial data and then one for each optimization step, allowing users to study the change produced by the optimization in the values of the calibration parameters. Moreover, there is one image for each parameter showing a histogram, a probability density function and a boxplot of the initial and optimized data set.

4.2.2.2 Stereo camera calibration

This application can be considered as an extension of the previous one to a stereo camera. The reason is that the calibration of a stereo camera requires data about the intrinsic parameters of each one of the two cameras. Then, in this case some images of a calibration pattern are taken for each one of the cameras and then some more images are taken from the same pattern, but in this last case the images are taken simultaneously with both cameras and viewing the pattern at the same position and at the same time (this methodology will be further detailed in Chapter 5).

The initialization of the software in this case is very similar to the one exposed for the calibration of one camera. The option selected must be "Stereo calibration", and the data about the images format and the calibration pattern has to be included too. The main difference is the organization of the folders, as instead of including the images directly in the folder called "images", four folders are included inside of "images"; "images_left", "images_right", "images_stereo_left" and "images_stereo_right". The first two folders contain the images for the individual calibration of the cameras and the second two the images for the calibration of the stereo camera. It has to be remarked that there exists the option of using the same images for the individual calibration of the cameras and the calibration of the stereo camera, but this depends on the nature of the project. Normally, if the baseline of the cameras is large in comparison to the distance of the cameras to the target object, it is better to use different images for calibration and stereo calibration. The reason is that in this cases the common field of view of the stereo camera is limited and small in comparison with the whole field of view of the cameras, leading to individual calibrations in which the calibration pattern does not covers the whole area of the images. On the other hand, if the cameras are really close to each other, it is possible to use the same images because the common and individual field of view is almost the

same.

The stereo calibration is a process which provides information about the relative position of two cameras making a stereo pair. The rotation matrix R and translation vector T , which are results of this application, estimate the location of the second camera with respect to the first one. Moreover, knowing R and T it is possible to calculate the essential matrix E , and knowing also the intrinsics of both cameras the fundamental matrix F can be obtained. This was previously explained and mathematically developed in Subsection 2.3.5.4.

As it has been remarked above, this application is nothing but an extension of the previous one. Therefore, the processes and the optimization commented in the previous point is performed with both cameras, obtaining the optimum camera calibration parameters for each one. Then, at this point is when the new part of the code starts:

- **Stereo calibration images analysis:** It is time to analyze the stereo calibration images and extracting their information. As these images are captured with the cameras in the same configuration, the size of the images will be the same during the whole process. Normally, the same calibration pattern is also used for individual calibration and stereo calibration, and the object points definition remains constant. If this is not the case and there is a new calibration pattern, the object points have to be redefined following the already known methodology. Later, the image points are extracted for the stereo pair images of both cameras.
- **Stereo calibration:** OpenCV has a function that allows us to calibrate a stereo camera: *cv2.stereoCalibrate*. The inputs of this function are the already calculated camera matrices and distortion coefficients of the cameras, the image size, the object points, and the image points of the cameras. It has to be remarked that the image points have to follow a certain order and coincide between pairs. This function returns the rotation matrix R between the first and the second camera coordinate systems, the translation vector T between the coordinate systems of the cameras, the essential matrix E and the fundamental matrix F .
- **Stereo reprojection error:** its calculation is done by projecting each image point of the first camera into the second camera image plane using the camera matrices, the distortion coefficients and the rotation and translation vectors and comparing with the image points of the second camera. Of course, if the individual calibrations are not performed accurately and the reprojection errors are high, the stereo reprojection error will grow rapidly and become prohibitive. The reprojection error of each stereo image pair is computed as the RMS of the errors for each image point. Then, the final reprojection error is calculated as the average of the reprojection errors of each image pair. The result is given in pixels

As in the previous application, an **optimization** algorithm has been implemented in order to ensure that the result of the stereo calibration is accurate and consistent. The optimization methodology is exactly the same: the total set of images is divided in subsets, the optimization steps, parameters and maximum errors are defined and the optimization loop is performed.

It must be clarified, that the rotation matrix has been transformed into a vector by using OpenCV function *cv2.Rodrigues* [34]. The reason is that a rotation vector is a more compact representation and makes the optimization of its components and norm possible. Therefore, the parameters that will be optimized are the stereo reprojection error and the norm and three components of the translation and rotation vectors. An example of the initialization is presented below:

```

1  [[ Calibration steps ]]
2
3  number steps = 3
4
5  [[[ step 1 ]]]
6  parameters = stereo_re, # (RE)
7  max error = 3, # max mean reprojection error
8
9  [[[ step 2 ]]]
10 parameters = Tnorm,
11 max error = 0.75,
12
13 [[[ step 3 ]]]
14 parameters = RVnorm,
15 max error = 0.75,
16
17 [[[ step 4 ]]]
18 parameters = Tx, Ty, Tz
19 max error = 1, 1, 1
20
21 [[[ step 5 ]]]
22 parameters = RVx, RVy, RVz
23 max error = 1, 1, 1

```

This application also generates a folder with results in which the values of the parameters, its progression and error analysis can be studied. Moreover, the images showing the histogram, PDF, and boxplot of the initial and optimized parameters are also available.

In conclusion, the optimization methodology ensures a result resistant to incorrect images and provide the software with a high grade of independence from the methodology followed to take the pictures. That is really useful in order to calibrate a stereo camera for photogrammetry

without a professional pre-calibrated equipment and brings flexibility to implement this methodology in the project and in any experimental facility .

4.2.2.3 3D reconstruction with calibrated stereo camera

Having an accurate calibration of the stereo camera is crucial to perform a correct stereophotogrammetry analysis. However, there are some more steps to be done in order to obtain the final result. As showed in Figure 4.5, the stereo camera already has to be rectified and the disparity map of the target stereo image pair has to be obtained in order to be able to compute the final 3D reconstruction.

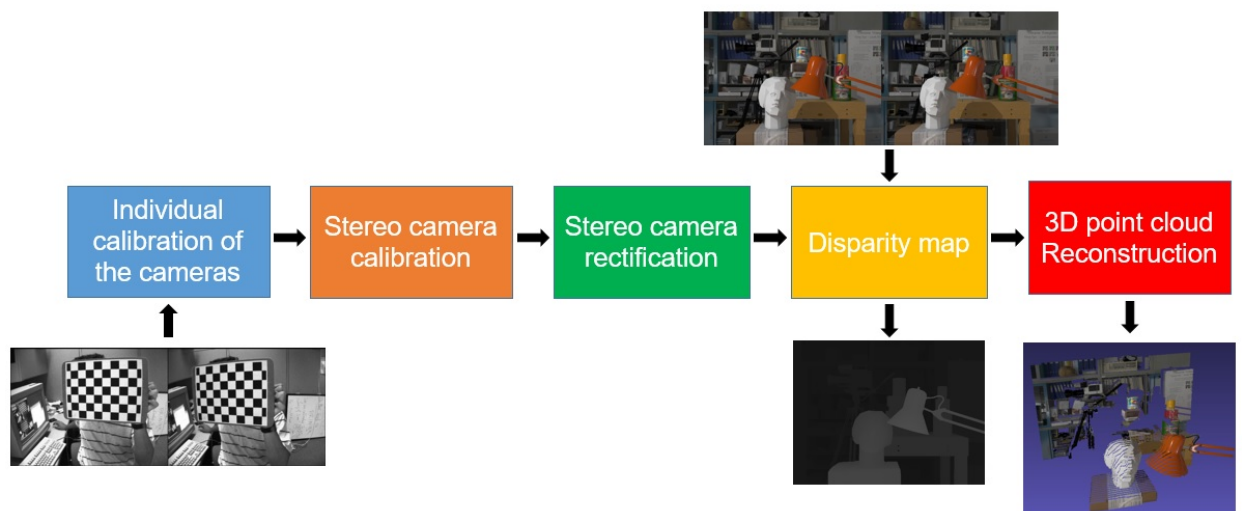


Figure 4.5: Schematic of *Stereo 3D Reconstruction* software application for calibrated stereo camera.

- **Stereo camera rectification:** As it was previously commented in Subsection 2.3.5, once the stereo camera is rectified, the image planes of the cameras virtually rotate to the same plane. The consequence of this process is that the epipolar lines become parallel and the stereo matching problem is significantly simplified, since the correspondences are searched along the epipolar line instead of the whole image.

OpenCV has a function that allows users to rectify a stereo camera: `cv2.stereoRectify`. The inputs of the function are the camera matrices and distortion coefficients of both cameras, the image size, the translation vector T between the coordinate systems of the cameras and the rotation matrix R between the coordinate systems of the first and the second cameras. The function returns the rotation matrix of the rectification transform of both cameras $R1$ and $R2$, the projection matrix in the new rectified coordinate system of both cameras $P1$ and $P2$ and the disparity-to-depth mapping matrix Q [34].

- **Disparity map:** The next step is to find the disparity map of the rectified image pair. As it was explained in Subsection 2.3.5 and showed in Figure 2.13, by finding its disparity it is possible to determine the depth of a certain point. In order to do this, OpenCV function `cv2.StereoSGBM_create().compute` returns the disparity map of a stereo image pair. Of course, in order to obtain a disparity value a point has to be found in both images. This is the reason why the points that does not lie in the common field of view of both cameras cannot be reconstructed.

In addition, it may be the case that finding common features in a pair of stereo images becomes a very difficult task for the program. In order to understand this, it is useful to define first the concept of feature from the computer vision point of view: "A feature in image processing is detected as a point or region around which there is a very large gradient in terms of pixel information". The features are normally specific structures like edges, points or objects. In order to detect these features, the computer analyzes the data of the feature pixel (RGB values) and the data of the surrounding pixels. An example of an image with a lot of features and one with a low number of features can be observed in Figure 4.6. As the final result depends on the quality of the disparity maps, it can be deduced that objects whose images does not have much features should be avoided as much as possible.



Figure 4.6: Comparison of images with a low and large number of potential features to be detected.

- **3D point cloud reconstruction:** As showed in Equation 4.6, from disparity and the Q matrix it is possible to obtain the 3D coordinates of any point seen by both cameras. This Q is the result of the rectification process in the software.

$$Q \begin{bmatrix} x \\ y \\ d \\ 1 \end{bmatrix} = \begin{bmatrix} X \\ Y \\ Z \\ W \end{bmatrix}, \quad Q = \begin{bmatrix} 1 & 0 & 0 & -c_x \\ 0 & 1 & 0 & -c_y \\ 0 & 0 & 0 & f \\ 0 & 0 & -1/T_x & (c_x - c'_x)/T_x \end{bmatrix} \quad (4.6)$$

It is necessary to comment the physical meaning of the components of this Q matrix in order to completely understand the process. T_x is the distance of the baseline in the x-axis

direction. The reason of using this component of the translation is that OpenCV assumes that the cameras are only translated in this direction. Figure 2.16 shows clearly this assumption. Of course, if the cameras are moved in any other direction it would also be possible to do a 3D reconstruction, but using other assumptions.

Continuing with the assumptions of OpenCV functions, the focal length of the cameras f is calculated as the average of the focal length values for each one of them. In addition, the value of the focal length of the cameras is assumed as the average of f_x and f_y (only valid in the case of perfectly square pixels). Moreover, the coordinates of the principal point c_x and c_y are also the average of the values obtained for the two cameras.

These assumptions can be also verified in Figure 2.16, in which the camera intrinsics are assumed to be equal for both cameras. Then, it can be deduced that using two equal cameras with the same configuration is mandatory in order to obtain a coherent Q matrix and obtain a 3D reconstruction. In conclusion, OpenCV allows users to do a 3D reconstruction, but with some constraints that cannot be violated. In the future development of the code, the implementation of an algorithm to perform a 3D reconstruction with a calibrated stereo camera formed by different cameras is proposed.

4.2.2.4 3D reconstruction with uncalibrated stereo camera

Having a calibrated system is always beneficial. However, OpenCV has the tools to perform a 3D reconstruction with an uncalibrated stereo camera. The process to do this is illustrated by Figure 4.7 and will be described in deep.

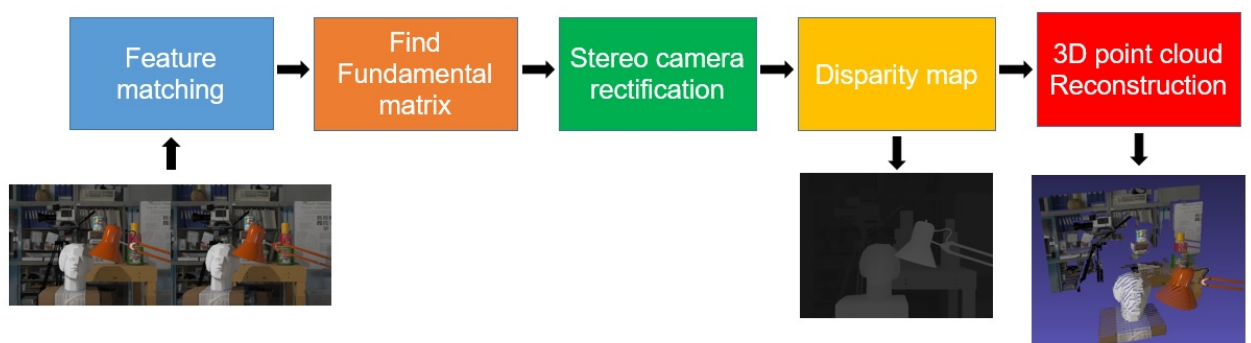


Figure 4.7: Schematic of *Stereo 3D Reconstruction* software application for uncalibrated stereo camera.

- **Features matching:** in this application a highly structured image is used instead of a calibration pattern. This image should have as much features as possible and they should be different one from another in order to avoid confusion during matching. Just one pair of stereo images is necessary. To analyze these images, OpenCV has a function to find the

feature points of the images: *sift.detectAndCompute*. This function implements SIFT algorithm (Scale-Invariant Feature Transform) ([34] page 165). This algorithm allows us to extract the features of both images, but they still have to be matched. Then, OpenCV function *cv2.FlannBasedMatcher* returns the matches between features, which are the image points of both images.

- **Fundamental matrix calculation:** once the features points have been matched and stored as image points, OpenCV function *cv2.findFundamentalMat* returns the fundamental matrix F by using RANSAC (Random sample consensus) algorithm. This is an iterative robust method to obtain the parameters of a mathematical model, especially useful in the cases in which the data sets contain outliers.
- **Stereo camera rectification:** with the image points and F matrix it is possible to initialize *cv2.stereoRectifyUncalibrated* function, which implements the algorithm [37]. The result is the rectification transformation, without knowing the intrinsic parameters of the cameras and its relative positions. In this case the output are the homography matrices $H1$ and $H2$, which represent the planar perspective transformation of the images.
- **Disparity map:** This process is exactly the same than in the previous application.
- **3D point cloud reconstruction:** In this application, the disparity-to-depth mapping matrix Q is not obtained from the rectification process; it is passed manually. The principal point is assumed to be the center of the image, so c_x and c_y will be width/2 and height/2 respectively. The focal length f is calculated knowing the focal length of the lens and the crop factor of the camera. Finally, T_x is assumed to be 1, so 3D reconstruction obtained will be in a certain scale depending on the real value of T_x . Therefore, it is compulsory to reconstruct first an object with a known shape in order to obtain the scale factor and the real T_x value before reconstructing the target object.

It is obvious that this application performs the 3D reconstruction faster, but not having a calibrated equipment and using approximated values for the intrinsics of the cameras induce a certain error in the final result. Moreover, it can be noticed that image distortion has not been computed and if it is large, the error will grow.

On the other hand, this methodology has several advantages. For example, it is useful to compare the results obtained in the rectification with the calibrated and the uncalibrated system in order to validate the results. Moreover, the disparity map values depend on the options selected in the initialization of the function. Therefore, as the values of disparity will determine depth of the points, it is also mandatory to use a 3D calibration object to calibrate the inputs that will be used in the function and ensure obtaining depth in a correct way. In conclusion, the

last two applications have been implemented in order to use both of them and increase the redundancy of the process to ensure correct results.

5 Experiments and results

This chapter is devoted to explain the experiments carried along this project and its results. First of all, a depth-of-field calculator has been implemented in order to ensure the obtainment of high-quality pictures. Then, an experiment with the tutorial pictures from OpenCV has been carried in order to understand the algorithms and applications of *Stereo 3D Reconstruction*. Finally, an experiment using VKI available equipment has been done in order to check the behavior of the code and the feasibility of its implementation in the VKI experimental facilities.

5.1 Camera Depth of Field calculator

The theory about Depth of Field (DoF) has been previously explained in Subsection 2.2.2.5. However, some more research can be done in the mathematics of this concept. Hyperfocal distance and far and near distances of acceptable sharpness can be calculated following the equations from [38].

Hyperfocal distance:

$$H = \frac{f^2}{Nc} + f \quad (5.1)$$

Far and near distances of acceptable sharpness:

$$D_f = \frac{s(H - f)}{H - s} \quad (5.2)$$

$$D_n = \frac{s(H - f)}{H + s - 2f} \quad (5.3)$$

Where s is the focus distance, N is the f-number, c is the circle of confusion in mm and f is the focal length in mm.

This DoF equations have been implemented in MATLAB and Excel in order to study the sensitivity of the total depth of field with respect to the camera parameters. As it is known that in Plasmatron the focused object is placed at 1m, this configuration is assumed by default ($s = 1000\text{mm}$). Then, aperture and focal length are modified in order to show how this affects depth of field.

This calculator is based in a DSLR camera Nikon D5000 with CMOS sensor and 12.3 MP. The camera mounts a AF-S DX NIKKOR 18-105mm f/3.5-5.6G ED VR lens manufactured by Nikon. This is one of the cameras used by the Aeronautics and Aerospace Department of VKI and therefore it has been chosen as a reference. Moreover, the results obtained in with this

calculator are really useful in order to configure the camera for every application.

First of all, diffraction equations (Subsection 2.2.2.3) have also been implemented in order to allow the detection of this effect and avoid it if some parameter of the configuration is modified (Table 5.1). It has to be remembered that the Airy disk must be smaller than the CoC and 3 times the pixel size in order to consider that there is no diffraction. It can be observed that using an aperture number $N = 8$ does not induce diffraction.

INPUTS		RESULTS	
Circle of confusion (mm)	0.022	Airy circle (mm)	0.011
Light wavelength (mm)	0.00055	3 x pixel size (mm)	0.017
Aperture number	8	CoC (mm)	0.022
Pixel size (mm)	0.0055		

Table 5.1: Example with the diffraction calculator.

Later, the DoF equations have been implemented with some default values that have to be settled as inputs, as it can be seen in Table 5.2. Then, the Excel file returns the data about the DoF and the hyperfocal distance, as it can be observed in Table 5.3.

INPUTS		CoC calculator	
Focus distance (mm)	1000	Sensor width (mm)	22.5
Focal length (mm)	50	Sensor height (mm)	169
Crop factor	1.54	Diagonal (mm)	28.14
Real focal length (mm)	76.88	CoC (mm)	0.022
Aperture number	16		
CoC (mm)	0.022		

Table 5.2: Inputs of the DoF calculator.

RESULTS	
Focus distance (mm)	1000
DoF	Near limit (mm)
	Far limit (mm)
	Total (mm)
	In front of subject (mm)
	%
	Behind subject
	%
Hyperfocal distance (mm)	16887.17

Table 5.3: Results of the DoF calculator.

In order to study in deep the behavior of the configuration parameters of the camera, the DoF equations have been implemented in MATLAB obtaining some enlightening results. In Figure 5.1 it can be observed that depth of field decreases dramatically with focal length for a

fixed N value of 11. The sample can be assumed to have a length of 10 cm or less, so an effective focal length of $f = 70$ ($f = 45$ and crop factor of 1.54) can be assumed to be the maximum allowable value. Moreover, in Figure 5.2 it can be observed that for a fixed $f = 76.5$ ($f = 50$ and a crop factor of 1.54) aperture should be at least $N = 16$ in order to have 10cm of depth of field. It can be also be commented that Depth of Field grows linearly with aperture.

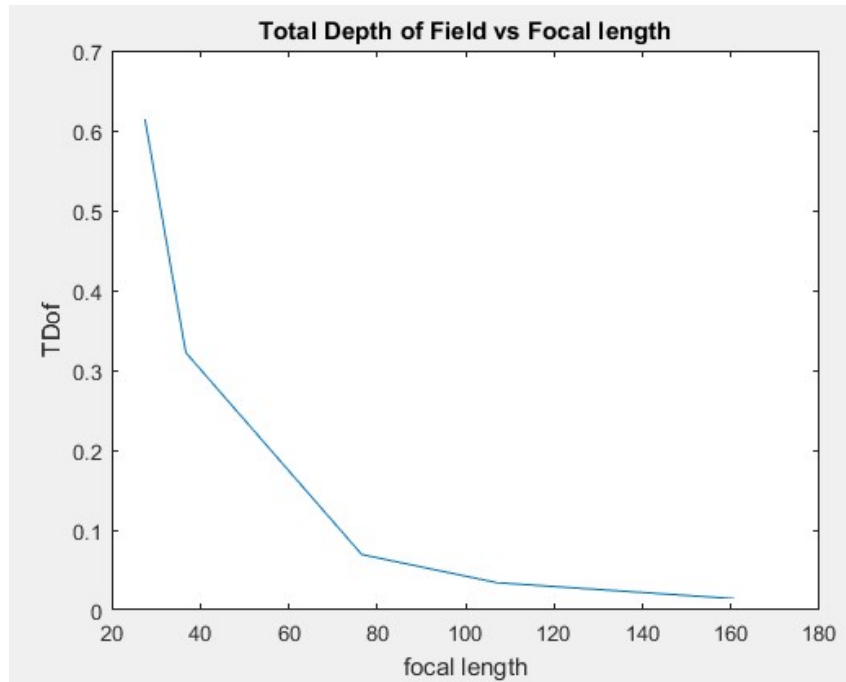


Figure 5.1: Total DoF in meters with respect to f in mm.

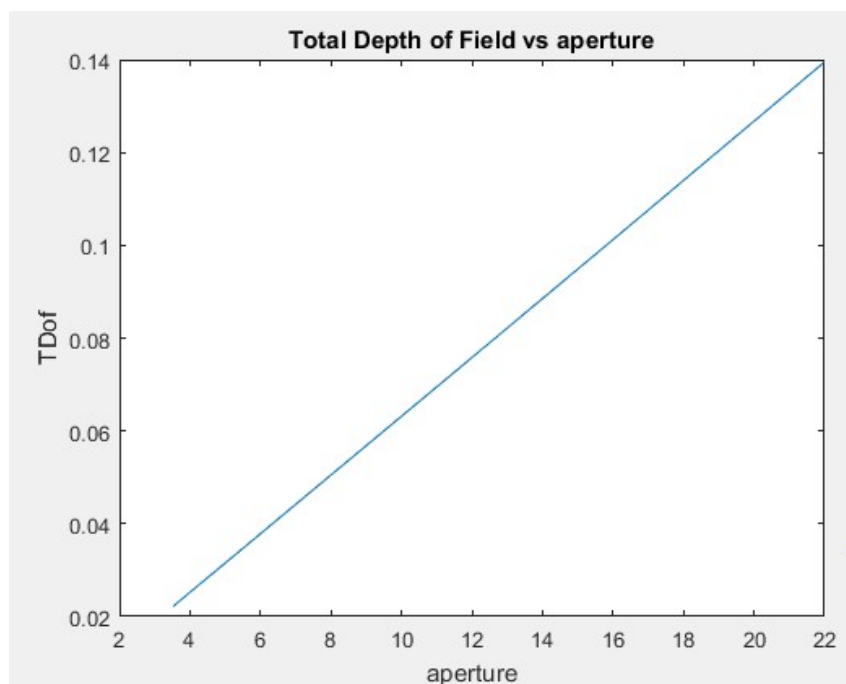


Figure 5.2: Total DoF in meters with respect to aperture number N .

Finally, in order to provide a more global image of how focal length and aperture affect the total Depth of Field, the contour plot of Figure 5.3 is presented. The above explained tendencies can be observed. Moreover, a point representing a normal configuration in VKI Plasmatron is marked with a red cross. The experiments made in this facility have such a huge amount of light that the aperture number of the camera is the largest and the shutter times are very low. This means that the DoF of the camera will be large enough to capture the probe with a perfect sharp image, and that the transient effects of the plasma in the probe will be captured thanks to the high frequency of image capture.

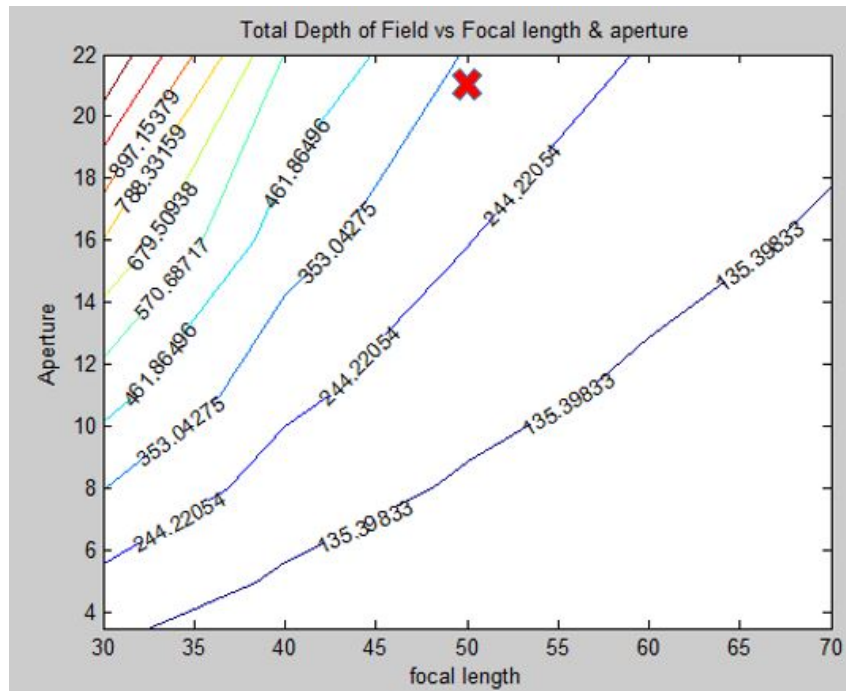


Figure 5.3: Total DoF in meters with respect to aperture number N and focal length in mm. The red cross of the diagram represents a possible configuration in VKI Plasmatron.

5.2 Stereo 3D Reconstruction validation with OpenCV images

OpenCV provides a set of images in its tutorial in order to check that the functions work correctly. However, there is not a set of images that allows to make a complete 3D reconstruction. Therefore, the applications of the program will be tested step by step, showing the results and the features of the code.

5.2.1 Individual calibration

An example of the image set provided by OpenCV can be observed in Figure 5.4. There is a total of 13 stereo pairs of images for the left and right camera. The equipment used is completely unknown, but it can be seen that they are basic webcams.

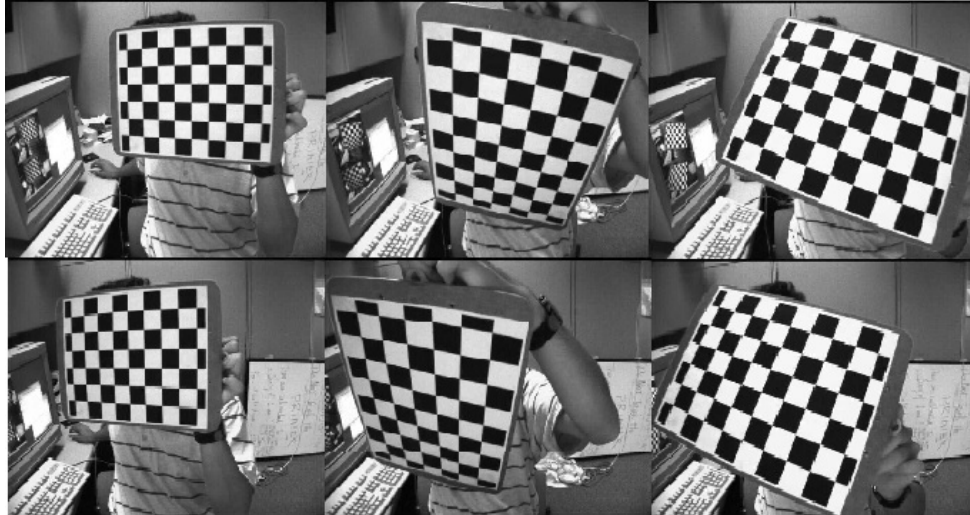


Figure 5.4: Example of three stereo image pairs provided by OpenCV [tutorials].

The aim of this point is to show the behavior of the software when calibrating a camera. The most remarkable characteristics of the code will be commented in order to show how the results are affected. Among them, the parallel computation, the field of view control and the parameters optimization by robust statistics will be the ones explained.

As explained in Section 4.2, a folder called "images" is created with the 13 calibration images. The initialization file is completed as showed in Figure 5.5.

As it can be observed, a number of subsets of images $C = 286$ will be created and analyzed for a total number of images $N = 13$ and $m = 10$ images per subset (Equation 4.4). The optimization steps show that four steps have been chosen: the first one defines a maximum mean reprojection error of 0.5, the second and third ones establish a maximum error of $\alpha = 1.5$ for the calibration coefficients, and the last step imposes $\alpha = 1$ for the camera matrix parameters. Finally, the results obtained from the calibration process show that the program has a great behavior and confirm its robustness, accuracy and reliability.

5.2.1.1 Parallell processing performance boost

Figure 5.6 describes the save in computational time that parallel processing of the images subsets means. It is also useful to observe that there is a point where increasing the number of processors working in parallel does not improves computational time. The reason is that the maximum number of processors of the computer is 16, which is the maximum number of parallel computations allowable. Someone may think that saving around 100 seconds does not suppose a big change. However, in the corner of the image there is a table which shows the total number of pixels of a DSLR camera and the number of analyzed subsets of a normal project (one with around 20 calibration images). It can be observed that there is a huge difference in the total number of pixels that should be analyzed, so having parallel processing can avoid prohibitive computational times in cases in which a high-quality camera is used to take a

```

# IN ORDER TO INITIALIZE THE CALIBRATION FOLLOW THESE STEPS:
# - Save the calibration images in a folder:
#   - 'images' if the calibration is performed just for one
camera.
#   - 'images/calibrate_left' & 'images/calibrate_right' for a
stereo calibration.
# - Complete this file to configure the optimization.

[ Calibration ]
# Set 1 the kind of calibration desired

Individual calibration = 1
image format = 'jpg'

Stereo calibration = 0
image format Left= 'jpg'
image format Right= 'jpg'

square size = 10 # size of pattern square in mm
mesh width = 9
mesh height = 6

images per set = 10

camera matrix parameters = fx, fy, cx, cy
distortion coefficients parameters = k1, k2, k3, p1, p2

[ Optimization ]

# Specify the calibration steps and constrains.
# Elements to optimize:
# - RE (Reprojection error).
# - Camera matrix elements (FX, FY, CX, CY).
# - Distortion coefficients (K1, K2, P1, P2, K3,...).
# Max error for the parameters
# - X for the RE in pixels
# - Y for the rest of parameters in max_deviation/average
# Constrains to set:
# - Minimum set of images.
# - Images in all quadrants.

[[ Calibration steps ]]

    number steps = 4

    [[[ step 1 ]]]
        parameters = re,
        max error = 0.5,

    [[[ step 2 ]]]
        parameters = p1, p2
        max error = 1.5, 1.5

    [[[ step 3 ]]]
        parameters = k1, k2, k3
        max error = 1.5, 1.5, 1.5

    [[[ step 4 ]]]
        parameters = fx, fy, cx, cy
        max error = 1.0, 1.0, 1.0, 1.0

```

Figure 5.5: Example of initialization file for individual calibration.

considerable amount of calibration images.

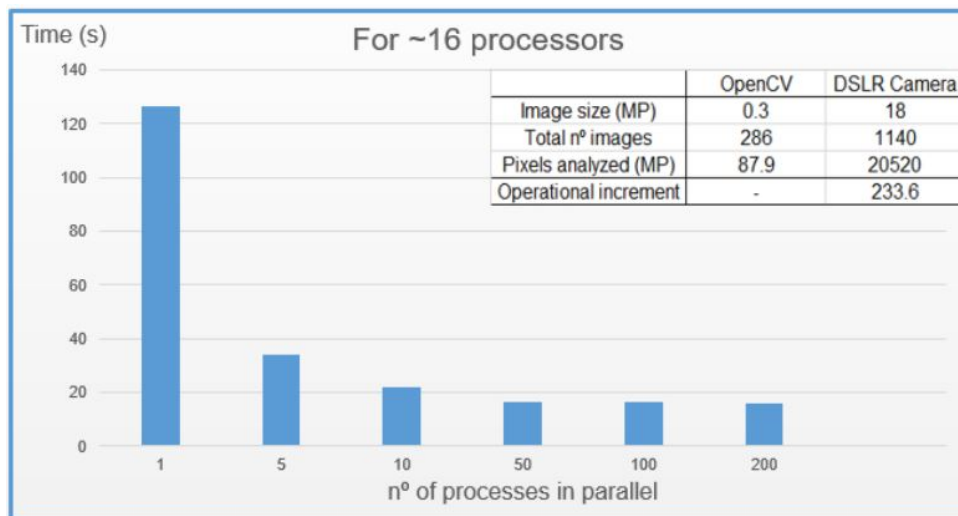


Figure 5.6: Example of initialization file for individual calibration.

It must be taken into account that it may happen that the simulation crashes if the maximum number of processors is exceeded and there is a huge amount of data to be analyzed by each one. Increasing the image size, having a large amount of images per subset and having poor images are the principal causes of slow computations. The reason is that the larger the number of pixels in which the computer has to search the chessboard corners, the larger the time needed to find them. Moreover, having more images means not only increasing the time by finding the image points of each one but also increasing the computational time needed to solve Zhang’s calibration algorithms. Finally, it may be a surprise that the factor that increases the most the computational time is the existence of poor images. The reason is that the program wastes a huge amount of time analyzing them and trying to place the image points, while in most of the cases it does not manage to find all of them and the image is discarded.

5.2.1.2 Distortion modeling control tool

Stereo 3D Reconstruction code contains a tool specially designed to control the distortion modeling. The behavior of this tool is pretty simple and lies in the definition of a good modeling methodology. OpenCV functions create a distortion model by finding the line of best fit for the image points with least square methods. Therefore, in order to obtain a correct mathematical model, it is mandatory to have a considerable amount of points covering the whole field of view. As it is explained in Subsection 5.3.2, the number of image points can be increased by taking more pictures or by increasing the number of points in the chessboard. However, having a large amount of points is not enough, as it is known that if the points are not uniformly distributed the model will be poor in the areas with less point. It is not a coincidence that the number of points is usually much less in the image corners, where the radial distortion effect is the largest. Therefore, it is very important to control the number of points used to model distortion and checking that the corners of the image have a fair density of points. As it can also be deduced, if

the target object that will be reconstructed is just placed in the center of the image, distortion modeling in the corners become less important. However, it is always interesting to have a distortion correctly modeled.

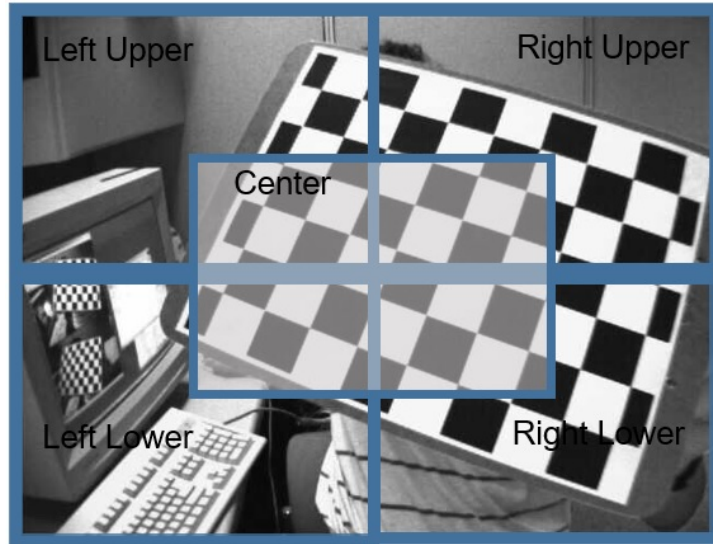


Figure 5.7: Distribution of areas in the image for the distortion modeling control tool.

Image area	% of points
Upper Left	22.3
Lower left	19.3
Upper Right	28.7
Lower Right	29.7
Center	60.6

Table 5.4: Image points distribution along the image area.

Figure 5.7 provides an example of the point distribution for one of the images used in the calibration. This figure allows to understand the regions considered to determine point density. As this program is devoted to be as much user-friendly as possible, changing the size and shape of the areas in the code is straightforward. As it can be observed in Table 5.4, the distribution is quite uniform along the quadrants, being the lower left the one with less points and the lower right the one with more. It can be also seen that 60.6% of the points are concentrated in the center region.

Moreover, OpenCV functions allows to check the final result using visual inspection. Figure 5.8 shows an image before and after undistortion using the mathematical model obtained in the calibration. As it can be observed, the straight lines of the real world are now straight lines in the image, showing the value of a good distortion model in photogrammetry.

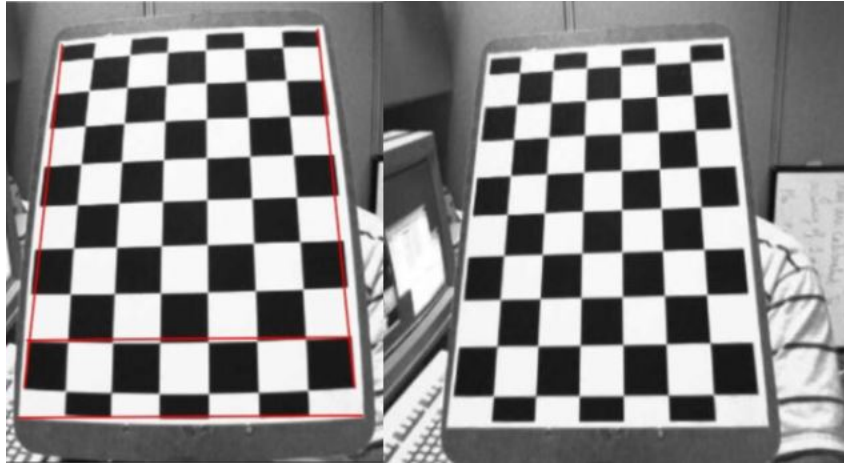


Figure 5.8: Undistortion effect in an image. Courtesy of OpenCV.

5.2.1.3 Calibration parameters optimization

As it was explained in the description of the code, a "Results" folder is created at the end of the simulation. This folder contains the error analysis (mean μ , standard deviation σ and error σ/μ in %) for each parameter of the camera matrix and for the distortion coefficients for each step of the calibration. Moreover, there is an image showing a histogram, a PDF and a boxplot of the initial and final optimized values. Of course, it means that there is a prohibitive amount of images to be showed in this report. However, the principal tendencies observed in the results will be commented and the more enlightening images will be showed. An example of the evolution of f_x is showed below and in Figure 5.9 in order to clarify the nature of the results obtained for each one of the nine calibration parameters.

```

1  [CALIBRATION PARAMETERS] = [[ fx ]]
2
3  STEP 0 (before optimization)
4  Average = 468.6753699029242
5  Standard deviation = 8.636026097196162
6  Error = 1.8426456032850467
7
8  STEP 1
9  Average = 468.6753699029242
10 Standard deviation = 8.636026097196162
11 Error = 1.8426456032850467
12
13 STEP 2
14 Average = 469.78568837528144
15 Standard deviation = 8.373602880733975
16 Error = 1.7824303906944148
17
18 STEP 3

```



```

19 Average = 468.77256879414597
20 Standard deviation = 7.043744340394227
21 Error = 1.5025931142927809
22
23 STEP 4
24 Average = 470.1188132546165
25 Standard deviation = 5.570165673848108
26 Error = 1.1848421115687844

```

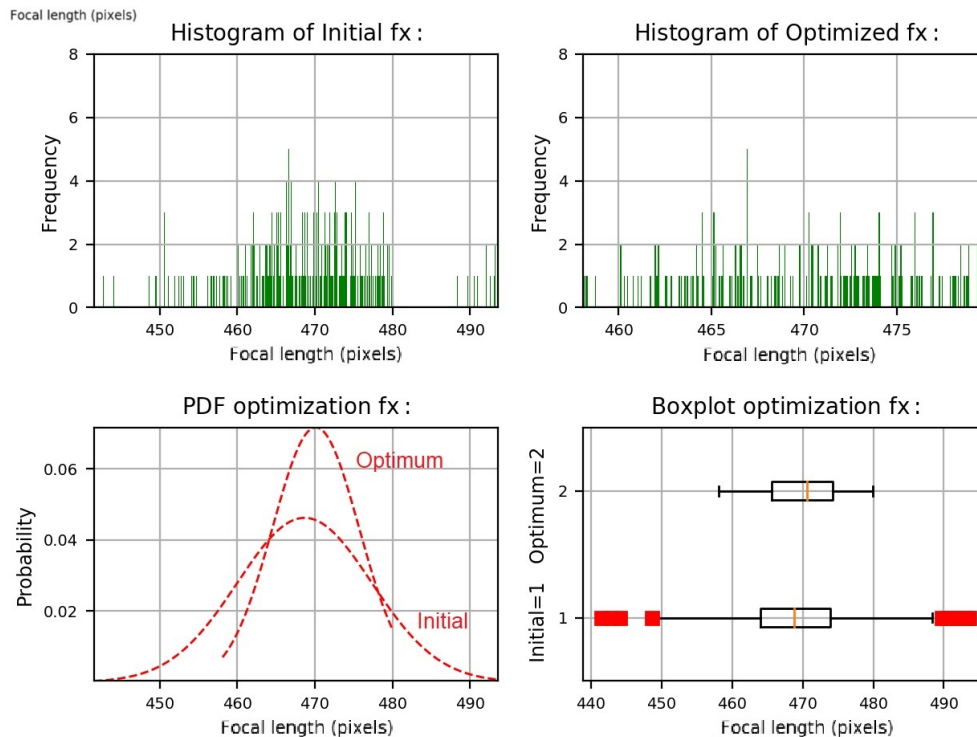


Figure 5.9: Optimization of f_x showed in Results folder: histogram, pdf and boxplot of the initial and optimum subsets.

Regarding the optimization results, these are the main conclusions obtained:

- **Camera matrix parameters:** As it can be observed in Figure 5.10 and Table 5.5, the focal length parameters had a larger error before calibration and some outliers can be identified. The same conclusions can be extracted from the principal point coordinates results seen in Figure 5.10 and Table 5.6. It can be also remarked that f_x and f_y are almost equal, which is reasonable taking into account that the camera is supposed to have squared pixels.
- **Distortion coefficients:** The behavior of the radial distortion coefficients along the optimization is exposed in Figure 5.12. On the other hand, the tangential distortion coefficients optimization is showed in Figure 5.13. As expected, the optimization finishes

	Initial 286 subsets		Optimum 176 subsets	
Parameter	Mean	Deviation	Mean	Deviation
f_x (pixels)	468.68	8.64	470.12	5.57
f_y (pixels)	468.92	9.06	470.7	5.59

Table 5.5: Focal length parameters before and after optimization.

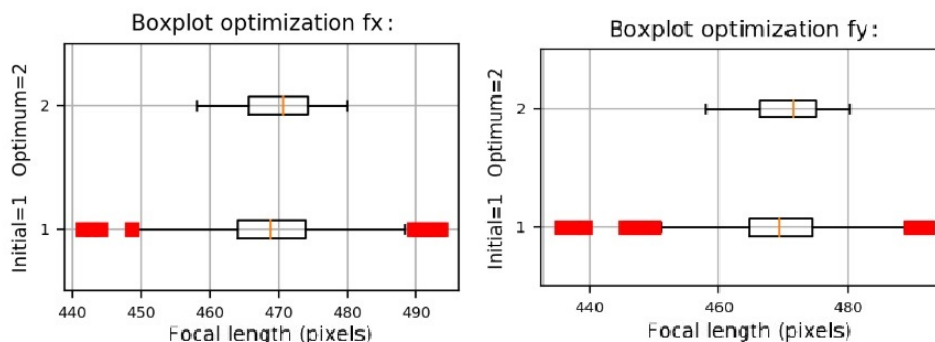


Figure 5.10: Optimization of the focal length parameters.

	Initial 286 subsets		Optimum 176 subsets	
Parameter	Mean	Deviation	Mean	Deviation
c_x (pixels)	342.28	1.26	341.89	0.85
c_y (pixels)	236.79	1.37	237.42	0.48

Table 5.6: Central point coordinates before and after optimization.

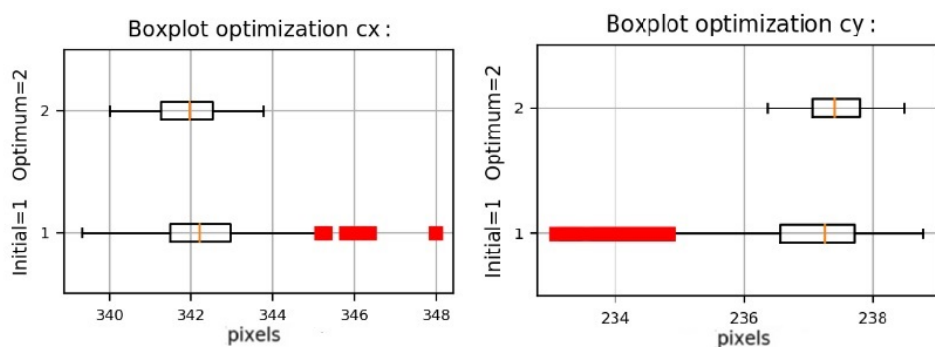


Figure 5.11: Optimization of the principal point coordinates.

reducing the dispersion of the entries.

- **Reprojection error:** The first mean reprojection error obtained was 0.40 and the final one 0.45. As it could be seen previously, the first optimization step does not change the distribution nor deletes any image subset. The reason is that the reprojection error is always under the maximum 0.5 imposed in the optimization step. This 0.45 pixels final reprojection error means a deviation on the reprojection equivalent to $2.48\mu\text{m}$, which can be accepted as a reasonable value.

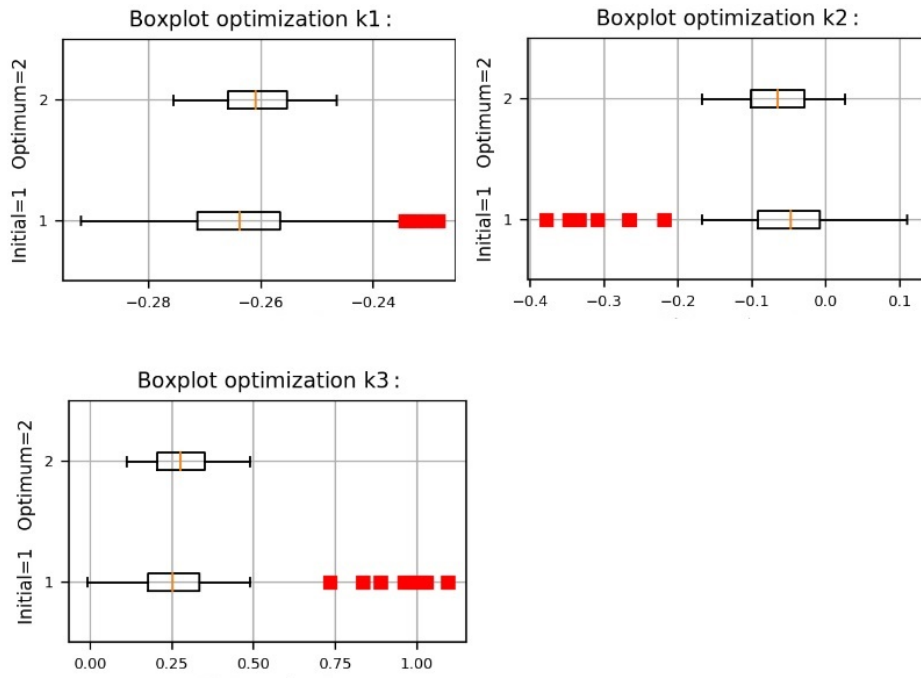


Figure 5.12: Optimization of the radial distortion coefficients.

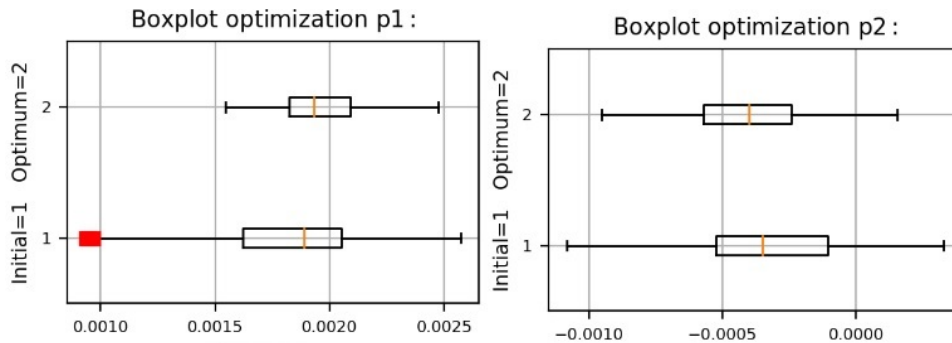


Figure 5.13: Optimization of the tangential distortion coefficients.

5.2.2 Stereo camera calibration

This application is used in order to calibrate the stereo camera formed by two webcams of unknown position. As it was previously explained, this application calibrates first the cameras individually and then performs the stereo calibration. In order to avoid redundancy, just the results of the stereo calibration will be analyzed. First of all, it must be commented that in this case the images used for the individual calibration of the cameras and the ones used for the stereo camera calibration are the same. Moreover, the initialization file is showed below in Figure 5.14 in order to show the optimization steps performed in this case.

In this case, there is the same number of images subsets equal to 286 (same images used for calibration and stereo calibration). After 3 optimization steps in which the first one is dedicated

```

# IN ORDER TO INITIALIZE THE CALIBRATION FOLLOW THESE STEPS:
# - Save the calibration images in a folder:
#   - 'images' if the calibration is performed just for one
camera.
#   - 'images/calibrate_left' & 'images/calibrate_right' for a
stereo calibration.
# - Complete this file to configure the optimization.

[ Calibration ]
# Set 1 the kind of calibration desired

Individual calibration = 1
image format = 'jpg'

Stereo calibration = 0
image format Left= 'jpg'
image format Right= 'jpg'

square size = 10 # size of pattern square in mm
mesh width = 9
mesh height = 6

images per set = 10

camera matrix parameters = fx, fy, cx, cy
distortion coefficients parameters = k1, k2, k3, p1, p2

[ Stereo Optimization ]

# Specify the stereo camera calibration steps and constrains.
# Elements to optimize:
# - Stereo RE (Stereo reprojection error).
# - Translation vector components and norm (Tnorm, Tx, Ty,
Tz).
# - Rotation vector components and norm (RV_norm, RVx, RVy,
RVz).
# Max error for the parameters
# - X for the RE in pixels
# - Y for the rest of parameters in max_deviation/average

[[ Calibration steps ]]

    number steps = 3

    [[[ step 1 ]]]
        parameters = stereo_re, # (RE)
        max error = 3, # max mean reprojection error

    [[[ step 2 ]]]
        parameters = Tnorm,
        max error = 0.75,

    [[[ step 3 ]]]
        parameters = RVnorm,
        max error = 0.75,

```

Figure 5.14: Example of initialization file for stereo camera calibration.

to the stereo reprojection error, the second and third one to the norm of the translation and rotation vectors respectively. The results obtained are explained as follows:

- **Translation vector:** It can be observed that the assumption of considering just a translation in the x-direction is pretty close to be valid, but there is also a non-negligible translation in the z-direction. Regarding the optimization it can be seen that all the components and the norm of the vector decrease their standard deviation, reducing the number of outliers until an acceptable level (Table 5.7). Another characteristic of this case is that having almost all the translation in one direction makes that the boxplot of the norm and the one of this component look almost the same (Figure 5.15). It has to be commented that just the norm of the vector has been optimized because doing it component by component and applying the same restriction may lead to deleting too much subsets. The reason is that the components with less weight in the translation usually have more standard deviation. Therefore, trying to eliminate the outliers of this almost negligible components implies deleting a lot of subsets while the real improvement in the location precision is negligible.

Parameter	Initial 286 subsets		Optimum 244 subsets	
	Mean	Deviation	Mean	Deviation
$ T $ (mm)	32.00	0.18	32.05	0.13
T_x (mm)	-31.26	0.18	-31.31	0.14
T_y (mm)	0.17	0.15	0.16	0.15
T_z (mm)	-6.83	0.12	-6.84	0.11

Table 5.7: Translation vector norm and components before and after calibration.

- **Rotation vector:** The optimization also generates lower standard deviations in the rotation vector norm and components (Table 5.8). Moreover, it is not a surprise to observe in Figure 5.16 the same behavior than in the translation vector due to the domination of the rotation one component. However, the most interesting feature to comment is that the modulus of the vector is $\sim 1.5^\circ$, which shows that the assumption of rectified images and a row-aligned stereo camera is not far away from reality.

Parameter	Initial 286 subsets		Optimum 244 subsets	
	Mean	Deviation	Mean	Deviation
$ Rv $ (rad)	0.0266	0.0017	0.0261	0.0012
Rv_x (rad)	-0.0030	0.0013	-0.0031	0.0013
Rv_y (rad)	-0.0261	0.0017	-0.0256	0.0012
Rv_z (rad)	-0.0039	0.0006	-0.0039	0.0006

Table 5.8: Rotation vector norm and components before and after calibration.

- **Reprojection error:** The initial stereo reprojection error is 1.57 and the optimized value is 1.56. The change is very small due to the fact that the maximum allowable mean

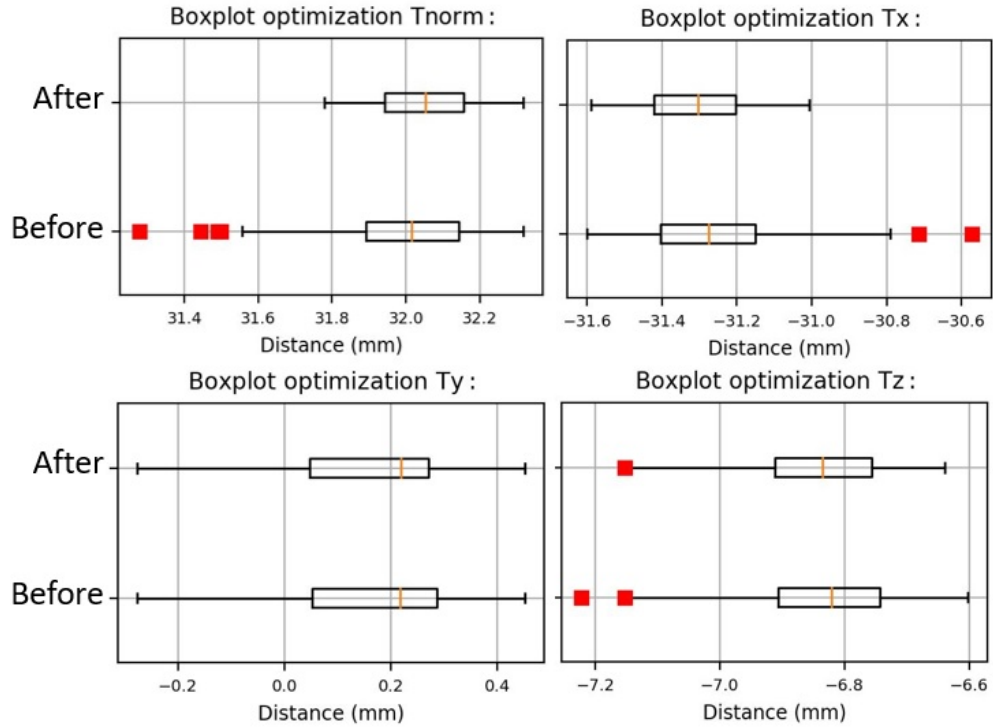


Figure 5.15: Optimization of the translation vector norm and its components showed in Results folder.

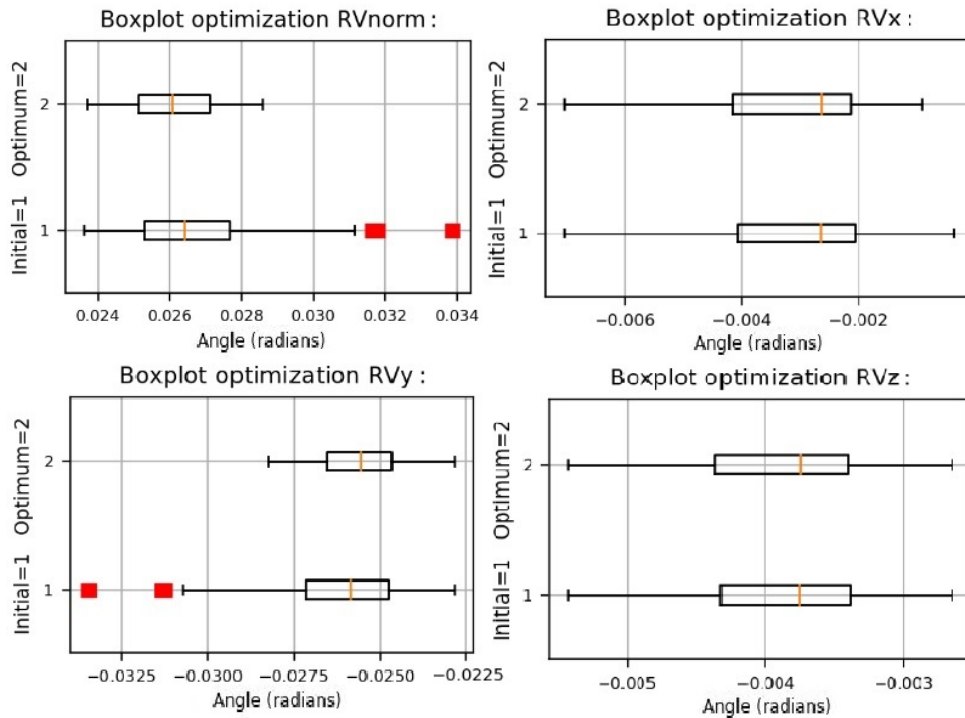


Figure 5.16: Optimization of the translation rotation vector norm and its components showed in Results folder.

imposed in the optimization was 3 pixels. It is known that the final result is above the maximum reprojection error of 1 pixel commented in the theory. However, taking into

account that just 13 images for the calibration were taken and that the equipment is two webcams it can be considered a very good results. In fact, this value of the reprojection error is $8.6\mu\text{m}$.

5.2.3 3D reconstruction

As it was previously said, OpenCV does not have images of an object taken with the same camera than the calibration images. Therefore, performing the whole 3D reconstruction process is not possible in this case. However, the images of an aloe vera plant can be used in order to check the behavior of the functions that reconstruct a 3D object with a calibrated camera. This stereo pair of images is showed in Figure 5.17.



Figure 5.17: Stereo pair of images of an aloe vera plant used in the 3D reconstruction. Courtesy of OpenCV.

As it can be deduced from the theory explained in Subsection 2.3.5.2, a stereo camera can be assumed rectified if the cameras are equal, row aligned and without distortion. Looking at the previous images, it could be assumed that the cameras are the same and that they are just displaced in the x-direction. So the assumption of having an already rectified camera is done and the next steps are explained. Of course, this is not a correct way to proceed and the results will not be as accurate as they could. However, for behavior analysis and function validation purposes it can be valid.

5.2.3.1 Disparity map calculation

The previous set of images is used in order to create the disparity map. The function that calculates disparity has several inputs, in this case the used ones are showed below. Figure 5.18 shows the disparity map returned by the function. For more information about the functions and inputs see [34] and [20]

\StereoSGBMcreate inputs	
minDisparity	16
numDisparities	112-min_disp
blockSize	3
P1	$8*3*window_size**2$
P2	$32*3*window_size**2$
disp12MaxDiff	1
uniquenessRatio	10
speckleWindowSize	100
speckleRange	32

Table 5.9: OpenCV StereoSGBMcreate inputs for disparity calculation.

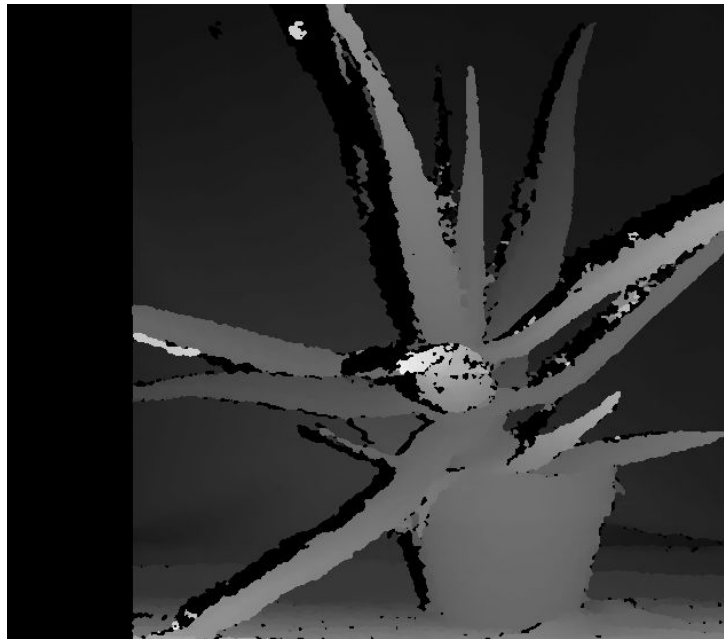


Figure 5.18: Stereo pair of images of an aloe vera plant used in the 3D reconstruction. Courtesy of OpenCV.

5.2.4 3D point cloud calculation

The last step is to calculate the three-dimensional coordinates of the points using the disparity map. As the stereo camera calibration is completely unknown, and arbitrary Q matrix have been used with common values. This values establish the principal point coordinates at the center o the image, assume the baseline equal to 1 and adjust the focal length in order to obtain a reasonable depth in the point cloud. This will return a 3D reconstruction whit a random scale and proportions. The final result is can be seen in Figure 5.19 and 5.19. Regarding the limitations of the procedure it can be seen that a pretty good 3D reconstruction has been obtained. The final result is a .PLY file that is observed in MeshLab.

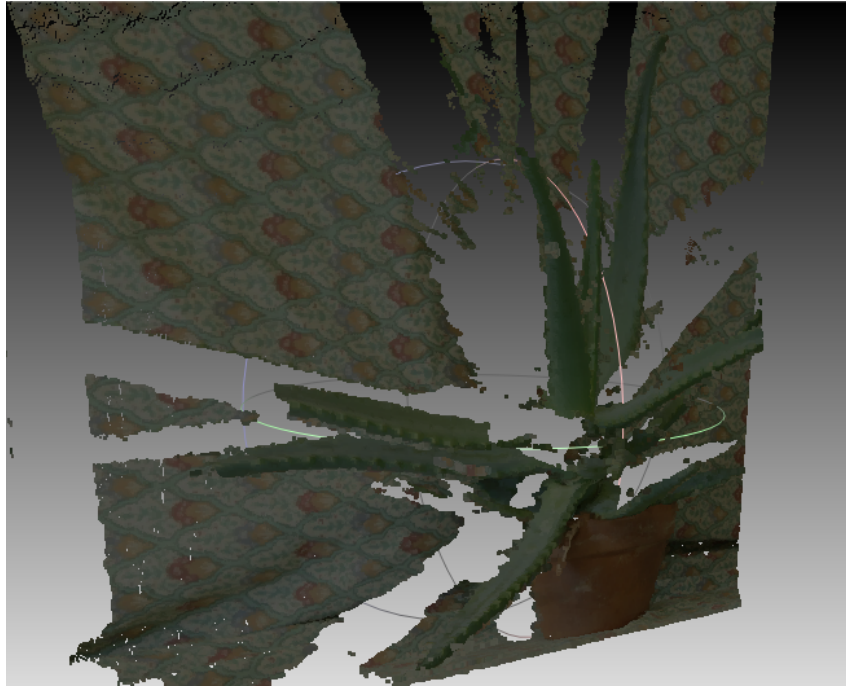


Figure 5.19: 3D Reconstruction of the aloe vera plant visualized in MeshLab.



Figure 5.20: 3D Reconstruction of the aloe vera plant visualized in MeshLab.

5.3 *Stereo 3D Reconstruction* experiment with VKI equipment

This section will explain the last experiment performed in the VKI. The objective of this experiment is to reconstruct a VKI Plasmatron sample and check that the software is capable to analyze its surface. It has to be remarked that due to the high demand of the optical equipment and the difficulty of obtaining them, two equal cameras were accessed during a period of less

than two hours. It will be observed that the results does not show an ideal photogrammetric reconstruction. However, taking into account the reduced amount of time that the equipment was available and the fact that it was completely unknown, it can be observed that the code has a good behavior and that it can perform the functions for which it has been created. Then, this experiment is the milestone that proves that *Stereo 3D Reconstruction* is a flexible and robust tool to perform stereophotogrammetry analysis and understand the nature of the results. Moreover, the results obtained in this section will be employed in the future development of the code.

5.3.1 Experimental setup

The experimental setup of this project is quite simple, which is one of the objectives of implementing a flexible tool to adapt photogrammetry to non professional laboratories. Two equal cameras *SP-12000-CXP4 12MP CMOS global shutter* are used (Figure 5.21). This fairly compact model is well suitable for future VKI Plasmatron experiments as it allows to capture 189 frames per second and has a CMOS sensor, which means that is more robust against electromagnetic noise. The main characteristics of the camera can be seen in Table 5.10. The camera DoD calculator was employed and an aperture number of 8 was used. The lenses focal length is variable and the configuration that made the sample cover perfectly the image was selected. Shutter time was selected in order to obtain the recommended exposure level.



Figure 5.21: SP-12000-CX4 camera model by Stemmer Imaging.

Model	SP-12000-CX4
Resolution (MP)	12
aspect ratio	4:3
pixel size (μm)	5.5 x 5.5
Frame rate	189 frames/s. @ 8-bit

Table 5.10: SP-12000-CX4 remarkable specifications.

The calibration pattern was created and printed according to the sample size. This was a very fast process thanks to the fact that the calibration pattern is just a printed paper placed in a planar surface (a glass in this case). The calibration pattern can be seen in Figure 5.22 and conforms a grid of 21 x 14 inner corners, providing a total of 294 image points per picture. The size of the squares is 3.75 mm.

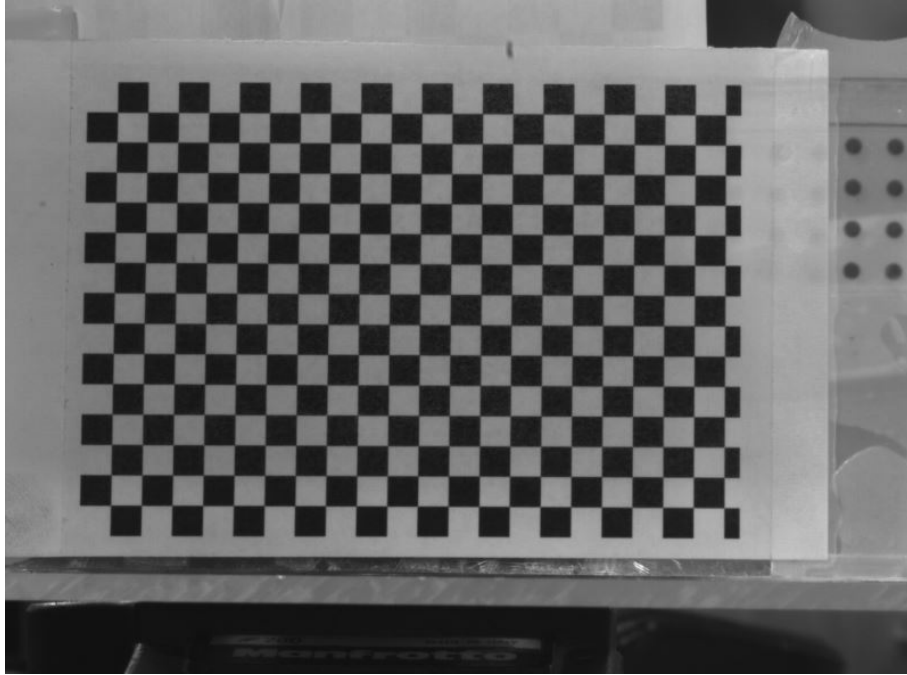


Figure 5.22: Picture of the calibration pattern.

Finally, it is time to present the target object. A cork sample frequently used in Plasmatron experiment was chosen due to the irregularity of its surface. The sample can be seen in Figure 5.23. It is important to say that this sample has a height of 50mm, a diameter of 41.6mm in the base and a diameter of 49mm in the head.

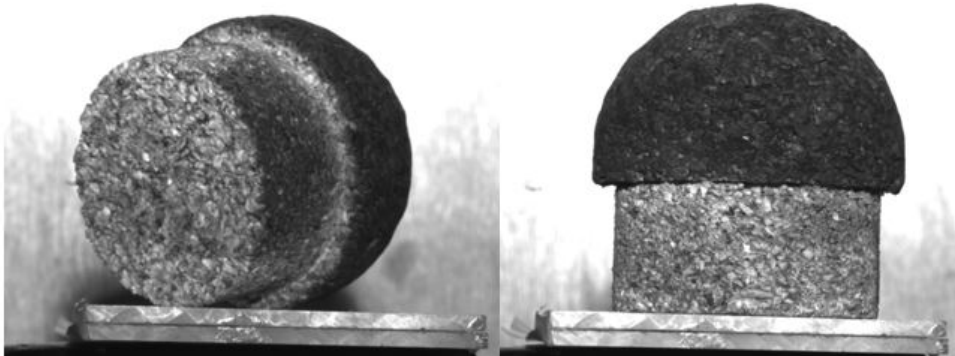


Figure 5.23: Picture of a VKI cork sample.

Due to the lack of time, accurate length measurement tools were not available during the

experiment and the camera and the samples were placed in approximated locations. This is not an issue as this positions are the results provided by the code. However, in order to provide the reader with an idea of the setup arrangement, it is fairly accurate to say that the cameras had an horizontal displacement of $\sim 100\text{mm}$ and the sample was placed at $\sim 600\text{ mm}$ of each one (forming a isosceles triangle). Therefore, it can be deduced that d/B coefficient will be ~ 6 .

5.3.2 Guide for accurate calibration image capturing

Some guidelines to follow in order to capture calibration images accurately will be exposed in this point. Examples are given in order to provide the reader with the guidelines to follow an accurate procedure that will enhance the final results. It has to be remembered that photogrammetry consist on measuring from photographs, so knowing how to obtain the best possible images is mandatory in order to obtain engineering-level results.

5.3.2.1 Camera configuration

The intrinsic parameters of the cameras have to be fixed all along the process, which means that focal length has to be imposed at the beginning. Moreover the camera focus must be set at the beginning and cannot be changed in order to have a fixed focal length. Keep the pattern in focus, but do not use autofocus is a mandatory rule. If zoom settings are changed between images, the focal length change and the calibration process is erroneous.

Moreover, it is recommended to fix shutter time, aperture, and ISO sensitivity depending on the focal length and the physical characteristics of the project. It is known that increasing the aperture number will increase the DoF. Therefore, having a low ISO sensitivity to avoid noise and a big aperture number to ensure the calibration pattern and the probe focus is recommendable. With this configuration the shutter time will need to be considerably large. However, during the VKI Plasmatron experiments luminosity reaches an astonishing level, allowing the reduction of the shutter time and therefore permitting the obtaining of a larger number of images. This large number of images is desirable as it will allow to analyze better the transitory effects of the experiment in the probe's material and surface. In addition, as the stereo camera calibration process does not require a high-frequency acquisition of images, shutter time can acquire large values.

5.3.2.2 Stereo camera position

It is well known that the cameras of the stereo imaging system remain in a fixed position all along the process. Once any of the two cameras is moved during calibration or after it, the whole process has to be repeated as the stereo calibration results will be no longer valid. It is desirable to mount the cameras on a robust plate and to capture the images by means of a

remote systems.

5.3.2.3 Calibration pattern

There is much to say about the calibration pattern. Choosing the adequate for the project characteristics and using it correctly marks the difference between a poor and a quality calibration.

- **Pattern position:** the objective of the calibration is to accurately measure the physical length of a determined object. Therefore, it is obvious that the best result will be obtained calibrating the camera to accurately measure in the same area in which the target object will be placed. During the process of calibration images acquisition it is necessary to place the calibration pattern in the area that will be occupied by the probe. Of course, this pattern can be moved in the close area, but it is important to model the region of interest. Moreover, as explained in 5.3.2.1, the shutter time during camera calibration can be really large. If the pattern is not in a fixed position (i.e it is placed in the hand), the image will be blurred. Therefore there are two options to solve this problem: use artificial light in order to enhance luminosity conditions, or fix the calibration pattern in a certain position during image capturing.
- **Calibration pattern size:** a common chessboard cannot be used when calibrating a camera used to measure a very small piece. The reason is that having the object covering a considerably large region of the field of view will make impossible to acquire images of a big chessboard in the same position. Therefore, the creation of a chessboard pattern with a size suitable for a calibration in order to reconstruct small object is proposed. Figure 5.24 shows one of the first calibration patterns used in this project. As it can be observed, its manufacturing is not a tedious process.
- **Chessboard grid density:** when talking about the grid density it is the number of points per areas what is been described. Then, the denser the grid the more points are calibrated afterwards. The ideal case would be to calibrate each pixel of the camera picture, so a dense grid is preferable. The dimensions of the grid have to be different in X and Y direction in order to avoid confusion of the software. The inner corners are the ones found. The dimension of the squares have to be known as it is an input for the software (object points). Moreover, increasing the density of the chessboard has another advantage. Zhang's algorithm for calibration and distortion mathematical model determination does nothing but finding the line of best fit for the image points with least square methods. Therefore, increasing the grid density will increase the number of image points for each picture. Then, with the same number of images than with a less dense grid, the denser one will allow a better mathematical model. An example of how the grid density increases can

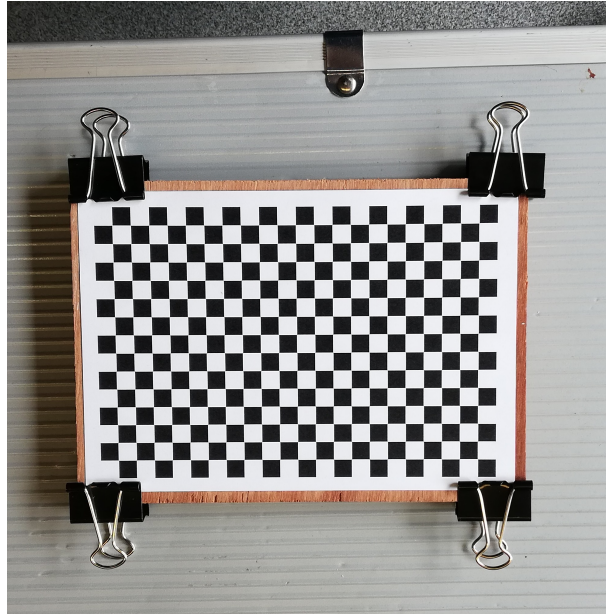


Figure 5.24: Example of one of the calibration patterns used along the experiments of the project.

be seen in Figures 5.25 and 5.25. In this example the number of image points increases from 24 points (6 x 4 grid) to 384 (24 x 16 grid).

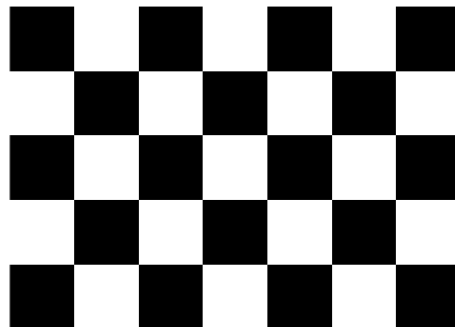


Figure 5.25: Example of corner density for a 6 x 4 grid calibration pattern.

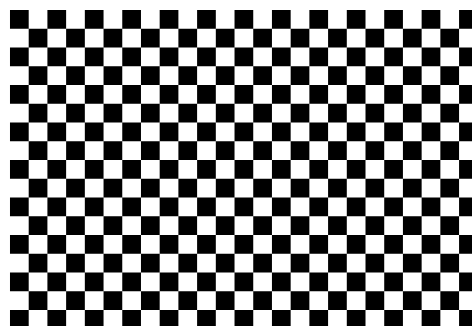


Figure 5.26: Example of corner density for a 24 x 16 grid calibration pattern.

5.3.2.4 Capturing process

There are some main rules to capture a correct set of calibration images:

- Acquire images of the pattern covering the whole field of view. Lens distortion increases radially from the center of the image and sometimes is not uniform across the image frame. In order to model lens distortion the pattern must appear close to the edges of the images. Figure 5.27 shows an example of a set of images for calibration.

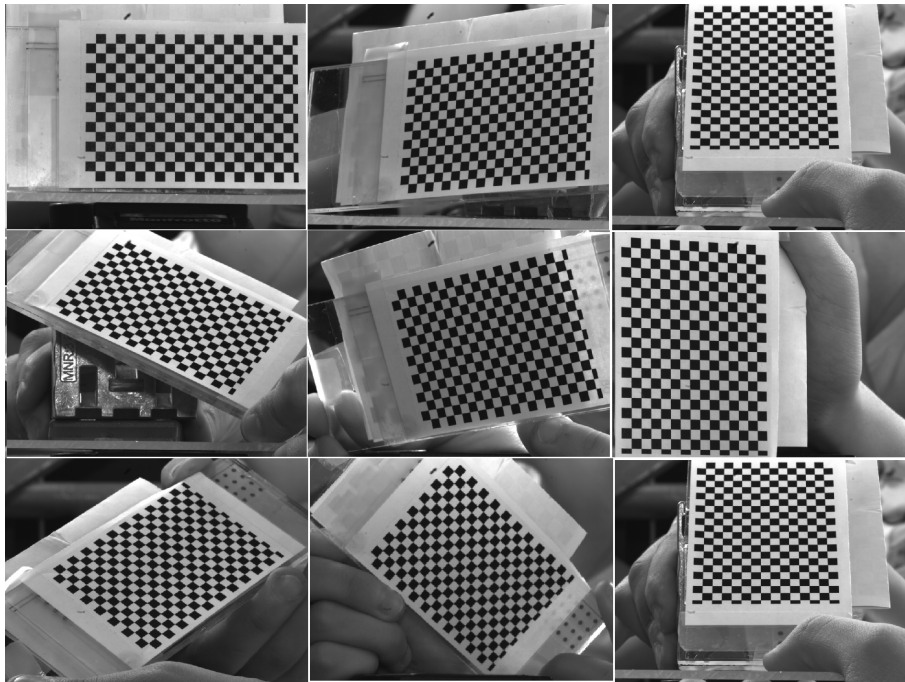


Figure 5.27: Example of one of the calibration patterns used along the experiments of the project.

- A badge of at least 10-20 images has to be obtained for each camera in order to obtain good calibration results with Zhang's algorithm. This number is more an advise than a fixed rule, the result depends more on the nature of the project than on the number of images.
- Take the image at a distance as close as possible to the one of the target object and do not crop or modify them. Remember that these images will be used to measure, and modifying them will induce wrong results.
- Capture the images of the checkerboard pattern at different orientations in the three Euler angles relative to the camera. A maximum angle of 45° from the camera plane can be imposed (Figure 5.28). Remember that Zhang's algorithm requires the translation and rotation of the calibration pattern in order to solve the equations and obtain the same results than a 3D object (2.3.4.1).

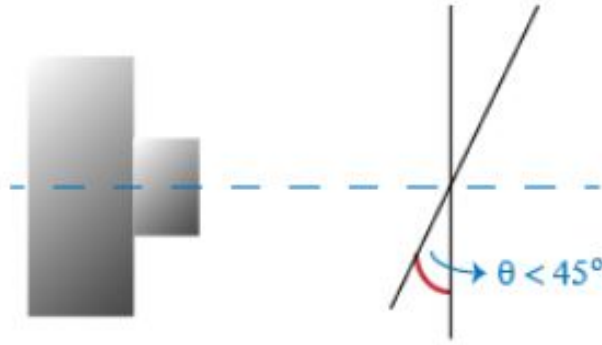


Figure 5.28: Maximum recommended angle of the calibration pattern during image acquisition. Courtesy of MathWorks.

- During the stereo calibration pictures acquisition, the calibration pattern must be visible in both images of the stereo-pair. The images must obtain an image of the pattern in the same position. A movement of the pattern between Left and Right camera would provoke negative effects in the stereo calibration.

5.3.3 Results

Now, it is time to comment the results obtained in this experiment. The nature of the results is exactly the same than in Section 5.2, so the theory and explanations provided will not be repeated to avoid redundancy. The program is tested following its applications steps and the results are analyzed. This methodology will allow the validation of each task and the recognition of the errors, making possible to determine the quality of the results obtained in each step.

5.3.3.1 Individual camera calibration

An example of the set of images used in this section is provided in Figure 5.29. There is a total of 20 calibration images for the left camera, and the same number of images for the right one. These images are placed in their corresponding "images_left" and "images_right" folders. The initialization file is completed as showed in Figure 5.37.

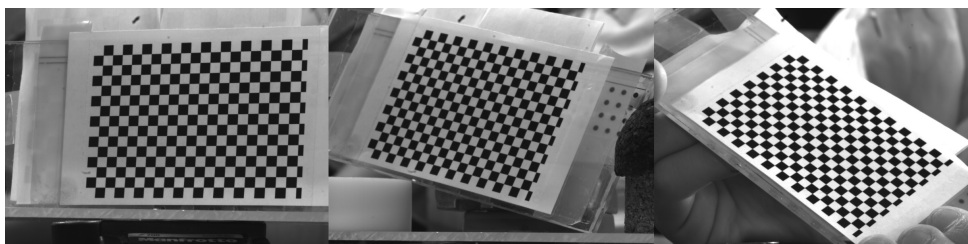


Figure 5.29: Example of images used for left camera calibration.

```

# IN ORDER TO INITIALIZE THE CALIBRATION FOLLOW THESE STEPS:
# - Save the calibration images in a folder:
#   - 'images' if the calibration is performed just for one
camera.
#   - 'images/calibrate_left' & 'images/calibrate_right' for a
stereo calibration.
# - Complete this file to configure the optimization.

[ Calibration ]
# Set 1 the kind of calibration desired

Individual calibration = 0
image format = 'tif'

Stereo calibration = 1
image format Left= 'tif'
image format Right= 'tif'

square size = 3.75 # size of pattern square in mm
mesh width = 21
mesh height = 14

images per set = 18

camera matrix parameters = fx, fy, cx, cy
distortion coefficients parameters = k1, k2, k3, p1, p2

[ Optimization ]

# Specify the calibration steps and constrains.
# Elements to optimize:
# - RE (Reprojection error).
# - Camera matrix elements (FX, FY, CX, CY).
# - Distortion coefficients (K1, K2, P1, P2, K3,...).
# Max error for the parameters
# - X for the RE in pixels
# - Y for the rest of parameters in max_deviation/average
  [[ Calibration steps ]]

      number steps = 4

      [[[ step 1 ]]]
          parameters = re,
          max error = 1.0,

      [[[ step 2 ]]]
          parameters = p1, p2
          max error = 1.5, 1.5

      [[[ step 3 ]]]
          parameters = k1, k2, k3
          max error = 1.5, 1.5, 1.5

      [[[ step 4 ]]]
          parameters = fx, fy, cx, cy
          max error = 0.75, 0.75, 0.75, 0.75

```

Figure 5.30: Example of initialization file for individual calibration.

The same number of images, number of subsets and optimization steps are implemented for both cameras. As it can be observed, a total number of subsets of images $C = 190$ will be analyzed for a total number of images $N = 20$ and $m = 18$ images per subset. It can be seen that four steps have been chosen for the optimization process: the first one defines a maximum mean reprojection error of 1.0 pixel, the second and third ones establish a maximum error of $\alpha = 1.5$ for the distortion coefficients, and the last step settles $\alpha = 0.75$ for the camera matrix parameters.

This time, the parallel processing boost and the distortion modeling tool will not be explained, as it has already been done before. However, Table 5.11 shows the image points distribution along the different areas described previously in order to show that there is a good density of points in each one of them. As is was desirable, the common field of view of both cameras is well modeled (the right region of the left camera and the left region of the right one).

	Left Camera	Right Camera
Image Area	% of points	% of points
Upper Left	24.3	30.2
Lower Left	19.4	32.4
Upper Right	27.6	20.1
Lower Right	28.7	17.3
Center	54.6	62.5

Table 5.11: Initialization file for the calibration optimization.

The following results to be commented will be the ones obtained for the camera parameters with each one of the cameras. The conclusions extracted from the optimization of each camera parameter are the following:

- **Camera matrix parameters:** as the lenses used in the project were variable focal lenses, it was tried to configure them to obtain the same focal length for both cameras. It can be observed both cameras have similar focal lengths, which is a necessary when doing a 3D reconstruction in OpenCV (Table 5.12 and Figures 5.31 and 5.32). Moreover, the optimization process reduces the standard deviation of the focal length entries, as it was expected. The same behavior is observed for the principal coordinates of both cameras, which used to have a considerable amount of outliers that have been deleted. However, if the mean value obtained for the principal point coordinates is analyzed, it can be observed that the result seems quite strange. Both the c_x and c_y coordinate of the cameras look different from one camera to another, in particular c_x from the Left camera has nothing to do with c_x of the Right camera. Moreover, the camera has been configured to capture images at a resolution of 2048 x 1536 pixels, so that the theoretical principal point of the sensor would be at 1024 x 626.5. The coordinates obtained in the calibration are extremely far away from the theoretical ones, so the result for both cameras cannot be considered feasible. This implies that the rest of the calibration and reconstruction process will be

polluted by this error.

	Parameter	Initial set		Optimum set	
		mean	deviation	mean	deviation
Left camera	f_x	10445.05	22.18	10435.01	10.32
	f_y	10458.54	18.41	10450.59	10.56
	c_x	643.76	54.04	670.58	22.43
	c_y	1293.94	26.71	1294.03	8.17
Right camera	f_x	10517.44	24.67	10515.22	13.88
	f_y	10515.79	23.78	10515.12	12.80
	c_x	1630.31	51.67	1658.42	20.90
	c_y	1203.45	33.4	1207.17	13.78

Table 5.12: Camera matrix parameters optimization.

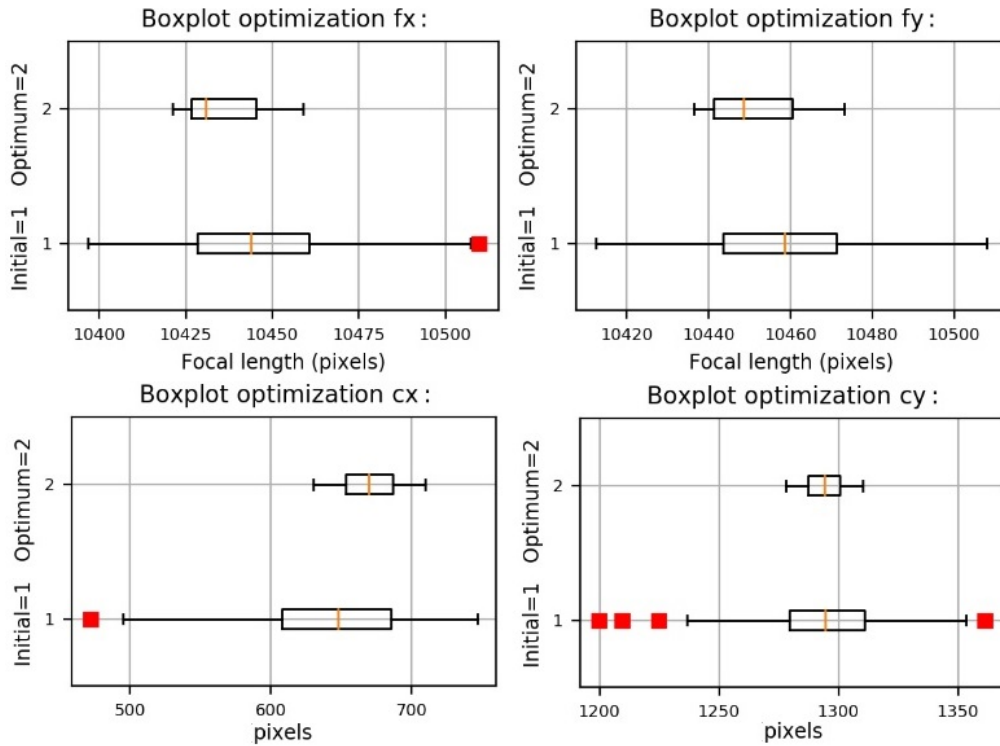


Figure 5.31: Optimization of the camera matrix calibration parameters for the Left camera.

- **Distortion coefficients:** the behavior of the radial distortion coefficients along the optimization of both cameras is exposed in Figures 5.33 and 5.34. The tangential distortion coefficients optimization is showed in Figures 5.35 and 5.36. As expected, the optimization finishes reducing the dispersion of the entries. It can be seen that tangential distortion is almost zero for these cameras. On the other hand, it can be observed that the radial distortion coefficients are different for each camera. It does not mean that the results are wrong, as this is just a mathematical model that adjust distortion. In order to say that a distortion model is wrong it would be necessary to check (visual inspection) that

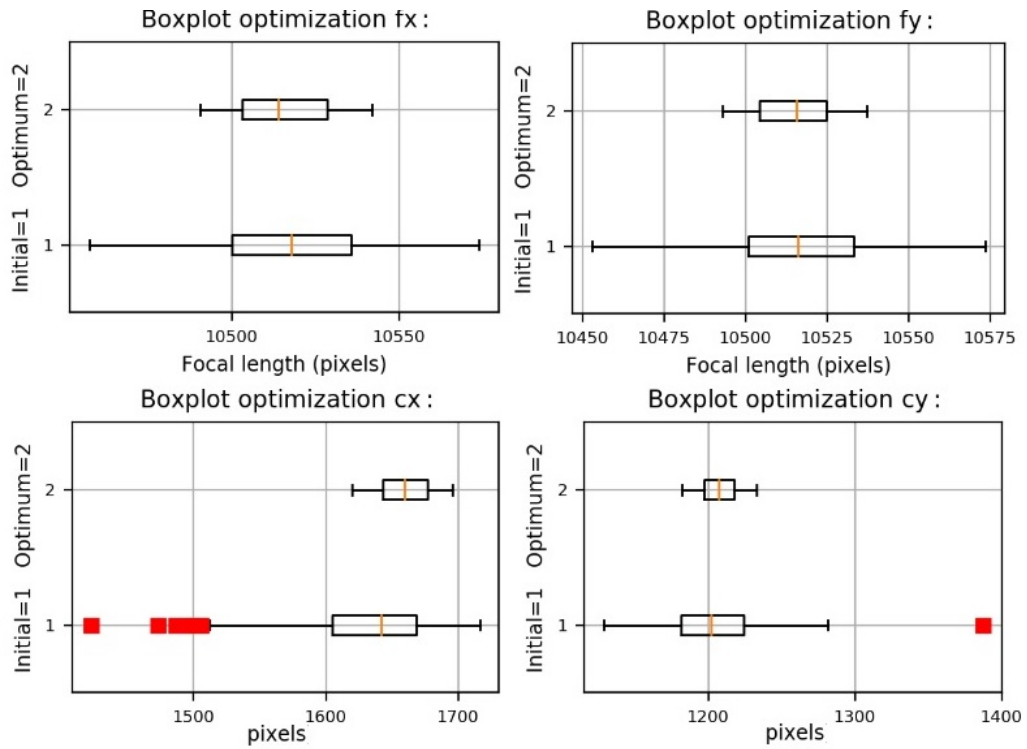


Figure 5.32: Optimization of the camera matrix calibration parameters for the Right camera.

radial distortion exists in the image after undistortion process, and this is not the case.

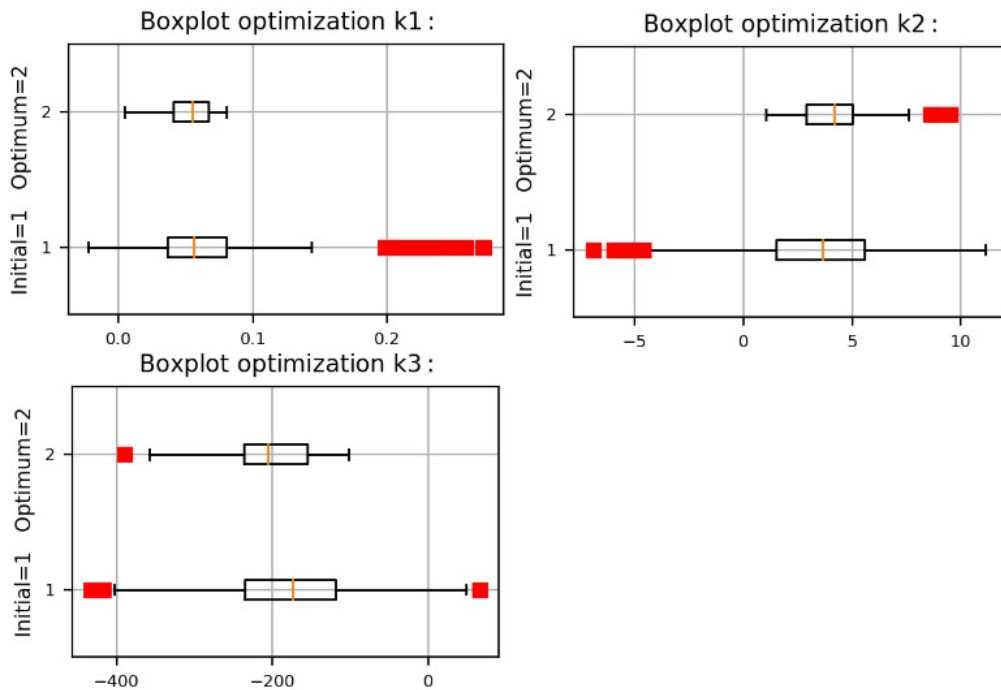


Figure 5.33: Optimization of the radial distortion coefficients for the Left camera.

- **Reprojection error:** it can be observed that the reprojection error starts in 0.84 and finishes in 0.85 for the Left camera. On the other hand, the Right camera starts with a

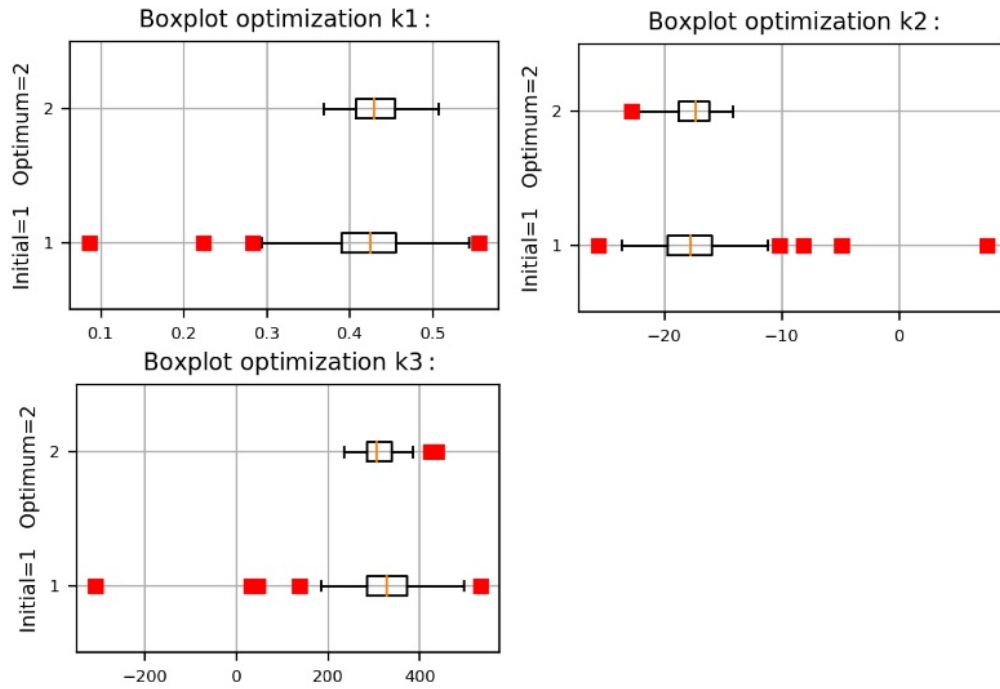


Figure 5.34: Optimization of the radial distortion coefficients for the Right camera.

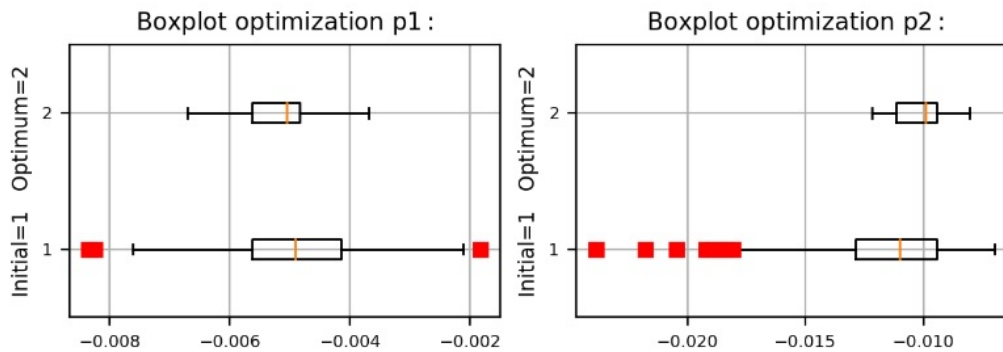


Figure 5.35: Optimization of the tangential distortion coefficients for the Left camera.

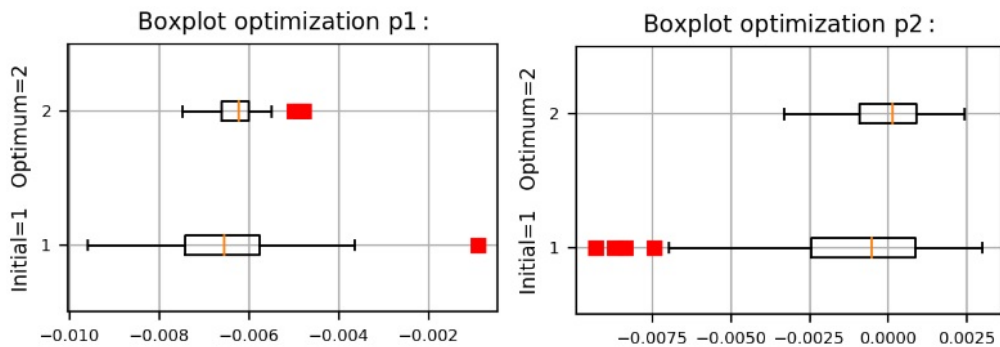


Figure 5.36: Optimization of the tangential distortion coefficients for the Right camera.

reprojection error of 0.98 and finishes with 0.99. It can be observed that the values are always under the maximum allowable mean reprojection error. However, it is really important to comment that having a reprojection error under a minimum does not means

that the calibration has been done correctly, as it was observed before that the principal point coordinates of the cameras were not feasible. In spite of the frustration of having obtained an incorrect result for a parameter, it has been showed that this software is able to analyze the nature of the results and their errors in order to have a flexible and robust tool to perform a calibration in a non-professional laboratory.

5.3.3.2 Stereo camera calibration

This application is used in order to calibrate the stereo camera formed by the two previously calibrated cameras. As this application uses the individual calibration of the cameras, it is obvious that the result will be polluted by the error obtained in the principal point coordinates of the cameras. However, the results will be equally commented and analyzed in order to show its physical meaning and behavior. Moreover, the initialization file is showed below in Figure 5.37 in order to show the optimization steps performed in this case.

In this case, a total of $N = 19$ stereo pair images is used for the stereo camera calibration. As the number of images per subset is maintained as $m = 18$, there is a total of $C = 19$ subsets of images (Equation 4.4). The optimization steps are the same than the ones used in Section 5.2. The results obtained after and before the optimization are showed in Figure 5.38 and Table 5.13 for the translation vector and Figure 5.39 and Table 5.14 for the rotation vector. It can be observed that there is a dominant translation component in the x-direction, while the translation in the other axis is at least one order of magnitude lower. Moreover, the rotation about any axis is always less than $1\hat{\text{A}}^\circ$, and the dominant component of the rotation vector is the one about the y-axis. The optimization process reduces the standard deviation of every parameter taken into account in the stereo camera calibration. As it was commented in 5.2, the outliers of the components with less weight in the translation and rotation are more difficult to eliminate, but it is not something prohibitive. Finally, it is also remarkable that initially there were 19 subsets of images and after optimization there are just 10 subsets. A question about the minimum and optimum number of subsets may arise when this cases appear. However, there is not a global answer to this question, as there is not a fixed rule to impose the number of images per subset, the minimum number of subsets or the optimum initial number of subsets, it depends on the case and the nature of the project. However, it is well known that statistics becomes effective when the number of entries is large.

Parameter	Initial set		Optimum set	
	mean	deviation	mean	deviation
$ T $	82.96	0.074	82.94	0.02
T_x	-82.51	0.076	-82.50	0.02
T_y	-0.0028	0.012	-0.028	0.01
T_z	8.58	0.035	8.58	0.016

Table 5.13: Translation vector norm and components optimization.

```

# IN ORDER TO INITIALIZE THE CALIBRATION FOLLOW THESE STEPS:
# - Save the calibration images in a folder:
#   - 'images' if the calibration is performed just for one
camera.
#   - 'images/calibrate_left' & 'images/calibrate_right' for a
stereo calibration.
# - Complete this file to configure the optimization.

[ Calibration ]
# Set 1 the kind of calibration desired

Individual calibration = 0
image format = 'tif'

Stereo calibration = 1
image format Left= 'tif'
image format Right= 'tif'

square size = 3.75 # size of pattern square in mm
mesh width = 21
mesh height = 14

images per set = 18

camera matrix parameters = fx, fy, cx, cy
distortion coefficients parameters = k1, k2, k3, p1, p2

[ Stereo Optimization ]

# Specify the stereo camera calibration steps and constrains.
# Elements to optimize:
# - Stereo RE (Stereo reprojection error).
# - Translation vector components and norm (Tnorm, Tx, Ty,
Tz).
# - Rotation vector components and norm (RV_norm, RVx, RVy,
RVz).
# Max error for the parameters
# - X for the RE in pixels
# - Y for the rest of parameters in max_deviation/average
[[ Calibration steps ]]

    number steps = 3

    [[[ step 1 ]]]
        parameters = stereo_re, # (RE)
        max error = 3, # max mean reprojection error

    [[[ step 2 ]]]
        parameters = Tnorm,
        max error = 0.75,

    [[[ step 3 ]]]
        parameters = RVnorm,
        max error = 0.75,

```

Figure 5.37: Example of initialization file for istereo camera calibration.

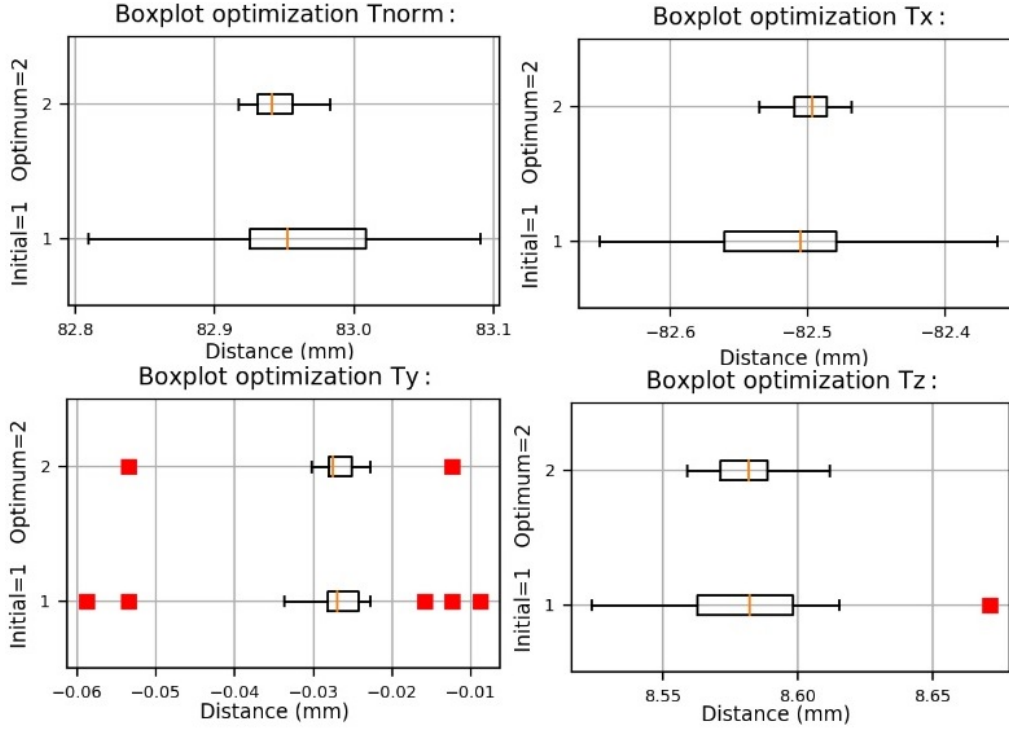


Figure 5.38: Optimization of the translation vector norm and components for the stereo camera.

Parameter	Initial set		Optimum set	
	mean	deviation	mean	deviation
$ R $	0.015	9.85e-5	0.015	1.17e-5
T_x	-0.009	1.89e-5	-0.009	1.16e-5
T_y	0.0113	0.00013	0.011	3.11e-5
T_z	-0.003	3.87e-5	-0.003	1.19e-5

Table 5.14: Rotation vector norm and components optimization.

5.3.3.3 3D reconstruction

As it has been commented, the values obtained in the principal point coordinates of the cameras makes that the stereo calibration of the camera cannot be considered reliable to do a 3D reconstruction. However, knowing the experimental setup and analyzing the values obtained in the stereo calibration, it can be observed that the cameras are mainly displaced in the x-direction and almost no rotated. Therefore, using the distortion model obtained for each camera and assuming that the pictures are row aligned and that the cameras have the same focal length, it is feasible to assume that the images are pretty close to be considered as rectified. Then, as in Section 5.2, the application for the 3D reconstruction with an uncalibrated camera will be used in order to check if the rest of functions of the software work properly. Moreover, in this case the stereo camera is not completely uncalibrated, as the values of the focal length will be assumed as valid and the baseline will be the one obtained in the stereo camera calibration. The only values that will be discarded are the principal point coordinates, as the principal point will be assumed to be at the center of the image. Therefore, the Q matrix used for the 3D

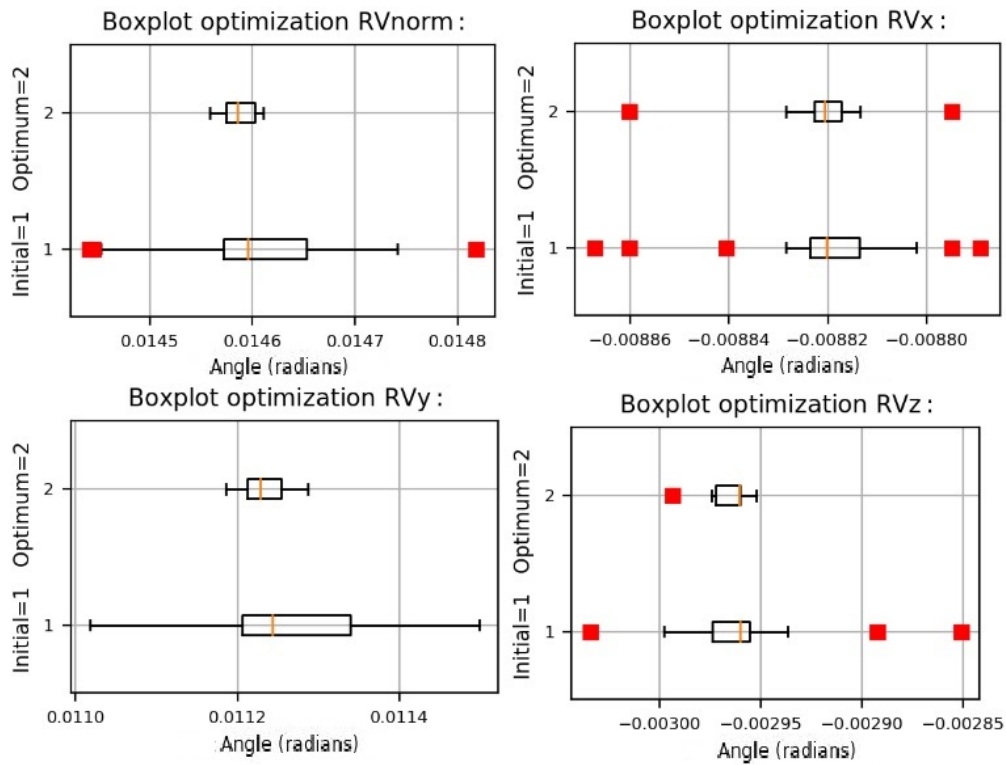


Figure 5.39: Optimization of the rotation vector norm and components for the stereo camera.

reconstruction can be obtained. The target images of the 3D reconstruction can be seen in Figure 5.40 and 5.41 (two reconstructions of the same sample in different positions will be done).



Figure 5.40: Images used for the first reconstruction of the VKI Plasmatron sample.

The next step is to obtain the disparity map of the target images. The inputs of the disparity function provided by OpenCV can be observed in Table 5.15. Moreover, an scaling process for disparity is needed in order to fit the reconstructed object with the real one. It is recommended

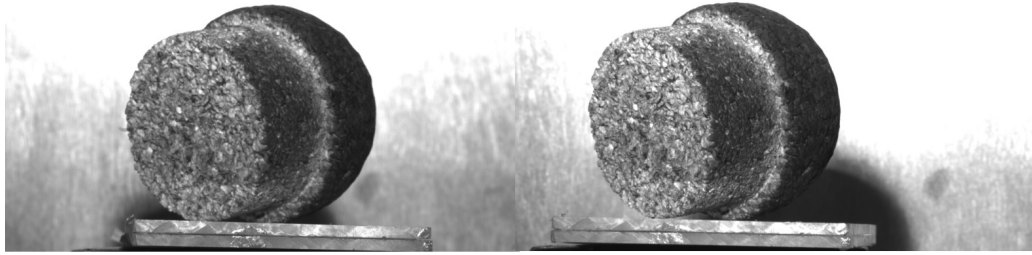


Figure 5.41: Images used for the second reconstruction of the VKI Plasmatron sample.

to calibrate the disparity configuration of a stereo camera with a defined 3D object, but doing it directly with the sample is possible as its size is known. In this case, the scaling process has been performed normalizing the disparity map and multiplying by a certain K scale factor.



Figure 5.42: Normalized disparity map of the VKI Plasmatron sample.

\StereoSGBMcreate inputs	
minDisparity	0
numDisparities	16*9
blockSize	11
P1	8*3>window_size**2
P2	32*3>window_size**2
disp12MaxDiff	1
uniquenessRatio	10
speckleWindowSize	100
speckleRange	32

Table 5.15: OpenCV StereoSGBMcreate inputs for disparity calculation.

Once the final disparity map has been obtained it is time to obtain the 3D point cloud of the

model following the steps explained in Chapter 4. The results are two .PLY files which are opened in MeshLab. Figures 5.43 and 5.43 show the final result, which seems to be quite accurate. Finally, Meshlab allows to clean the point cloud and perform the surface reconstruction from the 3D point cloud, as it can be observed in Figure 5.45.

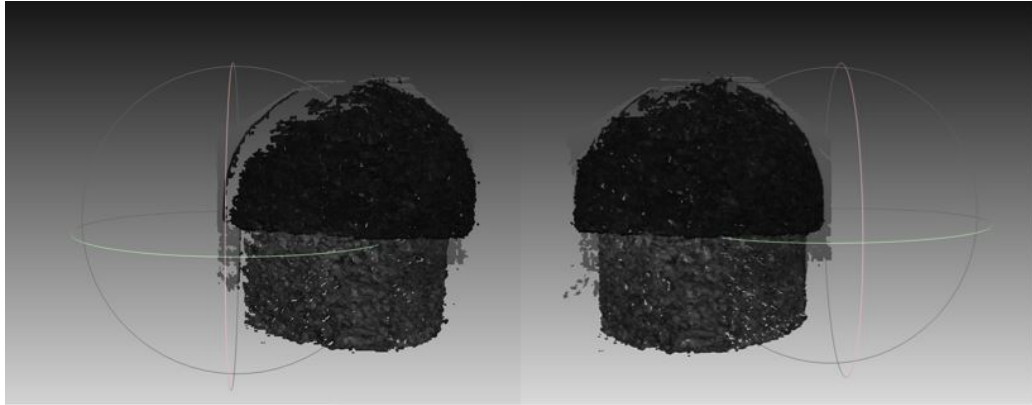


Figure 5.43: 3D point cloud result for the first reconstruction of the VKI Plasmatron sample.

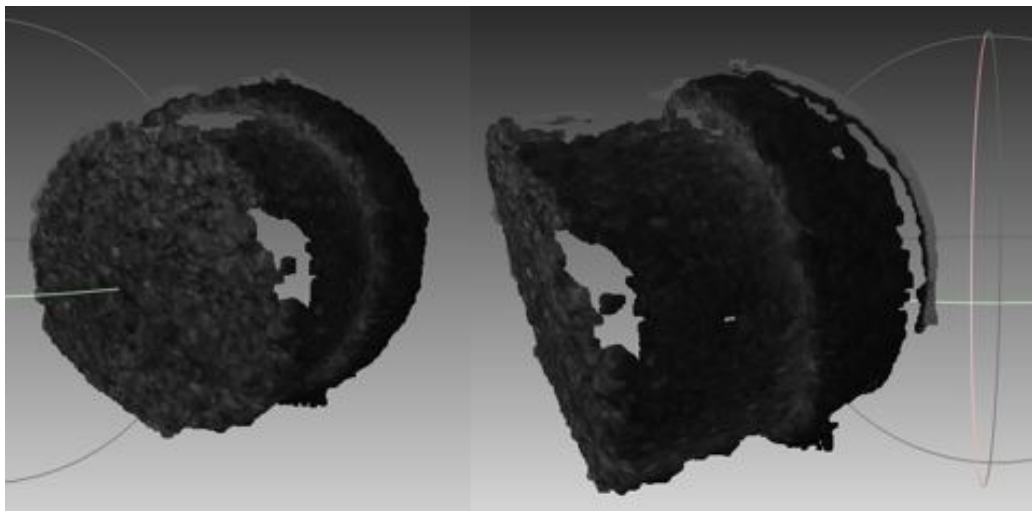


Figure 5.44: 3D point cloud result for the second reconstruction of the VKI Plasmatron sample.

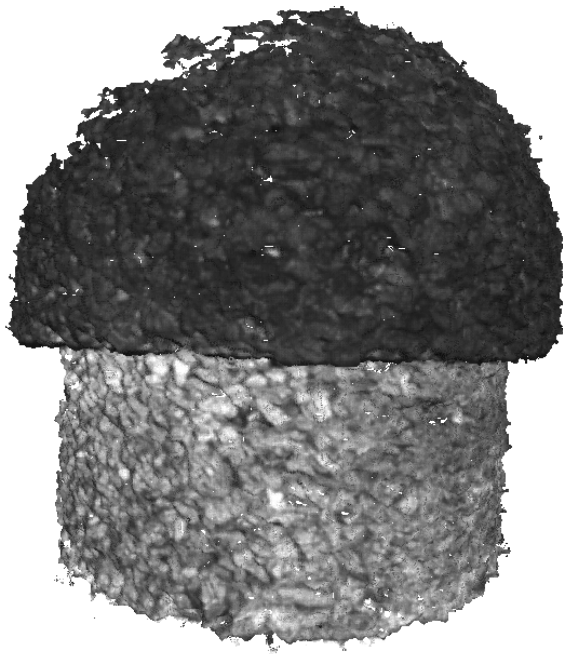


Figure 5.45: 3D mesh result for the first reconstruction of the VKI Plasmatron sample.

6 Conclusions

This project can be divided into two main parts. First, the development of an open-source based software: *Stereo 3D Reconstruction*. Second, the experimental part in which the software has been tested in order to understand its functionality and potential.

6.1 Conclusions and future work on *Stereo 3D Reconstruction*

The characteristics of *Stereo 3D Reconstruction* have been studied all along this thesis. Now it is time to summarize the most important ones and extract the conclusions.

- User defined initialization makes the software adaptable to each project depending on its nature and possibilities.
- a user-friendly environment which allows the use of the software avoiding the contact with the code provides a flexible tool to use at any facility. Moreover, having a folder with the results permits the user to check the quality of the results.
- The integrity of the software is based on the implementation of a robust statistics based optimization. Moreover, this optimization is also configurable, which allows to improve the results of the experiments and find the optimum settings.
- Two paths to perform a three dimensional reconstruction have been implemented in order to have more possibilities and to validate the results.

Moreover, it would be not only unrealistic but also conformist to assume that the software is completely finished. There is a solid foundation for future developments that will make this program more robust and complete. Some ideas for future developments are exposed as follows:

- There are several options to implement a more flexible software. For example, having the possibility to select manually the distortion mathematical model. Moreover, the parallel calculation can be also implemented to be configured from the initialization.
- The disparity map calculation should be implemented to be global, which means that it would be independent of the inputs of the function and its calibration. In theory, no calibration should be done for the disparity ($d = xl - xr$), but OpenCV function changes the results depending on the initial inputs. Then, a deep study of StereoSGBM is highly recommended as well as the development of an algorithm to uncorrelate the disparity calculation from its calibration.
- The restriction of using just two equal cameras comes from the nature of OpenCV functions. Therefore, making the software able to use more than two cameras and not having to use the same model with the same configuration will be valuable.

6.2 Conclusions and future work on experiments

Two different experiments have been done in this project. One with OpenCV images and another performing the whole process with the VKI equipment. The conclusions extracted from both of them are presented below:

- The functions implemented on every step of the reconstruction work correctly and provide promising results.
- Along the experiments, it has been observed that the relative position of the cameras, the methodology and the materials used in the calibration process influence the results.
- The lack of time to do experiments with two equal cameras has made impossible to improve the inaccurate results obtained in the calibration. However, calibration application has been proved to work properly with other equipment. Moreover, obtaining imprecise results has shown the value of having a tool to analyze the results for each step and avoid endanger the integrity of the final reconstruction.
- The cork sample has been successfully reconstructed, proving the potential of the code for future applications.

Finally, the future work of the experimental part of the project will consist on a whole software validation. In order to do that, when the upgrades of the software are completed, a 3D reconstruction of an object should be done using the two reconstruction paths implemented and comparing their results. Moreover, once this software is validated, an experiment in the VKI Plasmatron will be performed. This experiment will consist on reconstructing a VKI sample during a Plasmatron experiments. The images taken for each time step will be used to study the three dimensional surface recession of the thermal protection material.

6.3 UPV Photogrammetry collaboration

It is completely necessary to recognize the contribution of the UPV to this project. The support of the professionals that have collaborated with the VKI has been totally essential to understand the physics and nature of photogrammetry. Moreover, the software **FotoGIFLE** (from Pr. JL. Lerma) will be used as a professional reference to compare the future results obtained with *Stereo 3D Reconstruction* and check their accuracy. This will allow to value the feasibility of using photogrammetry in the study of TPS.

7 Budget

This chapter intends to value the cost of the work invested in the development of the present project. For this, the number of hours dedicated by the personnel involved in the project will be taken into consideration, as well as the materials and resources involved. The cost of the computer equipment used will not be taken into account, as it is fully amortized. The unit to quantify the effort invested by the personnel involved will be the *hour of work*. The monetary unit will be *euros per hour of work*.

The resources of the project can be included in two large groups: human and material. These are specified in Table 7.1. The breakdown of the human resources and personal costs can be seen in Tables 7.2 and 7.3. Then, another breakdown of the costs related with the materials employed can be observed in Table 7.4. Finally, considering human and material costs, the total cost of the project is 244045 euros.

Resources	
Human	Material
Engineer	Computer cluster
Advisor	2 x DSLR Cameras
Supervisor #1 (VKI)	2 x Lenses
Supervisor #2 (VKI)	Tripod
Supervisor #1 (UPV)	2 x SP-12000-CXP4 Cameras
Supervisor #2 (VK1)	

Table 7.1: Human and material resources.

Description	Units (hours)	Unit cost (euro/hour)	Cost (euro)
Engineer	1000	80	80000
Advisor	400	100	40000
Supervisor #1 (VKI)	200	150	30000
Supervisor #2 (VKI)	200	150	30000
Supervisor #1 (UPV)	200	150	30000
Supervisor #2 (VK1)	200	150	30000
TOTAL			240000

Table 7.2: Partial and total cost of the human resources of the project.

Description	Units (hours)	Unit cost (euro/hour)	Cost (euro)
Software development	500	80	40000
Experiments	50	80	4000
Analysis of results	150	80	12000
Report	300	80	24000
TOTAL			80000

Table 7.3: Breakdown of the personal cost of the project.

Description	Units	Units (hours)	Unit cost (euro/hour)	Cost (euro)
Computer cluster	1	500	5	2500
DSLR Camera	2	45	10	900
Lenses	2	45	5	450
Tripod	1	45	1	45
SP-12000-CXP4 Camera	2	5	15	150
TOTAL				4045

Table 7.4: Breakdown of the material resources of the project.

References

- [1] H. J. Allen and A. J. Eggers. *A study of the motion and aerodynamic heating of ballistic missiles entering the earth's atmosphere at high supersonic speeds*. Technical Report TR-1381, NACA, 1958.
- [2] B. Laub and E. Venkatapathy. *Thermal Protection System Technology and Facility Needs for Demanding Future Planetary Missions*. International Workshop on Planetary Probe Atmospheric Entry and Descent Trajectory Analysis, 2003.
- [3] Julio Magalhães. *Galileo Probe Heat Shield Ablation*. NASA Ames Research Center.
- [4] Julio Magalhães. *The Galileo Probe Spacecraft*. NASA Ames Research Center.
- [5] A. O. Quinn. *Legal Aspects of Photogrammetric Measurements for Surveying and Mapping*. Quinn Associates, Inc, 1983. 453–456.
- [6] Howard IP and Rogers BJ. *Binocular vision and stereopsis*. New York: Oxford University Press, 1995.
- [7] John D. Anderson Jr. *Hypersonic and High-Temperature Gas Dynamics, Second Edition*. American Institute of Aeronautics and Astronautics, May 21, 2001.
- [8] Mark Sirangelo. *NewSpace 2011: Sierra Nevada Corporation*. Spacevidcast.
- [9] Mark Sirangelo. *Flight Plans and Crews for Commercial Dream Chaser's First Flights: One-on-One Interview With SNC VP Mark Sirangelo (Part 3)*. AmericaSpace.
- [10] O. Chazot and F. Panerai. *Hypersonic Nonequilibrium Flows: Fundamentals and Recent Advances, chapter High-Enthalpy Facilities and Plasma Wind Tunnels for Aerothermodynamics Ground Testing*. Progress in Astronautics and Aeronautics at AIAA Reston, Virginia (USA), 2015.
- [11] A. Pope and K. L. Goin. *High Speed Wind Tunnel Testing*. Wiley, 1965.
- [12] D. W. Gates P. R. Dennis C. R. Smith and J. B. Bond. *Plasma Jet Technology*. Technology Survey Report SP-5033.
- [13] Bernd Helber. *Material response characterization of low-density ablators in atmospheric entry plasmas*. Rhode-St-Genese, belgium von Karman Institute, 2016. 225 pages.
- [14] Henry Phelps Gage. *Optic projection, principles, installation, and use of the magic lantern, projection microscope, reflecting lantern, moving picture machine*. 1914.
- [15] Todd Gustavson. *Camera: a history of photography from daguerreotype to digital*. New York, New York: Sterling Publishing Co, 2009.
- [16] Helmut Gernsheim. *A Concise History of Photography. Third edition*. Mineola, New York: Dover Publications, 1986.
- [17] Kodak. *History of Kodak*.

- [18] *Camera Exposure: Aperture, ISO and Shutter Speed*. URL: <https://www.cambridgeincolour.com/tutorials/camera-exposure.htm>.
- [19] *Circles of Confusion for Digital Cameras*. URL: http://www.dofmaster.com/digital_coc.html.
- [20] G. Bradski and A. Kaehler. *Learning OpenCV: Computer Vision with the OpenCV Library*. Sebastopol, CA O'Reilly, 2008.
- [21] *What Is Camera Calibration? - MATLAB and Simulink - MathWorks España*. Sept. 11, 2018. URL: <https://es.mathworks.com/help/vision/ug/camera-calibration.html#buvr2qb-2>.
- [22] R. Guerchouche and F. Coldefy. *Camera Calibration Methods Evaluation Procedure for Images Rectification and 3D Reconstruction*. 2008.
- [23] O. Faugeras and G. Toscani. *The calibration problem for stereo*. Proceedings of the International Conference on Computer Vision and Pattern Recognition.
- [24] O. Faugeras and G. Toscani. *Camera calibration for 3d computer vision*. International Workshop on Machine Vision and Machine Intelligence, 1987.
- [25] F. Coldefy R. Guerchouche and T. Zaharia. *Accurate camera calibration algorithm using a robust estimation of the perspective projection matrix*. SPIE Mathematics of Data/Image Pattern Recognition, Compression, and Encryption with Applications, 2006.
- [26] Z. Zhang. *Flexible camera calibration by viewing a plane from unknown orientations*. International Conference on Computer Vision (ICCV), 2006.
- [27] Z. Zhang. *A flexible new technique for camera calibration*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2000.
- [28] B. Bottin. *Aerothermodynamic Model of an Inductively-coupled Plasma Wind Tunnel*. PhD thesis, Universite de Liege, von Karman Institute for Fluid Dynamics, 1999.
- [29] A. F. Kolesnikov. *Conditions of Simulation of Stagnation Point Heat Transfer from a High-enthalpy Flow*. Conditions of Simulation of Stagnation Point Heat Transfer from a High-enthalpy Flow, 1993.
- [30] P.F. Barbante and O. Chazot. *Flight Extrapolation of Plasma Wind Tunnel Stagnation Region Flowfield*. J Thermophys Heat Transfer, 2006.
- [31] Dave Kuhlman. *A Python Book: Beginning Python, Advanced Python, and Python Exercises*. Rextx, 2012.
- [32] John V. Guttag. *Introduction to Computation and Programming Using Python: With Application to Understanding Data*. MIT Press, 2016.
- [33] Daniel G. Shafer. *Python Streamlines Space Shuttle Mission Design*. Python Software Foundation, 2003.
- [34] Alexander Mordvintsev and Abid K. *OpenCV-Python Tutorials Documentation. Release 1*. 2017. 273 pages.

- [35] Oleksandr Semeniuta. *Analysis of Camera Calibration with Respect to Measurement Accuracy*. Procedia CIRP, 2016. 765–770.
- [36] Ian Cook Graham Upton. *Understanding Statistics*. Oxford University Press. p. 55, 1996. 273 pages.
- [37] R.I. Hartley. *Theory and Practice of Projective Rectification*. IJCV 35 2, 1999. 115–127.
- [38] Allen R. Grenleaf. *Photographic Optics*. The MacMillan Company, 1950. 25–27.
- [39] J. Heikkila and O. Silven. *A Four-step Camera Calibration Procedure with Implicit Image Correction*. IEEE International Conference on Computer Vision and Pattern Recognition, 1997.