



**Politecnico  
di Torino**

**POLITECNICO DI TORINO**  
**INGEGNERIA PER L'AMBIENTE E IL**  
**TERRITORIO**

**Tesi di Laurea Magistrale**

**Reconstruction of the trajectories of rock fragments induced by blasts  
in an open pit mine by 3D analysis  
(Valdilecha Quarry, Madrid, Spain)**

Relatore: Marilena Cardu

Candidato: **Carmela Pagone**

Tutors: Luis Iglesias

Santiago Gomez

Academic Years 2022-2023



# Table of contents

Figures.....	4
Tables.....	6
Abstract.....	7
1. Introduction.....	8
2. Scope and objectives.....	9
3. Background.....	10
3.1 Previous works and TRLs.....	10
3.2 Photogrammetry.....	12
3.2.1 High speed photography.....	12
3.3.1 From 2D to 3D analysis.....	13
3.4 DIGIECO project.....	14
4. Methodology.....	18
4.1 Camera selection process.....	18
4.1.1 Features of the cameras.....	18
4.1.2 Motion analysis of the blast face.....	22
4.1.3 Optimal location of the cameras.....	33
4.1.4 Multi-objective optimization.....	41
4.1.4 Camera selection.....	48
4.2 Photogrammetric analysis.....	50
4.2.1 Stereo camera calibration.....	50
4.2.2 Object tracking.....	54
4.3 Trajectory reconstruction.....	54
5. Laboratory test.....	61
5.1 Equipment.....	61
5.2 Realization of the photogrammetric sets.....	66
5.3 Lens distortion correction.....	67
5.4 Result and discussion.....	69
6. Field trial.....	76
6.1 Features of the bench.....	76
6.2 Placement of the cameras.....	77

6.3 Calibration process .....	80
6.4 Blast recordings .....	83
7. Discussion .....	86
7.1 Issues .....	87
7.2 Recommendations .....	89
8. Conclusion .....	90
Bibliography and references .....	92
Appendix A .....	95

## Figures

Figure 1: KTAs and KSAs .....	17
Figure 2: Location of the benches under exploitation in Valdilecha Quarry .....	22
Figure 3: Simplified representation of the bench .....	23
Figure 4: Grain size distribution of Goltas Limestone Quarry, Turkey .....	24
Figure 5: Basic forces acting upon the flyrock fragment. Adapted figure from (Stojadinović et al., 2011). .....	25
Figure 6: Simplified 3D representation of the analysis of a fragment's trajectory .....	27
Figure 7: Coordinate system .....	28
Figure 8: 2D representation in XZ plane .....	28
Figure 9: 2D representation of the initial position of the fragment in XY plane .....	29
Figure 10: 2D representation of fragment's trajectory in X'Y' plane .....	30
Figure 11: Fragments projection .....	31
Figure 12: Fragments' trajectories .....	32
Figure 13: Coloured representation of fragments on the ground .....	32
Figure 14: Empirical Cumulative Distribution Function .....	33
Figure 15: Schematical representation of a possible placement of the cameras with their FOVs .....	35
Figure 16: Schematical representation of a possible placement of the cameras, their distances from the fragments .....	36
Figure 17: Trapezoidal function .....	40
Figure 18: Pareto front .....	44
Figure 19: Pareto front, utopia point and trade-off solution .....	45
Figure 20: Optimal position and orientation of the cameras .....	47

Figure 21: Graphical representation of the relationship between fragment size-max distance and pixel size-focal length .....	48
Figure 22: The three rotation angles of the right camera with respect to the left camera.....	51
Figure 23: Example of a calibrated stereopair with Camera Calibrator App from MATLAB.....	53
Figure 24: Zoom in of the calibrated chessboard patterns .....	53
Figure 25: Trajectory seen from the two cameras. Each position (1-5) forms a stereoscopic pair....	55
Figure 26: Collinearity condition for the two cameras .....	56
Figure 27: Camera Sony DSC-RX100 VII .....	62
Figure 28: TSL08CN00 Slim Tripod.....	63
Figure 29: Ball as tracked object.....	63
Figure 30: Chessboard pattern .....	64
Figure 31: Setup of the equipment.....	65
Figure 32: Relationship between radial lens distortion and corrections to x and y coordinates from (Wolf, 2014).....	68
Figure 33: Extrinsic parameters obtained with calibration .....	70
Figure 34: Mean reprojection errors per image .....	71
Figure 35: Application of the motion based multi-object algorithm in frame 108 (Left camera) .....	71
Figure 36: 2D representation of the centroids coordinates - Left camera.....	72
Figure 37: 2D representation of the centroids coordinates - Right camera .....	72
Figure 38: Object trajectory in 2D by the left camera .....	73
Figure 39: Object trajectory in 2D by the right camera .....	74
Figure 40: Trajectory of the object in 3D .....	74
Figure 41: 3D trajectory of the object with cameras and chessboards locations .....	75
Figure 42: Design of the blast of the bench n.3 .....	77
Figure 43: Optimal placement of the cameras in the field.....	78
Figure 44: Overview of the site.....	79
Figure 45: Setup of the equipment.....	79
Figure 46: Zoom in of the location of the cameras .....	80
Figure 47: Calibration pattern used in the field .....	81
Figure 48: Example of calibration, chessboard in position 1 .....	81
Figure 49: Example of calibration, chessboard in position 2.....	82
Figure 50: Stereo calibration result, with the chessboard in position 2 .....	82
Figure 51: Zoom in of the stereo calibration result.....	83
Figure 52: Example of representative frames of the blast (Left camera).....	84

Figure 53: Example of representative frames of the blast (Right camera) .....	85
Figure 54: Example of calibration issues .....	87
Figure 55: Application of the motion based multi-object algorithm in frame 124 (Left camera) .....	88
Figure 56: Application of the motion based multi-object algorithm in frame 208 (Left camera) .....	89

## Tables

Table 1: TRLs scale .....	11
Table 2: Main differences between TRL 3 and TRL 4 based on (TEC-SHS, 2008) .....	11
Table 3: Relationship between 2050 roadmap, KTAs, KSAs and challenges from (European Union, 2021) .....	16
Table 4: Type of cameras and related HFR availability .....	20
Table 5: Geometrical parameters of the bench .....	23
Table 6: Physical constants .....	26
Table 7: Relation between cameras features and geometrical parameters.....	49
Table 8: Intrinsic and extrinsic parameters coming from the calibration .....	70
Table 9: Geometrical parameters of the bench under investigation.....	76
Table 10: Comparison between the theoretical and in-field values .....	80

## **Abstract**

This work is based on a proof of concept, whose main goal is to determine whether it is feasible to rebuild rock fragment trajectories ejected from blasts.

The initial focus is on creating and improving methods for precisely estimating the three-dimensional motions of rock fragments using photogrammetric analytic tools to improve the safety in mining operations. The objective is to test the effectiveness of the suggested methodologies and establish their applicability in actual mining scenarios, passing from the conceptual stage to an experimental pilot phase. The reconstructed trajectory methodology must be successful in moving from proof of concept to actual use in order to increase safety precautions and provide information that can be used to optimize blasting procedures.

To achieve this, two cameras with High Frame Rate (HFR) capabilities are required to record the blasts, which will be carried out in the Valdilecha Quarry near Madrid (Spain). As proposed by the proof of concept, the photogrammetric analysis has been employed to extract the 3D motion of the objects, allowing the estimation of their movements using computational models by analysing each frame of the recorded videos.

The Sony DSC-RX100 VII is chosen as the preferred camera for the study. It satisfies the requirements for HFR availability (used to capture slow-motion sequences), frame rate, pixel size and shooting time, ensuring optimal data capture for the application of the photogrammetric techniques.

Finally, the developed approach has been tested in the field to reach the experimental pilot phase. As common in experimental work, moving from the controlled laboratory environment to the field presents its own difficulties. As expected, there were several problems in the analysis of the field recordings, due to the different environmental conditions. However, these difficulties have offered important information for enhancing the process in future iterations. The process can be further refined to produce more accurate and dependable outcomes in real-world circumstances through ongoing improvement and adaption.

# 1. Introduction

One of the most important steps in guaranteeing safety and improving mining operations is reconstructing the paths of the rock particles that are released during a blast. The potential benefits of moving a proposed methodology from theoretical viability to actual implementation and testing are enormous. According to (European Union, 2021) numerous advantages of fragment movement prediction include increased safety, better extraction planning, reduced damage to nearby structures, blast design optimization, lower costs, and better environmental management.

First of all, forecasting the trajectory of pieces makes it possible to identify risk areas and take the appropriate safety measures to safeguard people, nearby buildings, and the environment. Potential dangers can be reduced by putting in place safety barriers, maintaining proper safety distances, and creating efficient evacuation procedures.

Secondly, a more effective extraction planning is made possible by knowing the trajectory of the fragments before. Ground-moving machinery can be placed strategically to speed up the removal of bulk materials and shorten the time required for restoration following blasting activities.

Thirdly, precise trajectory prediction reduces damages to nearby structures. By locating possible impact zones, precautionary actions can be taken to lower the chance of damage.

Additionally, the optimization of blast design can be affected by the prediction of fragment trajectories. Previous blast data can be used to improve future designs by adjusting drilling and explosive settings to maximize effectiveness and reduce undesirable consequences.

Furthermore, good fragment trajectory prediction helps mining operations working with lower costs, as they can be greatly decreased by preventing costly damage or delays. Cost savings are further increased by minimizing damages to nearby structures and shortening the time needed for cleaning and repair.

A methodology based on the use of photogrammetric tools has been proposed to achieve these goals. This approach relies on the investigation of face movement (Jiménez, 2022) carried out on previous blasts in El Aljibe Quarry (Almonacid de Toledo, Spain). In particular, the work of (Jiménez, 2022) presents 2D and 3D approaches for the reconstruction of fragment trajectories ejected from blasting by using photogrammetric techniques.

Starting from that, this project intends to develop the proposed methods for reconstructing rock fragment trajectories to increase safety and operational efficiency in mining operations by moving

from theoretical viability to real implementation and testing. For this reason, laboratory experiments have been conducted to validate and improve the approach together with a field study carried out in Valdilecha Quarry in collaboration with Hanson Company to evaluate the viability and efficacy of the established methodology.

This work is also in accordance with the DIGIECO QUARRY project, which focuses on integrating digital technology into the mining industry for improved safety and sustainability. The project intends to maximize quarry operations and reduce their negative environmental effects by employing advanced digital methods like photogrammetry and data analysis.

## **2. Scope and objectives**

This chapter addresses the work's main objectives, which rely on the reconstruction of rock fragment trajectories to improve the safety in mining operations.

The major objective is the transition from a *concept validation* phase to the *experimental pilot* phase. The proposed approaches and techniques have been evaluated for their viability and efficacy by raising the technological readiness level (TRL) from 3 to 4 and carrying out controlled experiments to test and validate them. This will stand as the groundwork for future developments and applications of the reconstructed trajectory methodology in real-world mining environments.

Furthermore, another important goal is to use a dual-camera configuration to record three-dimensional data to increase the accuracy of the photogrammetric analysis. To pass from 2D to 3D analysis, the addition of a second camera enables the simultaneous capture of videos from different perspectives.

Moreover, the work aims to develop a comprehensive methodology that makes it possible to precisely recreate rock fragment trajectories after an explosion. Photogrammetric analytic tools are employed to predict the three-dimensional paths of rock fragments and better comprehend their behaviour. The 3D reconstructed trajectories can be used to determine safe operating distances, identify potential hazard zones, and improve blast design. By achieving this objective, valuable information about the movement and dispersion of rock fragments can be acquired, enhancing mining operations' safety standards.

The work additionally points to examine the face movement as a result of previous blasts conducted in El Aljibe Quarry.

Finally, this investigation will provide fundamental information about the flight paths and dispersion patterns of the fragments, which will aid in a better understanding of the dynamics of the quarry, increasing the overall safety and efficacy of subsequent mining operations.

### **3. Background**

The background of this work is based on earlier research in the field of photogrammetry, a method for extracting three-dimensional data from two-dimensional pictures or videos. Additionally, the DIGIECO project has a big impact on the goals of the study as it aims to integrate digital technologies into the mining sector for increased sustainability and safety.

#### **3.1 Previous works and TRLs**

As previously mentioned, this work is based on the proof of concept (*Conceptual pilot*) developed by (Jiménez, 2022) and aims to denote a more advanced state of technological development.

A defined technique that evaluates the maturity of technologies, throughout the program's acquisition phase (TEC-SHS, 2008) is the Technology Readiness Levels (TRLs). The scale goes from 1 to 9, which indicates the most advanced technology. TRLs make it possible to have consistent information about the technical maturity of many technology fields. They are considered a useful tool for decision-making process for funding and implementing new technologies and offer a common knowledge of the state of technology and support risk management.

The TRLs stages considered in the work are two: TRL 3 and TRL 4. According to the table below (Table 1), this work belongs to the earliest stage of the development phase, i.e., *Experimental pilot* (Héder, 2017). In this scale, TRL 4 and TRL 3 are two key stages, and each has its own goals and focuses.

Table 1: TRLs scale

Phase	TRL	Hardware	Software
Research	1	Basic principles	
	2	Concept and application formulation	
	3	Concept validation	
Development	4	Experimental pilot	
	5	Demonstration pilot	
	6	Industrial pilot	
Deployment	7	First implementation	Industrialization detailed scope
	8	A few records of implementation	Release version
	9	Extensive implementation	

The technology has advanced in TRL 4, which emphasizes performance validation, and quantitative characterization in order to get ready for future system applications. TRL 3, on the other hand, focuses on proving the viability and practicality of the concept, while combining analytical and experimental approaches. TRL 4 denotes a greater degree of technological maturity and can be considered the first step towards real-world applications.

The table below (Table 2) shows the main distinctions between TRL 4 and TRL 3.

Table 2: Main differences between TRL 3 and TRL 4 based on (TEC-SHS, 2008)

TRL 3	TRL 4
Beginning of research and development activities	Based on TRL 3
Verification of the proof of concept with analytical investigations and lab-based experiments	Integration of fundamental technological components to ensure their functionality
Use of analytical and experimental methods	Consistent validation with the requirements of potential system applications
Need of physical experimental validation for complex systems or phenomena	Quantitative characterization of applications and performance definition through laboratory demonstrations

In conclusion, the main goal of the TRL 3 is to prove that a technology and its idea are workable, while the TRL 4 provides a more refined understanding of the performance requirements for field trials. Considering all these elements, this study fits into TRL 4 and it is common to have to manage problems and challenging results in this experimental stage.

The strength of this tool lies in its ability to assess the advancement and potential of developing technologies by understanding the particular characteristics of each TRL stage.

### **3.2 Photogrammetry**

A strong method for extracting precise and accurate three-dimensional data from two-dimensional videos is photogrammetric analysis. It makes it possible to use computer models to estimate fragments movements and to recreate real-world coordinates. An in-depth comprehension of the motion is obtained through the camera calibration procedure, the object tracking stage, and 3D reconstruction.

A photogrammetric analysis of video data must go through several necessary procedures. First of all, it's essential to calibrate the cameras used for recording. The link between the camera's image plane and the physical coordinate system, as well as the intrinsic and extrinsic camera parameters, like focal length and lens distortion, are all established by camera calibration.

Moreover, a variety of methods can be used for object tracking, including feature-based tracking, point correspondence matching, and optical flow algorithms. To precisely estimate an object's position between two frames, these techniques examine how its position changes across the frames. Furthermore, the final step consists in the application of the triangulation method, able to relate camera characteristics (focal length, lens distortion) and coordinates from two-dimensional (2D) videos to get the real-world coordinates (3D coordinates) of the object in each time instant.

#### **3.2.1 High speed photography**

The primary issue with open-pit mining is the high cost of drilling and blasting, which greatly raises the total cost of mining (Afum & Temeng, 2015; Božić, 1998; Palangio et al., 2005). The effectiveness and expense of drilling, blasting, and following mining activities are directly impacted by the quality of the blasting (Bowa, 2015). One of the most challenging features to be faced when analysing blasts is its duration. Blast often lasts less than three seconds and for this reason it requires an adequate time foundation to observe and understand short duration occurrences, according to (Chiappetta & Mammele, 1987). Since events happen too quickly for the human eye to

discern, visual investigation is insufficient for studying blasting. The most popular technique for analysing the blasting process is to employ data from high-speed photography, slowing down the blast movement to allow for qualitative and quantitative analysis (Blair, 1960; Chiappetta & Mammele, 1987; Franco, 2022; Navarro et al., 2018; Onederra & Esen, 2003; Segarra, 2004).

The High-Speed cameras (Balch, 1999) are employed to record fast-moving objects as photographic images, with frame rates much higher than 250 fps. Usually, they are used for aerodynamic studies, where, through the sequences of images, a motion analysis can be carried out in either 2D or 3D (if more than one camera is used). The ability of high-speed cameras to record videos at extremely high frame rates enables the depiction of quick processes. In this way, it is possible to slow down the blast movement by employing high-speed cameras, making it possible to examine the explosion phenomenon in great detail. The fragmentation patterns, shockwaves, initial velocities, response times and trajectories of the blasting process can all be better understood using this method. The better understanding of blast dynamics, improved blast design, and improved overall efficiency and cost-effectiveness of mining operations may all be accomplished using this information.

It has been considered as the proper technique, for both qualitative and quantitative analysis of blasting operations.

The only downside of these cameras is the high cost, that is the reason why valid substitutes, such as High Frame Rate cameras, have been considered for the photogrammetric analysis.

The use of HFR technology has been used to capture the slow-motion sequences, providing enough detailed information about the trajectory of the rock fragments.

In conclusion, the estimation of fragment movements, recreating real-world coordinate, is achieved by using photogrammetric analysis, which appears to be a reliable and successful technique. For the purpose High Frame Rate cameras offer a practical and affordable alternative for obtaining trajectory information in slow-motion sequences, even though high-speed cameras are frequently employed for motion analysis.

### **3.3.1 From 2D to 3D analysis**

There can be difficulties and data loss while analysing a three-dimensional phenomenon from a two-dimensional viewpoint. When 2D assumptions, as for example the hypothesis on the motion of the fragments happen in a plane perpendicular to the bench, are used, real velocities and explosion directionality are underestimated. A 3D methodology built on stereophotogrammetric analysis is employed to get over this restriction. By integrating the spatial information from two photos

obtained from separate positions, this method reconstructs dynamic processes. This method needs precise knowledge of the camera's internal calibration and external orientation to reconstruct the real trajectory of moving objects (Moser & Gaich, 2007). The internal camera calibration refers to all the information belonging to the image formation process of the camera, the external camera orientation provides accurate information about the camera position and the angle of view while recording the videos.

The 2D analysis with high-speed cameras has its limitations. It only records the blast events in two dimensions, losing some spatial information. As a consequence, it is important to consider the limitations and potential errors that may result from the projection of a three-dimensional phenomenon onto a two-dimensional image while performing 2D analysis. On the other hand, three-dimensional approaches, such as stereophotogrammetric analysis, have been employed to overcome these limitations. With stereophotogrammetric analysis it is possible to reconstruct the three-dimensional movement of objects recorded in high-speed videos, to gain more precise and complete data.

At the end, three-dimensional approach offers a more thorough and accurate analysis of the blast events, enabling a better comprehension of the blast dynamics and optimization of blasting operation, without making wrong assumptions.

### **3.4 DIGIECO project**

The extraction of aggregates is a large industry in the EU, providing raw materials for the infrastructure and construction sectors. However, the industry must strengthen its long-term position and minimize its reliance on imports. The DIGIECO project underlines the significance of establishing internal value chains and securing trustworthy and environmentally friendly access to raw materials (RM).

The primary issues impacting the industry of aggregates are the following:

1. Health, safety, and security: the sector is habitually affected by accidents and serious injuries because it involves the use of heavy equipment and operate sin different dynamic environments. Upgraded safety precautions are required, as well as enhanced interaction among workers, equipment, and sites.
2. Efficiency, selectivity, and profitability: most of the industries are dominated by small and medium-sized companies and that is why adaptable solutions are needed to increase overall

efficiency. Nowadays the technologies employed in multi-sites quarries result in a lack of interconnection, providing issues.

3. Environmental impact: quarry activities can harm ecosystems by disrupting water flow, generating trash, and polluting the air and noise levels. Minimizing these effects and promoting sustainable activities are essential.

4. Social acceptance: social tensions, such as worries about traffic and the environment, can occur as quarries approach on densely inhabited areas. Most of the development of the industry depends on promoting social acceptance and clarity.

In this background, the DIGIECOQUARRY project aims to transform all the operations in fully automated and digitalized in order to address the issues facing the sector, such as social acceptance, environmental effect, efficiency, and profitability. The project forecasts the creation of an Innovative Quarrying System (IQS) that combines sensors, processes, tools, and data management to supply real-time process control.

The project's goals include the following:

1. Enhanced worker health and safety as a result of automated and managed operations.
2. Improved efficiency and selectivity in quarrying operations to rise profits and sustainability.
3. Promoting circular economic principles while minimizing emissions, enhancing water management, and maximizing sustainability and resource efficiency.
4. Greater social acceptance through interacting with stakeholders and dealing with community issues.
5. Validation of the IQS in five prototype/pilot scenarios to show that it is reliable and effective.
6. Improving public impressions and perceptions of the quarrying sector through knowledge sharing, international collaboration, and research contributions.

In four years, the project will establish six key technological areas (KTAs) and concentrate on two key social areas (KSAs) with a consortium of 25 partners from several disciplines. It intends to produce 38 exploitable outputs, including tools, services, and hardware and software solutions.

In the table below (Table 1), the KTAs and the KSAs are listed, together with the actions demanded in the 2050 roadmap for European raw materials<sup>1</sup> and the related main challenges.

---

<sup>1</sup> The European Union (EU) created a strategic document called the Vision and Roadmap for European Raw Materials in 2050 to lay out the long-term objectives and steps for the sustainable management of raw materials inside the EU. The plan focuses on finding sustainable and responsible solutions to problems with raw material availability, access, extraction, processing, and recycling (ETP SMR, 2016).

Table 3: Relationship between 2050 roadmap, KTAs, KSAs and challenges from (European Union, 2021)

Actions needed according to 2050 roadmap for RM	KTAs, KSAs	Challenges
<p>Action 1: Improvement of the excavation (extraction process) through innovative drilling &amp; blasting systems.</p> <p>Action 2: Developing new and adapting existing technologies and operations to increase automation in quarries to reduce energy consumption in muck pile digging, loading, and hauling for fossil-free production.</p>	<p>KTA1   Improved Extraction, Rock Mass Characterisation and Control</p>	<p>H&amp;S Security Economic efficiency Environmental Impact Social Acceptance</p>
<p>Action 3: Create innovative crushing, (selective) grinding, and sorting technologies to reduce energy use in mineral processing.</p>	<p>KTA2   Innovative Treatment Processes</p>	<p>H&amp;S Security Economic efficiency Environmental Impact Social Acceptance</p>
<p>Action 4: Develop and integrate industry solutions to improve process control for optimised primary and secondary processing. Fully control the material flows allowing on-demand customisation.</p>	<p>KTA3   Smart Sensors, Automation and Process Control</p>	<p>H&amp;S Security Economic efficiency Environmental Impact Social Acceptance</p>
<p>Action 5: Develop sophisticated Big Data and ICT-based production technologies that allow for optimised process efficiency and management throughout production (energy, resource, water). Advances in digitalisation and ICT have enabled a real-time integrated process control for quarrying.</p>	<p>KTA4   Quarry Full Digitalisation Through ICT Solutions, BIM &amp; AI</p>	<p>H&amp;S Security Economic efficiency Environmental Impact Social Acceptance</p>
<p>Action 6: Improve the working environment for operators and develop new systems for improving Health and Safety.</p>	<p>KTA5   Applications for Health and Safety (H&amp;S) Improvement</p>	<p>H&amp;S Security</p>
<p>Action 7: Minimise environmental impact of quarrying operation Data generation and management, including life-cycle assessment (LCA), environmental impact, and mass flow analysis.</p>	<p>KTA6   Applications for Environmental Impact Minimisation</p>	<p>Economic efficiency Environmental Impact</p>
<p>Action 8 - Develop knowledge on societal influence and acceptance including research on areas of conflict with regard to human health, ethics, gender, rural development and urban life, environmental, social, and economic aspects.</p>	<p>KSA7   Social Acceptance and Social Influence</p>	<p>Social Acceptance</p>
<p>Action 9: Communication with policy makers and public bodies and synergies with other initiatives will be key to improving processing.</p>	<p>KSA8   Communication with Policy Makers &amp; Public Bodies</p>	<p>H&amp;S Security Economic efficiency Environmental Impact Social Acceptance</p>

According to the diagram below (Figure 1), each of the proposed KTAs and KSAs includes many developments:

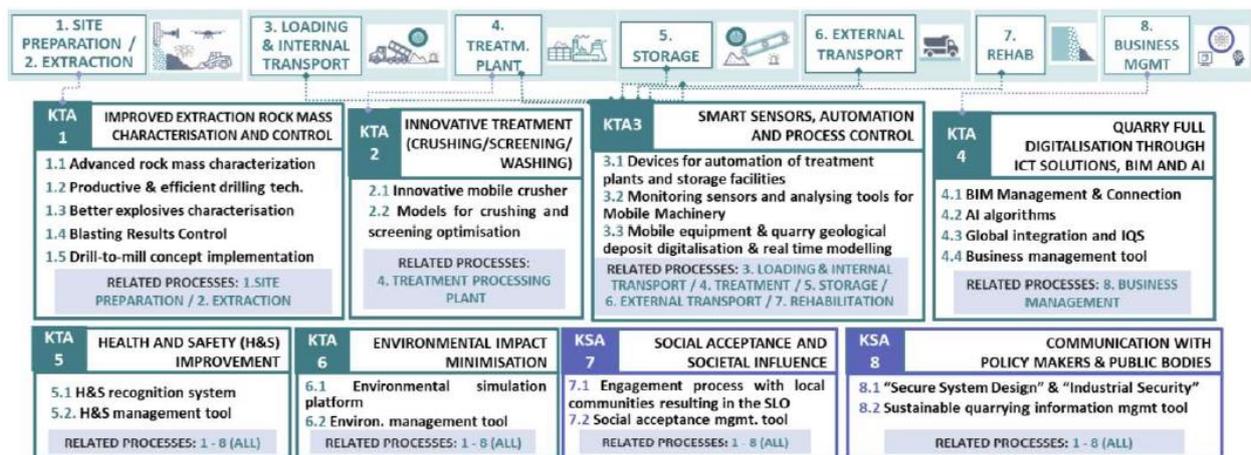


Figure 1: KTAs and KSAs

This work belongs to the KTA1: Improved extraction, rock mass characterisation and control and in particular to the section *1.4 Blasting Results Control*. This specific development aims to achieve significant progress in blasting technology to enhance blast outcomes and optimize numerous variables, including rock fragmentation, wall stability, vibrations, and overall operational efficiency. One of the most significant developments in this field is the post-blast field measurements, which are used to control blast performance and to validate and calibrate developed models (European Union, 2021). These metrics aid in determining the blast's actual results and offer useful information for improving blasting design. In particular, the development of multiple characterisation techniques (muck pile's movement vectors and fragmentation patterns) to create post-blast models in relation to rock fragmentation is getting an increasing interest in the field of blasting technology. It has always been difficult and frequently wrong to determine the grain size distribution of a muck pile. Through this development it would be feasible to estimate it more precisely by evaluating the post-blast models derived from photogrammetric methods.

The strategy tries to link the in-situ block size distribution of the rock mass with the characteristics of the muck pile. By making this connection, the new developed blasting design software is able to fully comprehend the attributes of the rock mass and incorporate that knowledge into the models, getting more accurate outcomes and having the possibility to continuously improve the models.

The importance of multiple characterisation techniques lays in their capacity to fully improve the blasting stage. A more accurate evaluation of the fragmentation affects all the subsequent procedures like crushing and grinding. Additionally, it aids in assessing the blasting design's effectiveness and pinpointing potential improvement areas.

Furthermore, knowing the muck pile's movement vectors offers information about the distribution of blasted debris and the following shape of the excavation. This knowledge is helpful for determining the stability of walls and guaranteeing the security of people and machinery nearby.

By including photogrammetry methods (based on post-blast models), movement vectors, and fragment size distribution data into the blasting design software, it is possible to achieve better results in terms of loading/digging cycles, lower energy consumption, decreased equipment maintenance costs, and improved haulage fleet efficiency.

For sectors like mining, these developments in rock classification and modelling hold considerable potential because they offer the possibility to improve blasting techniques in being more precise and effective, increasing the productivity and saving costs.

Overall, the DIGIECOQUARRY project aims to transform the aggregates sector by utilizing automation and digital technology to meet industry difficulties and support the EU's strategic objectives for raw materials and sustainability.

## **4. Methodology**

The methodology underlying the work is based on three main fields of investigation: the camera selection process, the photogrammetric analysis of the recorded videos and the three-dimensional reconstruction of the object's trajectory.

### **4.1 Camera selection process**

The camera selection process involves market research to identify the potential camera brands and models together with their main features that meet the required criteria.

Moreover, an analysis of blast face movements, based on the previous works on El Aljibe quarry, has been conducted. The goal of this step is to understand whether the cameras are able to capture moving object from a certain distance from the source and what can be the optimal placement of the cameras, influenced by the geometrical configuration of the site.

#### **4.1.1 Features of the cameras**

Some camera features that affect the camera selection process are the HFR availability, the frame rate, the possibility of being remotely triggered using a Wi-fi connection, the focal length, the resolution, the pixel size and so on. All these features are defined in the following part.

The frame rate is the speed (rate) in which consecutive images (frames) are acquired and captured by the camera (Brunner). It is expressed in frames per second (fps). If the frame rate is too low, images will appear jagged and jerked. The higher the frame rate is, the smoother and clearer the images will appear. If the purpose is to have slow motion effects, at least a camera with 120 fps is needed.

The High Frame Rate (HFR) technology allows the realization of slow-motion sequences by capturing more frames per second (Joseph, 2022).

A trigger system is a mechanism able to remotely activate the shutter on the camera (Hunter et al., 2022). The direct triggering would not be possible in our case for safety reasons and a remote control cannot be used because it does not work at distances higher than 80 – 100 m. That is why the cameras must be physically connected to two computers, which will be linked to a third device, via Wi-fi, through the TeamViewer software. The third device will send the signal to the other two devices, which will activate the cameras to start the recordings.

The focal length is the distance between the sensor of the camera and the point at which the camera can take the sharpest photograph of the subject, focusing the camera to infinity (Nikon Inc. et al., 2023). The focal length plays an important role in detecting any fragment. It gives information on how much of a scene a lens can capture, and how big objects will appear.

Another important parameter is the resolution. It is defined as the total amount of pixels that can be captured. The visual quality and the accuracy of the imagery-based classifications can both be affected by the resolution (Fryer & McIntosh, 2001).

The pixel size of a camera, usually measured in  $\mu\text{m}$ , refers to the physical dimensions (height and width) of the pixel unit on the camera sensor (Versus, 2016). This indication plays an important role when choosing a camera because it affects the amount of light caught by the pixels (sensitivity) and its ability to capture details. In case of low-light environments, a sensor with a larger pixel size is better because they will be able to capture more light, although the detection of fine details within the image can get worse.

The shooting time defines the longest continuous recording period that a camera can make before running out of space, battery life, shooting modes, frame rates, and other factors (Ascher & Pincus, 1999). The High Frame Rate setting has a significant impact on the shooting time because the camera will catch, gather, and process more frames per second when recording in High Frame Rate mode. This may result in shorter shooting times, quicker battery depletion, and higher memory usage.

Moreover, the memory cards needed should be compatible firstly with the camera and then we should consider the HFR records, that usually require more memory than a normal video. Capacity is one parameter to consider when choosing a memory card.

Generally, another important parameter to be considered is the battery life of a camera. In the case under exam, there are not specific requirements that have to be fulfilled because the average duration of a blast is about seconds.

Three different camera brands (Sony, 2016, 2020, 2022), (Fujifilm), and (Canon, 2019) have been analysed to determine which of them is better and suitable for our purpose. The initial screening is based on some camera features focused on HFR availability, followed by a filter based on frame rate to ensure clear and non-blurred images.

The Table below (Table 4) shows the different camera brands that have been analysed and the first filtering approach that was applied to the final choice.

*Table 4: Type of cameras and related HFR availability*

<b>Brand</b>	<b>Type</b>	<b>HFR Availability</b>
Sony	DSC-RX100 VII	Yes
	ZV-1	Yes
	ILME-FX30	No
	RX10 III	Yes
	Alpha 7S III	No
Fujifilm	X-T4	No
Canon	G5X Mark II	Yes

Despite good performances in terms of resolution, focal length, and triggering system, the Sony ILME-FX3, the Fujifilm XT 4, and the Sony Alpha 7S III cameras have not the High Frame Rate availability and for this reason they have been discarded. Whereas in the table above (Table 4), the list of possible cameras that fulfil the initial requirements are presented together with the main parameters that will affect the final choice.

In the following table (Table 5), all the cameras that have the HFR availability are listed. For these cameras, a deeper investigation has been made, considering their intrinsic features (provided by the suppliers) as the sensor size, the resolution, the focal length, the frame rate and HFR, the shooting time and the cost.

Table 5: Cameras features

Brand	Type	Features								
		Sensor		Resolution (Mpx)	Focal length (mm)	Frame Rate	HFR availability	HFR (fps)	Shooting time with HFR mode	Cost (€)
		Type	Size (mm)							
Sony	DSC-RX100 VII	1.0-type Exmor RS CMOS sensor	13.2 x 8.8	20.1	9.0 - 72	[60p 50M]/[50p 50M] [30p 50M]/[25p 50M] 24p 50M (NTSC)	YES	240/250 480/500 960/1000	Approx. 7s Approx. 6/7s Approx. 6s	1300
Sony	ZV-1	1.0-type Exmor RS CMOS sensor	13.2 x 8.8	20.1	9.4 - 25.7	[60p 50M]/[50p 50M] [30p 50M]/[25p 50M] 24p 50M (NTSC)	YES	240/250 480/500 960/1000	Approx. 3s Approx. 3s Approx. 3s	800
Sony	RX 10 III	1.0-type Exmor RS CMOS sensor	13.2 x 8.8	20.1 Effect.	8.8 - 220	[60p 50M]/[50p 50M] [30p 50M]/[25p 50M] 24p 50M (NTSC)	YES	240/250 480/500 960/1000	Approx. 4s Approx. 4s Approx. 4s	1600
Canon	G5X Mark III	1.0-type Exmor RS CMOS sensor	13.2 x 8.8	20.1	8.8 - 44	119.9 / 100 / 59.94 / 50 / 29.97 / 25	YES	119.9 / 1000.0	Up to 7 min. 29s	930

According to Table 5, although the camera from Canon has the highest shooting time duration, it has been discarded because it has the lowest frame rate value in HFR mode. This parameter represents the reason why a second screening has been done for the cameras listed because, as previously stated, high frame rate values in HFR mode is necessary to have a record with good quality.

Moreover, all the cameras listed have the possibility to be remotely triggered via Wi-fi and have a battery capacity that can be capable of withstanding the average duration of a record for our purpose.

#### 4.1.2 Motion analysis of the blast face

As one of the objectives is to evaluate the motion of the blast face, developing a methodology, the ballistic trajectory of the fragments has been determined using differential equations in order to get the final position of the fragments in the ground and their distance from the bench. It has been possible to determine the position of each fragment in time and space by employing the velocity and the acceleration data of the fragments after being blasted, coming from the blasts hold in El Aljibe Quarry (Franco, 2022; Jiménez, 2022), together with the geometrical parameters of the Valdilecha site. This motion analysis has been done to know and evaluate the minimum distance at which the cameras can be placed, based on statistical models, and using the information coming from previous studies (Jiménez, 2022).

The site under investigation is the Valdilecha Quarry, near the municipality of Valdilecha (Madrid). Limestone is extracted from the quarry and sold to cement factories to produce concrete (Hanson Hispania, 2020). The figure below (Figure 2) shows the map of the site, highlighting the benches under exploitation in the quarry.

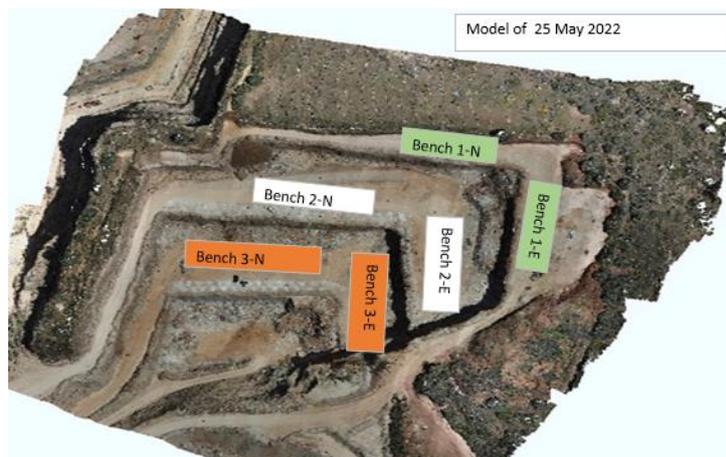


Figure 2: Location of the benches under exploitation in Valdilecha Quarry

The geometrical parameters employed to set up a methodology refer to the Bench 2-E and in particular to the blast performed on 1<sup>st</sup> July 2022, as reported in Table 5.

Table 5: Geometrical parameters of the bench

<b>Bench 2-E - Blast from 01/07/22</b>		
<b>Parameters</b>		<b>Units</b>
Number of holes	24	-
Number of rows	2	-
Number of holes per row	12	-
Height of the bench	14.2	m
Length of the bench	84	m
Burden	5.75	m
Spacing	7	m
Stemming	3	m
Inter-row delay	25	ms

A scheme of the blast (Figure 3) has been drawn to have a schematical representation of the bench. For simplicity, only one row of holes is represented and the blastholes are assumed to be vertical even if in reality they are not.

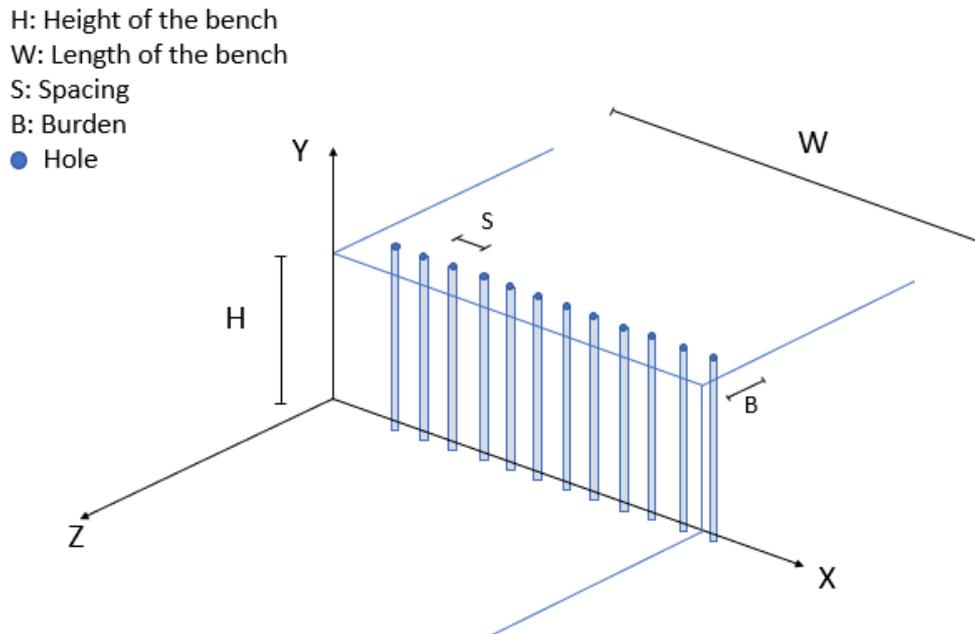


Figure 3: Simplified representation of the bench

The final position of the fragments on the ground has been estimated for 5000 fragments ( $i = 1, \dots, 5000$ ) coming from one row of with twelve blastholes ( $j = 1, \dots, 12$ ).

Moreover, a grain size distribution of the muck pile is necessary because the mass of the fragment can affect its velocity and acceleration and consequently the distance at which it is projected. Due to the lack of information about the fragment size in the site under investigation, the grain size of another limestone quarry has been considered. The Figure 4 below shows the grain size distribution of the Goltas Limestone Quarry in Turkey (Ozkahraman, 2005).

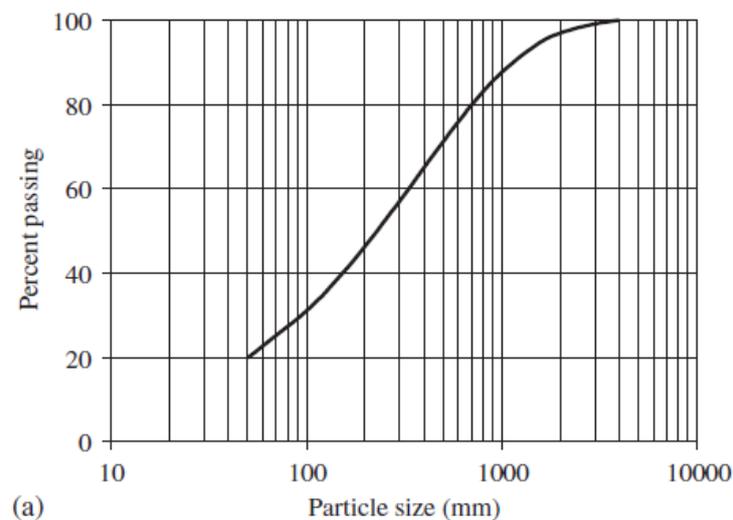


Figure 4: Grain size distribution of Goltas Limestone Quarry, Turkey

It is reasonable to assume that this grain size distribution can be representative of the grain size distribution of the Valdilecha quarry, because in both quarries limestone is mined (Hanson Hispania, 2020).

It is important to outline that the curve has been truncated at 20% of passing to not randomly sample fragments that are smaller than 5 cm. If the fragments are too small, they may be confused with each other, hidden by the dust coming from the blasts, and the detection of their trajectory can be challenging.

For the sake of simplicity, the rock fragments have been considered as spherical particles.

The analysis of the movement of the blast face is based on the equations of motion used to determine the velocity and the acceleration of each fragment after being blasted, its final position on the ground, its distance from the bench, the time at which it impacts the ground and the duration of the blast, as proposed by (Jiménez, 2022).

To get the trajectory of each particle, a detailed analysis of all the forces that are acting at the centre of gravity of each fragment during the whole flight, is required (Segarra, 2004; Stojadinović et al., 2011).

Rock fragments' kinetic behaviour during blasting has been discussed by (Segarra, 2004; Stojadinović et al., 2011). It is required to understand all of the forces acting on each rock's centre of gravity throughout its entire flight in order to apply the second Newton equation of motion and determine the trajectory of each particle, as shown in Figure 5. The study is equally valid for all the fragments of the bench.

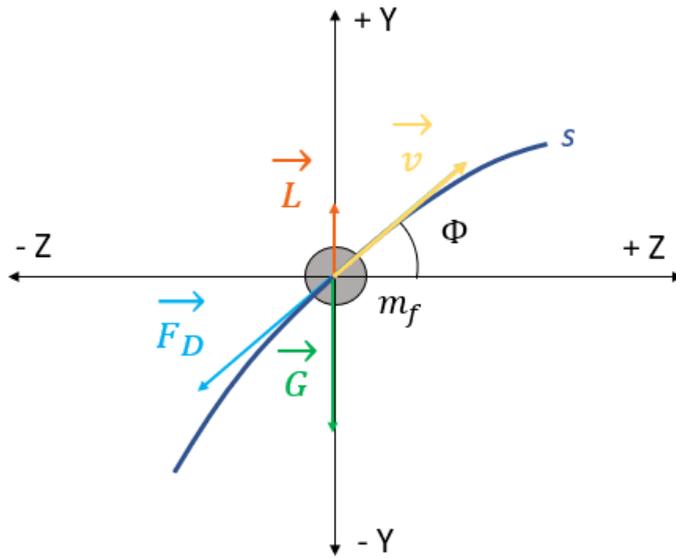


Figure 5: Basic forces acting upon the flyrock fragment. Adapted figure from (Stojadinović et al., 2011).

These forces are the weight (**G**), the drag force (**F<sub>D</sub>**) and the lift force (**L<sub>f</sub>**):

$$\mathbf{G} = m_f \mathbf{g} \quad \text{Eq. (1)}$$

$$\mathbf{F}_D = \frac{1}{2} \cdot C_D \cdot \rho_a \cdot S_r \cdot V^2 \quad \text{Eq. (2)}$$

$$\mathbf{L}_f = \frac{1}{2} \cdot C_D \cdot \rho_a \cdot S_r \cdot V^2 \quad \text{Eq. (3)}$$

Where:

- $m_f$  is the mass of a fragment, which depends on the diameter and on the rock density (kg).
- $g$  is the acceleration of gravity ( $\text{m/s}^2$ ).
- $C_D$  is the aerodynamic coefficient.
- $\rho_a$  is the air density ( $\text{kg/m}^3$ ).
- $S_r$  is the cross section of the fragment ( $\text{m}^2$ ).
- $V$  is the velocity of the fragment ( $\text{m/s}$ ).

Generally, other forces such as the influence of the wind and the Coriolis effect play an important role in the analysis of the ballistic movements, but in this case, they have been considered as negligible together with the lift force (Stojadinović et al., 2011).

The following equations are known for each spherical particle and for each instant.

$$m_f = \frac{4\pi}{3} \cdot \rho \cdot \left(\frac{\varnothing}{2}\right)^3 \quad \text{Eq. (4)}$$

$$S_r = \pi \cdot \left(\frac{\varnothing}{2}\right)^2 \quad \text{Eq. (5)}$$

$$C_D = \frac{24}{Re} + \frac{2,6 \cdot \left(\frac{Re}{5}\right)}{1 + \left(\frac{Re}{5}\right)^{1,52}} + \frac{0,411 \cdot \left(\frac{Re}{263000}\right)^{-7,94}}{1 + \left(\frac{Re}{263000}\right)^{-8}} + \frac{Re^{0,8}}{461000} \quad \text{Eq. (6)}$$

$$Re = \frac{\rho_a V \varnothing}{\mu} \quad \text{Eq. (7)}$$

Where:

- $Re$  is the Reynolds number, function of the air density, the dynamic air viscosity, the velocity, and the diameter of the fragment.

Table 6 shows the physical constants which have been used in the model.

Table 6: Physical constants

Parameter	Value
Acceleration of gravity ( $g$ )	9.8 m/s <sup>2</sup>
Fragment density ( $\rho$ )	2721 kg/m <sup>3</sup>
Air density ( $\rho_a$ )	1.1614 kg/m <sup>3</sup>
Dynamic air viscosity ( $\mu$ )	1.8 · 10 <sup>-5</sup> kg · m <sup>-1</sup> · s <sup>-1</sup>

A three-dimensional representation of the bench and the trajectory of a single fragment ( $i$ ) are shown in Figure 6. Moreover, two holes have been represented to understand which is the nearest blasthole to the fragment. The individuation of the nearest blasthole determines the ejected fragment direction, which is identified by the line passing through the position of the fragment and the nearest blasthole, also known as azimuth angle.

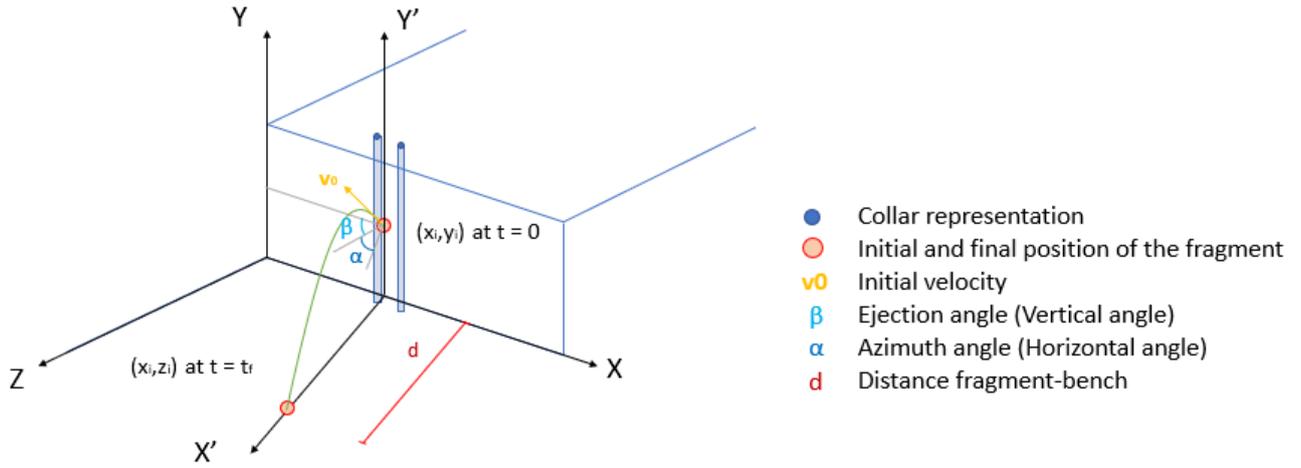


Figure 6: Simplified 3D representation of the analysis of a fragment's trajectory

The initial and final positions of the fragment have been represented by a circle and its trajectory is shown in green.

When the blast is fired ( $t = 0$ ), the fragment is part of the rock mass, still belonging to the bench and its initial position is defined by  $(x_i, y_i, 0)$  coordinates. Once the blast progresses, the fragment is ejected with a certain angle ( $\beta$ ), called ejection angle, and a given initial velocity ( $v_0$ ). This velocity is known as initial velocity and is defined as the speed at which each fragment is ejected. It is evaluated according to empirical models (Jiménez, 2022) obtained from the blast fired at El Aljibe quarry.

The ejected fragment follows a ballistic trajectory before hitting the ground, reaching its final position given by  $(x_i, 0, z_i)$  coordinates at  $t = t_f$ , when hitting the ground.

To understand which hole can be considered as the source of the fragment, it is important to determine the distance between the fragment ( $i$ ) and its closest hole ( $j$ ), defined as the horizontal distance. This relation is given by the following equation:

$$d_h = \min(x_{fi} - x_{hj}) \quad \text{Eq. (8)}$$

$$i = 1, \dots, 5000 \quad \text{Eq. (9)}$$

$$j = 1, \dots, 12 \quad \text{Eq. (10)}$$

Where:

- $d_h$  is the horizontal distance between a fragment and the closest hole.
- $i$  is the index of the fragments.
- $j$  is the index of the holes.
- $x_{fi}$  is the x-coordinate of the i-th fragment in the XY plane.
- $x_{hj}$  is the x-coordinate of the j-th hole in the XY plane.

To get the position of the fragment in the XZ plane, a change of coordinates is necessary.

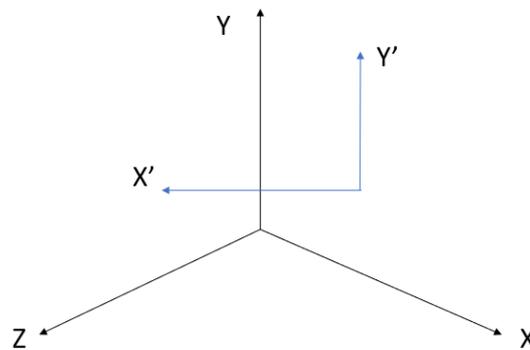


Figure 7: Coordinate system

As represented in Figure 8, a geometrical relationship exists between the azimuth angle ( $\alpha$ ), the burden ( $B$ ), and the distance hole-fragment ( $d_h$ ), that is described by the following equations (11-13).

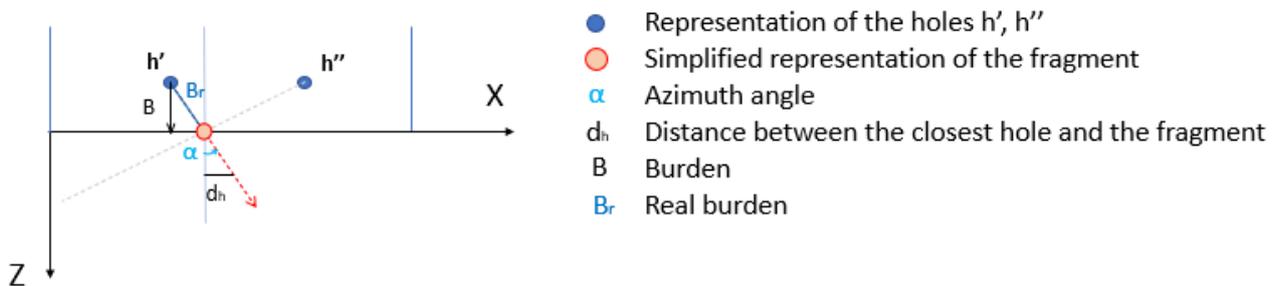


Figure 8: 2D representation in XZ plane

$$\tan(\alpha) = \frac{B}{d_h} \quad \text{Eq. (11)}$$

$$\alpha = \text{atan}\left(\frac{B}{d_h}\right) \quad \text{Eq. (12)}$$

$$B_r = \sqrt{B^2 + d_h^2} \quad \text{Eq. (13)}$$

Where:

- $\alpha$  is the azimuth angle ( $^\circ$ ).
- $B$  is the burden (m).
- $B_r$  is the real burden, the minimised distance between the hole and the free face (m).

Moreover, the trajectory of a single fragment ( $i$ ), its initial and final positions are shown with two-dimensional simplified representations in the three different planes XY and ZY.

The following figure (Figure 9) outlines the initial position (at  $t = 0$ ) of a rock fragment ( $i$ ) in the XY plane.

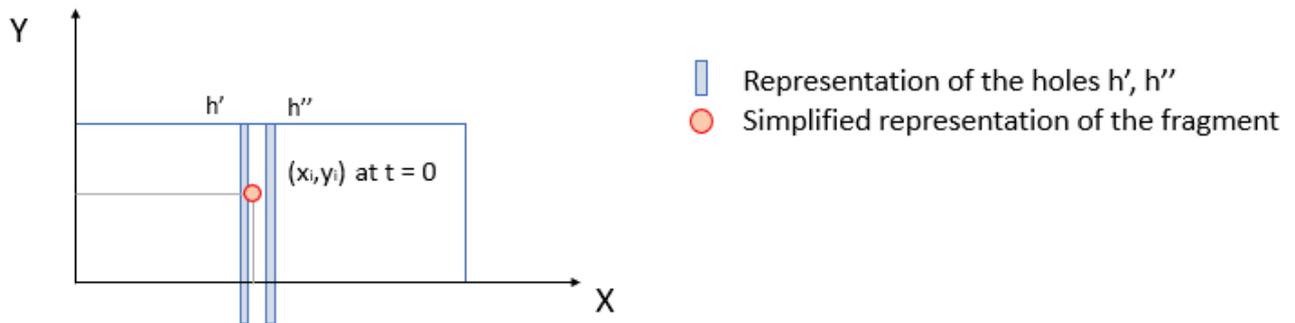


Figure 9: 2D representation of the initial position of the fragment in XY plane

Furthermore, the movement of the fragment has been analysed in the plane X'Y', as shown in the following figure (Figure 10).

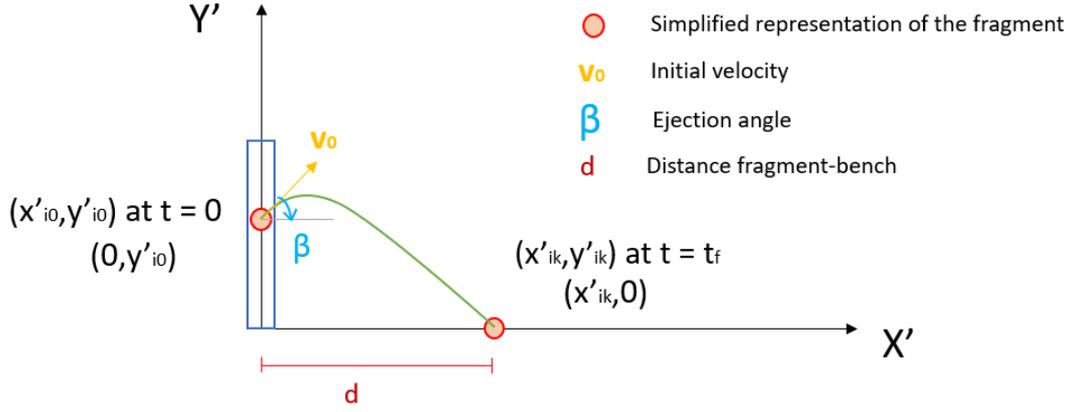


Figure 10: 2D representation of fragment's trajectory in  $X'Y'$  plane

The position of the rock fragment is given by the  $x'y'$  coordinates. At  $t = 0$ , its initial position is set by  $(x'_i = 0, y'_{i0})$ . Once the blast occurs, the fragment is ejected with an angle  $\beta$  and an initial velocity ( $v_0$ , evaluated according to the models coming from El Aljibe quarry). The fragment reaches its final position at  $(x'_i, y'_i = 0)$ . Note that in Figure 8 the fragment reaches the ground ( $y'_i = 0$ ) at a certain distance from the bench ( $d$ ).

Once the initial velocity ( $v_0$ ) has been evaluated, it is possible to obtain the final position of the rock fragments, through the following equations of motion. It is an iterative process computed for each fragment and for each time.

$$\dot{\mathbf{x}}'_{k+1} = \dot{\mathbf{x}}'_k + \ddot{\mathbf{x}}'_k \cdot (t_{k+1} - t_k) \quad \text{Eq. (14)}$$

$$\dot{\mathbf{y}}'_{k+1} = \dot{\mathbf{y}}'_k + \ddot{\mathbf{y}}'_k \cdot (t_{k+1} - t_k) \quad \text{Eq. (15)}$$

$$v_k = \sqrt{\dot{\mathbf{x}}'^2_k + \dot{\mathbf{y}}'^2_k} \quad \text{Eq. (16)}$$

$$\beta = \arctg\left(\frac{\dot{y}'_0}{\dot{x}'_0}\right) \quad \text{Eq. (17)}$$

Where  $\dot{\mathbf{x}}'_0$  and  $\dot{\mathbf{y}}'_0$  are the components of the initial velocity ( $v_0$ ) in the projection plane  $x'y'$  and  $\beta$  is the ejection angle.

$$\dot{\mathbf{x}}'_{k+1} = -\frac{1}{2} \cdot \frac{\rho_a \cdot C_D \cdot S_r}{m_f} \cdot \dot{\mathbf{x}}'_k \cdot \sqrt{\dot{\mathbf{x}}'^2_k + \dot{\mathbf{y}}'^2_k} \quad \text{Eq. (18)}$$

$$\dot{\mathbf{y}}'_{k+1} = -g - \frac{1}{2} \cdot \frac{\rho_a \cdot C_D \cdot S_r}{m_f} \cdot \dot{\mathbf{y}}'_k \cdot \sqrt{\dot{\mathbf{x}}'^2_k + \dot{\mathbf{y}}'^2_k} \quad \text{Eq. (19)}$$

$$\mathbf{x}'_{k+1} = \mathbf{x}'_k + \dot{\mathbf{x}}'_k \cdot (t_{k+1} - t_k) \quad \text{Eq. (20)}$$

$$\mathbf{y}'_{k+1} = \mathbf{y}'_k + \dot{\mathbf{y}}'_k \cdot (t_{k+1} - t_k) \quad \text{Eq. (21)}$$

The solution is a set of  $(x', y')$  positions defining the flight at discrete time intervals. Moreover, the Reynolds number, the aerodynamic coefficient, and the drag force have been computed for each time instant.

As shown in Figure 7, the motion of the fragment is first analysed in the X'Y' reference system. Then, to return to the original coordinate reference system (Figure 6), the following equations have been used:

$$x(i) = x_i^0 + x'(i) \cdot \cos(\alpha(i)) \quad \text{Eq. (22)}$$

$$y(i) = y_i^0 + y'(i) \quad \text{Eq. (23)}$$

$$z(i) = z_i^0 + y'(i) \cdot \sin(\alpha(i)) \quad \text{Eq. (24)}$$

Where:

- $x_i^0, y_i^0, z_i^0$  are the initial coordinate of the  $i$ -fragment.
- $\alpha$  is the azimuth angle, defined by the Eq. (12).
- $x'(i), y'(i)$  are the coordinates of the  $i$ -th fragment in the plane X'Y'.
- $x(i), y(i), z(i)$  are the new coordinates of the  $i$ -th fragment in the XYZ reference system.

With this change of coordinates, it has been possible to represent all the whatever fragments on the ground, defined by XZ plane, as shown in Figure 11.

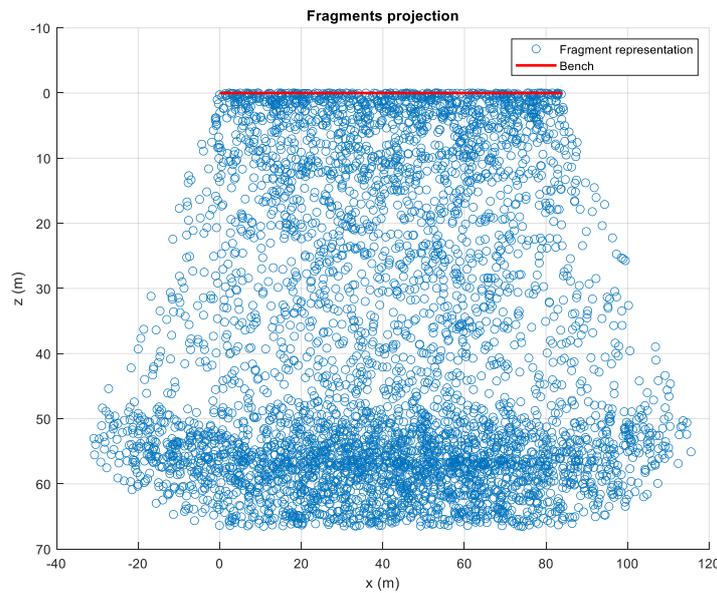


Figure 11: Fragments projection

Figure 12 shows the trajectories of ten fragments chosen arbitrarily to better visualize their paths.

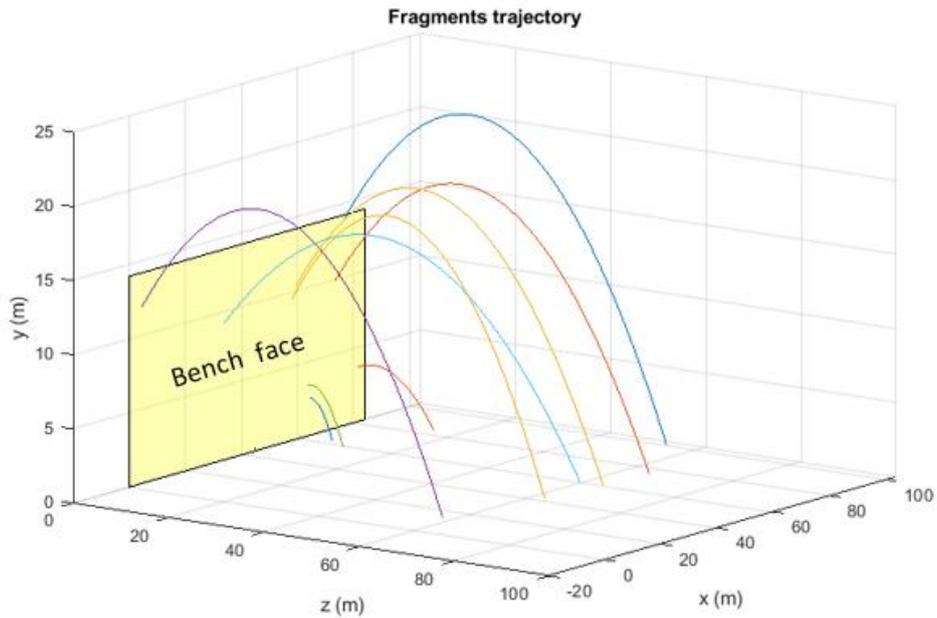


Figure 12: Fragments' trajectories

The figure below (Figure 13) shows all the fragments on the ground in a colour scale based on the nearest blasthole. The colour scale is subdivided into twelve different groups and each group represents one of the twelve holes. In this way, the fragments originated by different holes are highlighted.

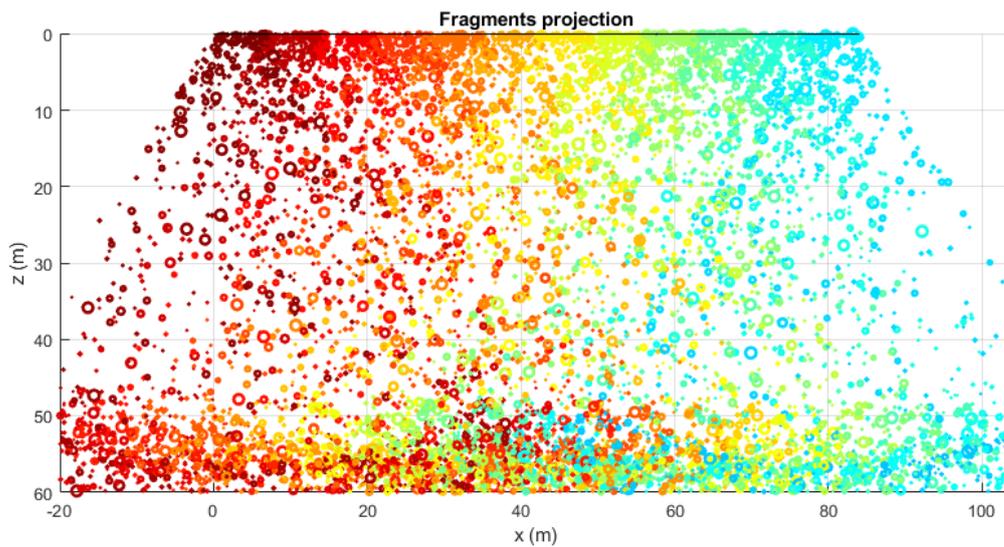


Figure 13: Coloured representation of fragments on the ground

The statistical tool of the empirical distribution function is used to understand which is the distribution of the distances travelled by the fragments.

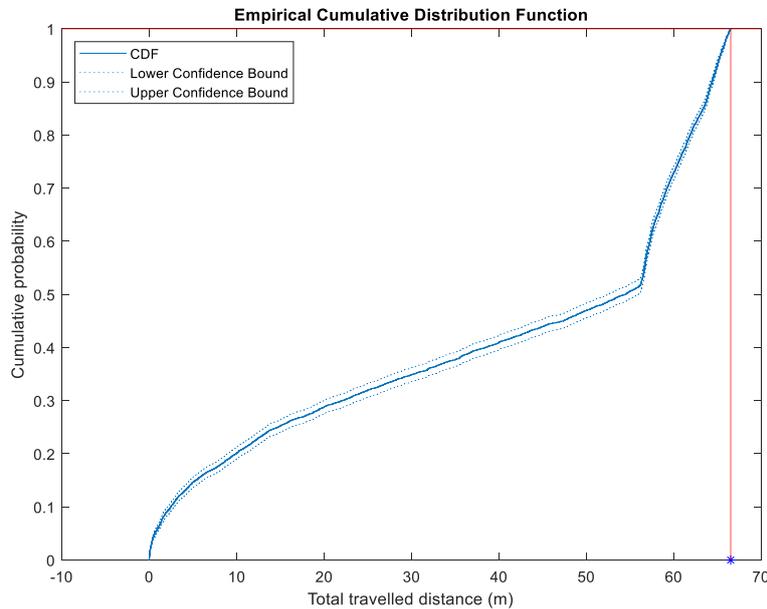


Figure 14: Empirical Cumulative Distribution Function

The farthest distance travelled by the fragments is 66.50 m as shown in the Figure 14. The importance of this parameter is not to be underestimated because it represents the minimum distance at which the cameras will be placed to not be hit by the ejected fragments. For safety reasons the distance between the cameras and the bench must be higher than 66.50 m.

### 4.1.3 Optimal location of the cameras

Once the minimum distance bench-fragment has been determined, it is necessary to optimize the position of the two cameras trying to maximize the number of fragments that are visible from both cameras and optimizing the coverage of a ground-based region of interest. Together with these parameters, also the optimal distance between the cameras and the bench and the distance between the two cameras are determined.

The cameras are assumed to have a fixed range and field of view ( $\delta$ ), and they are positioned at fixed heights above the ground. The position of each camera is described by the distance between it and the centre of the bench and by two angles: the angle  $\vartheta$  that defines the position of the first

camera and the angle  $\omega$  which describes the orientation of the camera in relation to the horizontal plane. As the cameras are two, two different angles are used to describe their directions:  $\omega_1$  for the first camera and  $\omega_2$  for the second one, respectively. Moreover, the distance between the two cameras needs to be evaluated.

The optimal location of the cameras has been acquired by firstly finding the coordinates of the two cameras and the boundaries of their fields of view. Then, the intersection of the two fields of view, which represents the area on the ground that is visible from both cameras, has been evaluated. This intersection has been used to calculate the overlap percentage between the two fields of view, as well as the number of points that are in the intersection area. Finally, it has been determined the existing relation between the farthest fragment and the cameras and the distance between one camera and the other. The distance between the two cameras strongly depends on the position of the farthest ejected fragment on the ground. This relationship has been described by a function, limited by a defined range of values (Adil et al., 2022).

The following figures (Figure 15 and Figure 16) show arbitrary situations of the cameras, highlighting the most important parameters that affect their optimal placement.

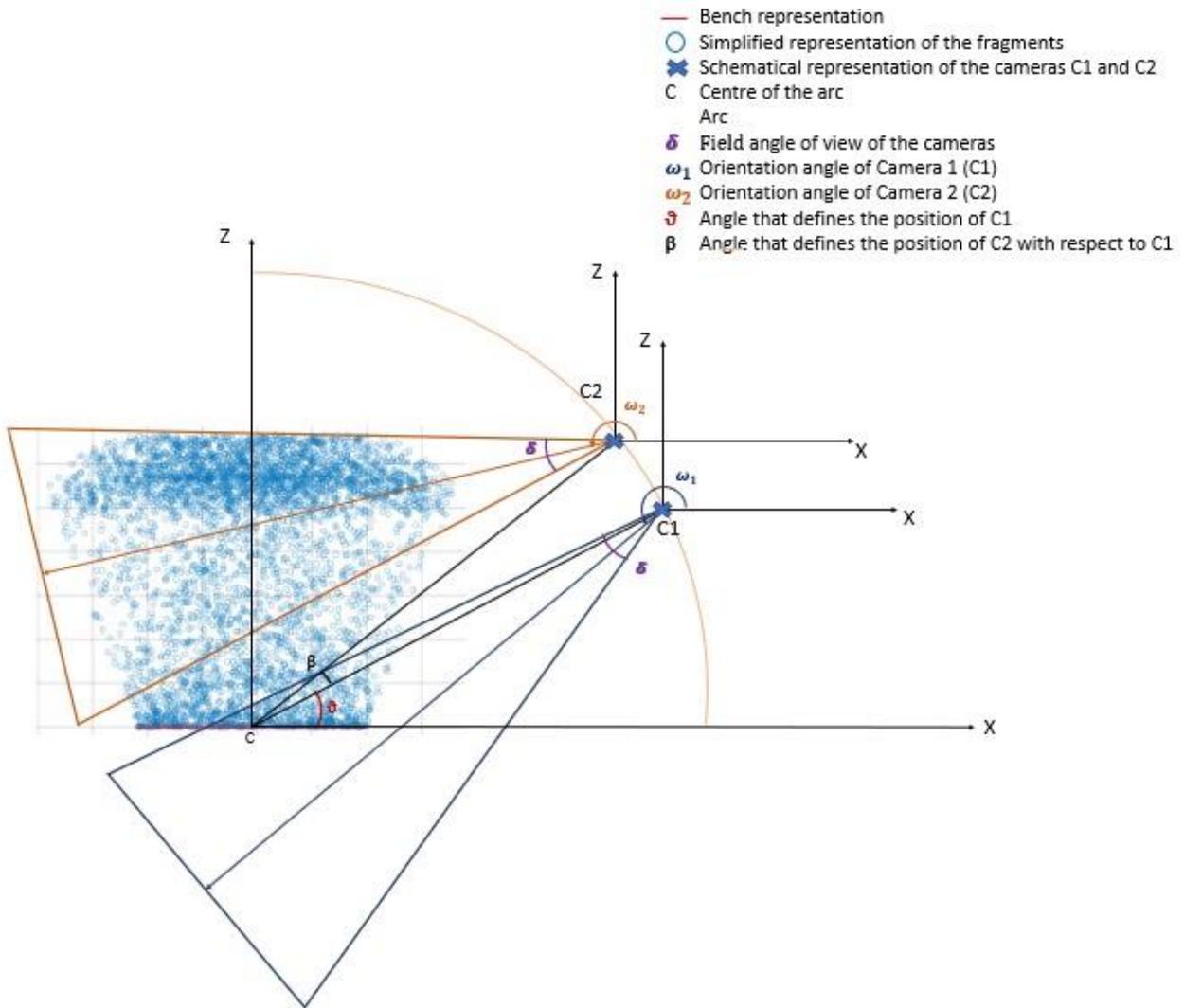


Figure 15: Schematical representation of a possible placement of the cameras with their FOVs

The position of the Camera 1 ( $C1$ ) can change along the arc according to the angle  $\vartheta$ . The arc has a radius equal to the value of  $r$ , that should be at least 66.5 m, according to Figure 13. For simplicity,  $\vartheta$  ranges from 0 to  $\frac{\pi}{2}$  because the analysis is assumed to be symmetrical. The position of Camera 2 ( $C2$ ) depends on the position of the first camera ( $C1$ ), and it is described by the angle  $(\beta + \vartheta)$ . Moreover, the distance between the two cameras ( $D$ ) can vary between 1 and 50 m. In the configuration shown in Figure 14, it is evident that the position and the orientation of the cameras is not describing an optimal condition because there is no intersection between the fields of view of the two cameras, represented by the triangles. This means that a three-dimensional reconstruction of the trajectory of the projected fragments is not possible because the fragments on the ground (represented by blue circles) will not be detected by both cameras at the same instant. This explains

why the goal of the analysis is to find the optimal location of C1 and C2, defined by  $\theta$ ,  $\omega_1$  and  $\omega_2$ , which must guarantee the intersection of the two fields of view of the cameras. The optimal configuration is reached when the fields of view are intersecting each other, and the resulting overlapping area is the greatest possible. At the same time, this overlapping area should cover the greatest number of fragments.

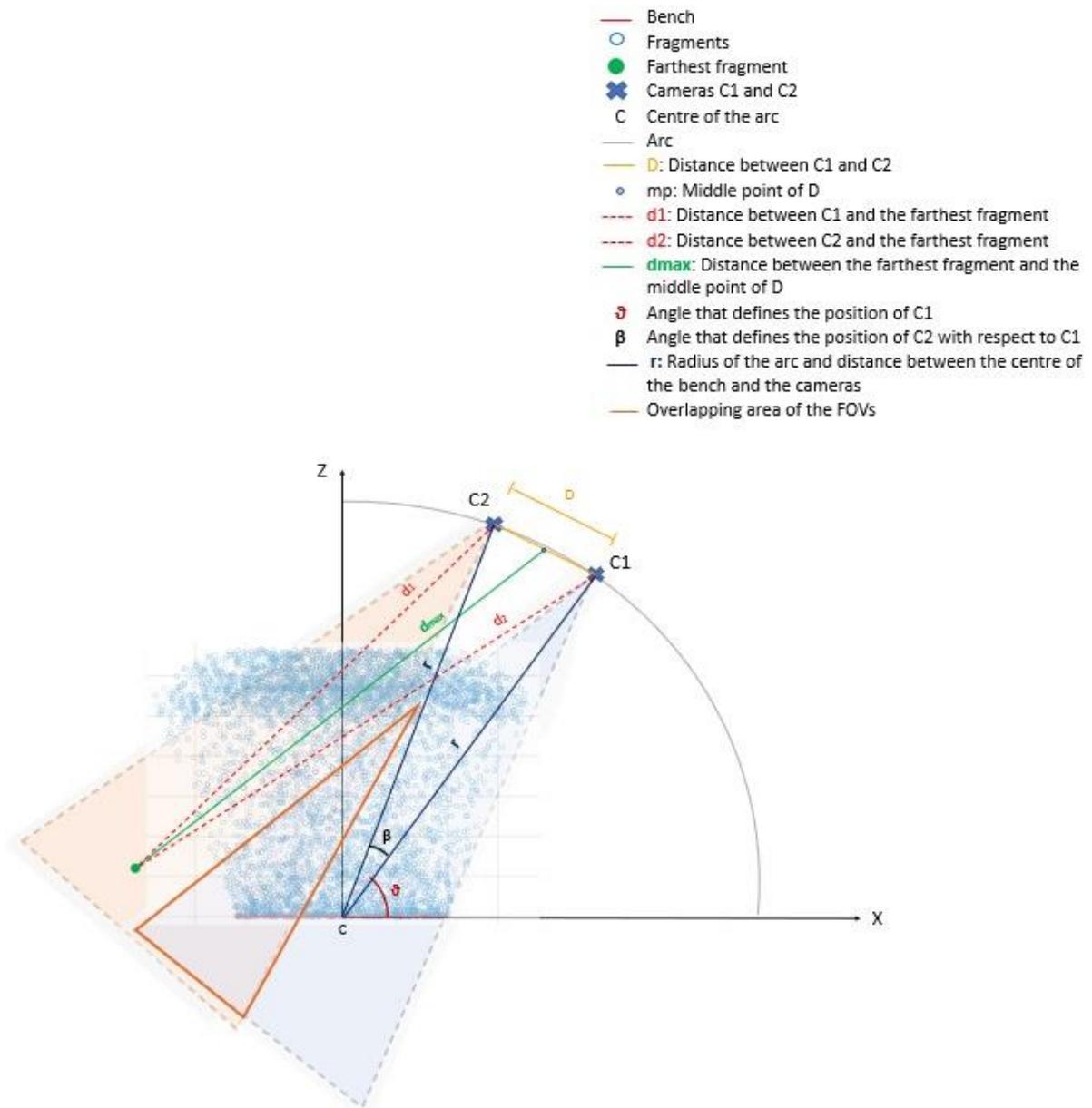


Figure 16: Schematical representation of a possible placement of the cameras, their distances from the fragments

In Figure 16, the importance of the distance between the cameras and fragments is highlighted. Among all the distance values there is one representing the maximum value that determine/define the position of the farthest fragment from the cameras. The identification of the farthest point is needed for our analysis because the cameras must be placed in a position where they can detect even the most distant fragments ( $d_{max}$ ). Moreover, according to (Over et al., 2021), it exists a relation between  $D$  and  $d_{max}$ : their ratio must be between 2 and 7. This means that the distance between the two cameras depends on how far the farthest fragment is. The evaluation of  $d_{max}$  is done by considering the maximum distance between the middle point of the chord defined between C1 and C2 (represented by a segment that links the two cameras) and all the fragments on the ground.

As stated before, the position of the cameras is described by several angles and is defined as follows:

$$C_1 = [C_{1x} \quad C_{1z}] = [r \cdot \cos(\vartheta) + C_x \quad r \cdot \sin(\vartheta) + C_z] \quad \text{Eq. (26)}$$

$$C_2 = [C_{2x} \quad C_{2z}] = [r \cdot \cos(\vartheta + \beta) + C_x \quad r \cdot \sin(\vartheta + \beta) + C_z] \quad \text{Eq. (27)}$$

$$\beta = 2 \cdot \text{asin}\left(\frac{D}{2 \cdot r}\right) \quad \text{Eq. (28)}$$

Where:

- $C_1, C_2$  are the coordinates of the cameras.
- $C_{1x}, C_{2x}$  are the x-coordinates that define the position of the two cameras.
- $C_{1z}, C_{2z}$  are the z-coordinates that define the position of the two cameras.
- $C_x, C_z$  are the coordinates of the centre of the arc. For simplicity, the centre of the arc is in the middle of the bench, defined by a length of  $W$ . So, in particular  $C_x = \frac{W}{2}$  and  $C_z = 0$ .
- $r$  is the radius of the arc, that represents the distance between the cameras and the centre of the bench.
- $\vartheta$  is the angle that describes the position of the first camera. It ranges from 0 to  $\frac{\pi}{2}$ .
- $\beta$  is the angle that together with  $\vartheta$  defines the position of the second camera.
- $D$  is the distance between the two cameras.

Two sets of vertices have been used to create two polygons (triangles), one for each field of view, to represent the FOVs of the cameras on the ground.

$$T_1 = \begin{bmatrix} C_{1x} + R \cdot \cos\left(\omega_1 - \frac{\delta}{2}\right) & C_{1z} + R \cdot \cos\left(\omega_1 - \frac{\delta}{2}\right) \\ C_{1x} + R \cdot \cos\left(\omega_1 + \frac{\delta}{2}\right) & C_{1z} + R \cdot \cos\left(\omega_1 + \frac{\delta}{2}\right) \\ C_{1x} & C_{1z} \end{bmatrix} \quad \text{Eq. (29)}$$

$$T_2 = \begin{bmatrix} C_{2x} + R \cdot \cos\left(\omega_2 - \frac{\delta}{2}\right) & C_{2z} + R \cdot \cos\left(\omega_2 - \frac{\delta}{2}\right) \\ C_{2x} + R \cdot \cos\left(\omega_2 + \frac{\delta}{2}\right) & C_{2z} + R \cdot \cos\left(\omega_2 + \frac{\delta}{2}\right) \\ C_{2x} & C_{2z} \end{bmatrix} \quad \text{Eq. (30)}$$

Where:

- $T_1, T_2$  are two 3-by-2 matrixes that define the vertices of the fields of view of the two cameras. The elements in the first column are the x-coordinates and the ones in the second column are the z-coordinates.
- $\delta$  is the field of view of the cameras and it is fixed at  $\delta = 72^\circ$ , according to the cameras features.
- $\omega_1, \omega_2$  are the orientations of the two cameras on the horizontal plane. They can range from 0 to  $2\pi$ .
- $R$  is length of the sides of the triangles (m).

Moreover, the areas ( $A_1, A_2$ ) of the two triangles and their intersection have been evaluated.  $A_1, A_2$  represent the areas on the ground that are visible from each camera. The two triangles (fields of view) are equal because they are defined by the same angle  $\delta$  and arbitrary sides ( $R = 300 \text{ m}$ ), which means that the areas of the two polygons are the same ( $A_1 = A_2$ ).

Then, the intersection of the two polygons is calculated, together with the area of the resulting intersecting polygon ( $A_{int}$ ).  $A_{int}$  outlines the area on the ground, visible from both the cameras. The intersecting polygon is defined by  $n$ -vertices and its area is used to calculate the percentage of overlap between the two fields of view.

$$Int_{polygon} = T_1 \cap T_2 \quad \text{Eq. (31)}$$

$$\% \text{ of overlap} = \frac{A_{int}}{A_1} \cdot 100 \quad \text{Eq. (32)}$$

Then, the built-in MATLAB function called *inhull* (D'Errico, 2006) has been employed to determine whether the fragments are within the intersection polygon. Therefore, the greater the intersection polygon is, the greater number of fragments would lie inside that area. Only the fragments that lie inside the intersection polygon have been considered, because they can be simultaneously visible from both the cameras.

Once the coordinates of the cameras have been determined, the middle point of the chord  $D$  is defined as follows:

$$MP = [MP_x \quad MP_z] = \left[ \frac{C_{1x}+C_{2x}}{2} \quad \frac{C_{1z}+C_{2z}}{2} \right] \quad \text{Eq. (33)}$$

Then, the distance between the between the middle point and all the fragments inside the intersection polygon has been evaluated according to the following equation (Eq. (34)).

$$d_{MP-f} = \sqrt{(MP_x(i) - x(i))^2 + (MP_z(i) - z(i))^2} \quad \text{Eq. (34)}$$

Where:

- $d_{MP-f}$  is a column vector, whose size depends on the number of points inside the intersection polygon. The elements of the vector are distance values between the middle  $MP$  and each fragment.
- $x(i), z(i)$  are the coordinates of the  $i$ -th fragment within the intersection polygon.

One of these distance values represents the maximum distance that defines the location of the farthest fragment from the cameras. It may be calculated by:

$$d_{max} = \max(\sqrt{(MP_x(i) - x(i))^2 + (MP_z(i) - z(i))^2}) \quad \text{Eq. (35)}$$

Where:

- $d_{max}$  is the maximum value of distance between  $MP$  and a fragment  $(x(i), z(i))$ .

The existing relationship between  $d_{max}$  and the chord  $D$  can be defined as the ratio (Over et al., 2021):

$$a = \frac{d_{max}}{D} = 2 - 7 \quad \text{Eq. (36)}$$

The following equation describes the function needed for keeping the parameter  $a$  within a specified interval. This parameter outlines the relationship that exists between the spacing between the two cameras and the distance of the fragments from the cameras.

$$y = \begin{cases} 0 & a < 2 \\ -a + 2 & a \geq 2 \text{ \& } a < 4 \\ -2 & a \geq 4 \text{ \& } a < 5 \\ a - 7 & a \geq 5 \text{ \& } a < 7 \\ 0 & a \geq 7 \end{cases} \quad \text{Eq. (37)}$$

The determination of this function is needed to limit the variation of the  $a$ -parameter in a predefined interval and to ensure that the distance between the cameras ( $D$ ) is consistent with the maximum distance between the cameras and the fragments ( $d_{max}$ ).

The figure below (Figure 17) shows the function  $y$  when the parameter  $a$  varies.

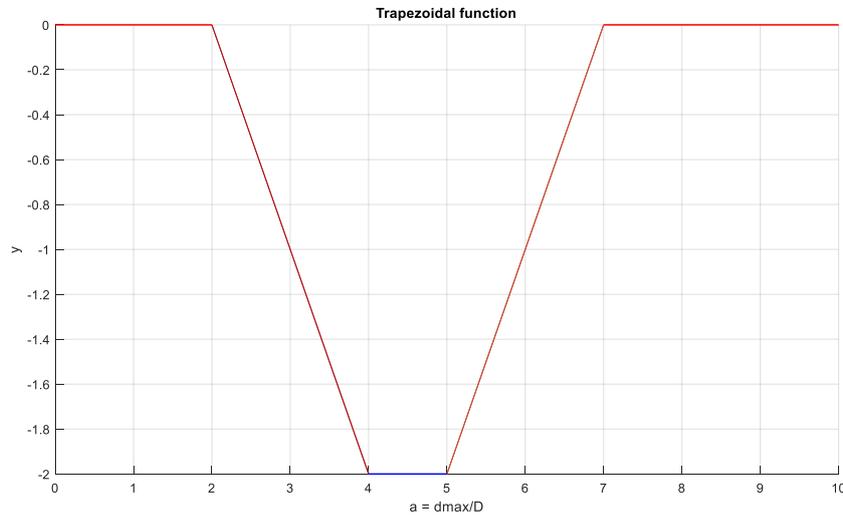


Figure 17: Trapezoidal function

The figure above (Figure 17) represents a trapezoidal function, defined in five intervals. It illustrates the values that the function  $y$  can assume in the  $a$ -interval that ranges from 2 to 7. All the  $y$ -values outside the range ( $a < 2$  and  $a > 7$ ) are equal to 0.

At the end, the position of the cameras is given by five variables:

- The angle  $\vartheta$ , which defines the position of the first camera ( $C_1$ ),
- The angle  $\omega_1$ , which represents the orientation of  $C_1$  in relation to the horizontal plane,
- The angle  $\omega_2$ , which outlines the orientations of  $C_2$  in relation to the horizontal plane,
- The segment  $r$ , that is the distance between the centre of the bench and cameras, also defined as the radius of the arc on which the cameras can move,
- The chord  $D$ , representing the distance between the two cameras.

The optimal placement of the cameras, with respect to the bench, to the fragments on the ground and between themselves, has been detected with a multi-objective optimization, able to:

- Maximize at the same time the overlapping area and the number of fragments lying inside this area,
- Determine the optimal distance between the cameras ( $D$ ), using a function that can describe how the relation between  $D$  and  $d_{max}$  can vary in a given range.

#### 4.1.4 Multi-objective optimization

The overlap of the cameras FOV, the number of fragments covered by the FOV intersection and the relation between distances fragment-cameras and camera-camera are the functions that need to be maximized. They depend on the position and the orientations of the cameras, defined by three angles  $\vartheta$ ,  $\omega_1$  and  $\omega_2$ , on the distance cameras-centre bench ( $r$ ) and on the distance between the first camera and the second one ( $D$ ).

The optimization problem, in which more than one function must be optimized at the same time, is known as multi-objective optimization.

Since three different functions need to be optimized simultaneously, a multi-objective optimization has been used to get the optimal values of  $\vartheta$ ,  $\omega_1$ ,  $\omega_2$ ,  $r$  and  $D$ , also known as independent variables. The solution of the multi-objective optimization is given by the Pareto front. The Pareto front represents a set of optimal solutions that works as a trade-off between these two objectives (Ref.).

One of the advantages of the Pareto optimization is that it allows a more comprehensive and clearer understanding of the optimization problem. Instead of focusing on a single objective, Pareto optimization considers multiple objectives and their interdependencies.

The multi-objective optimization (Custodio et al., 2011) is a type of optimization problem where there is the need of optimizing multiple functions simultaneously. These objectives may conflict with each other, meaning that improving one objective may produce the worsening of another.

The goal of multi-objective optimization is to find a set of solutions that are optimal with respect to multiple objectives, rather than a single optimal solution. This set of solutions should represent a good trade-off between the different objectives (Hwang, 1979).

One way to formulate a multi-objective optimization issue is as proposed by (Karim, 2000):

$$\max_{x \in X} (f_1(x), f_2(x), \dots, f_k(x)) \quad \text{Eq. (38)}$$

Where:

- $k$  is the number of objective functions. In the multi-objective optimization:  $k \geq 2$ .
- $X$  is the set of decision vectors, where  $X \subseteq \mathbb{R}^n$  and it depends on the  $n$ -dimensional domain.

$X$  can be defined by some constraint function

To be noted that the maximization of the objective function is equivalent to the minimization of its inverse, as follows, because MATLAB always minimises:

$$\min_{x \in X} (-f_1(x), -f_2(x), \dots, -f_k(x)) \quad \text{Eq. (39)}$$

An optimal solution of a multi-objective optimization is the one that achieves the best possible compromise among the different objectives. The Pareto-optimal solution or Pareto front is a set of solutions that represent a good trade-off between different objectives, where improving one can worsen another (Karim, 2000).

Finding the Pareto front is useful because it allows to compare the different conflicting objectives and choose between different trade-offs. By visualizing the Pareto front, it is possible to see which solutions are feasible and which trade-offs are available.

#### *Mathematical definition of the Pareto optimization*

The Pareto optimization is defined as follows:

$$\bullet \quad \forall i \in \{1, \dots, k\}, f_i(x_1) \leq f_i(x_2) \quad \text{Eq. (40)}$$

$$\bullet \quad \exists i \in \{1, \dots, k\}, f_i(x_1) < f_i(x_2) \quad \text{Eq. (41)}$$

Where:

- $x_1 \in X$  and  $x_2 \in X$  are the feasible solutions
- $k$  is the number of objectives, in this particular case  $k = 3$ .

A solution  $x^* \in X^*$  is called Pareto optimal if there does not exist another solution that dominates it. The set of Pareto optimal solutions ( $X^*$ ) is called Pareto front. Moreover, the Pareto front of a

multi-objective optimization is limited by upper and lower bounds. This means the set of solutions lies in a range, that must be defined as input.

The MATLAB algorithm adopted is *Paretosearch.m* algorithm. It is an iterative algorithm which defines a set of points, that must satisfy all the bounds and the constraints at each iteration. The resulting points must converge to the points near the true Pareto front. The problem has five decision variables or free variables, for which the upper and lower bounds have been defined.

The outputs of the Pareto search algorithm are the decision variable values ( $x^*$ ) and the objective function values ( $f_1(x^*), f_2(x^*), f_3(x^*)$ ).

The only point that represents the best possible outcome for all the objectives, simultaneously, is the so-called utopia point. This point is important because it provides a reference point for evaluating the trade-off and the quality of the solutions.

$$P_{utopia} = \min (-f_1(x^*), -f_2(x^*), f_3(x^*)) \quad \text{Eq. (42)}$$

Where:

- $P_{utopia}$  is a vector whose elements are the coordinates on the Pareto front of the utopia point.
- $f_1(x^*), f_2(x^*), f_3(x^*)$  are the objective functions.

Then, the distances between the utopia point and all the points on the Pareto front are calculated using the Euclidean distance formula. The trade-off has been found considering the point on the Pareto front that has the minimum distance from the utopia point.

The multi-objective function is defined as:

$$F_{x^* \in X^*} = [-f_1(x^*) \quad -f_2(x^*) \quad f_3(x^*)] \quad \text{Eq. (43)}$$

The inputs are the upper and lower bounds of the decision variable values.

$$ub = \left[ \frac{\pi}{2} \quad 2\pi \quad 2\pi \quad 250 \quad 50 \right] \quad \text{Eq. (44)}$$

$$lb = [0 \quad 0 \quad 0 \quad 100 \quad 1] \quad \text{Eq. (45)}$$

The outputs are the values of the objective functions, which are the elements of the vector  $F$ , that has  $i$ -th rows and 3 columns, as shown in Eq. (43) and the decision variable values (Eq. (46)).

$$x^* = [\theta \quad \omega_1 \quad \omega_2 \quad r \quad D] \quad \text{Eq. (46)}$$

Where:

- $F$  is a vector containing the three objective functions: the elements of the first column are  $f_1(x^*)$ , the ones of the second column are  $f_2(x^*)$  and the elements of the third column are  $f_3(x^*)$ .
- $f_1(x^*)$  is the objective function that maximise the overlap between the two fields of view.
- $f_2(x^*)$  is the objective function that maximise the number of fragments that are covered by the two fields of view.
- $f_3(x^*)$  is the objective function that describes how the function  $y$  (Eq. (37)) can vary if  $a = \frac{a_{max}}{D} \in [2, 7]$ .
- $x^*$  is the decision vector made by the three independent variables (decision variables).
- $ub$  is the upper bound for the decision variables.
- $lb$  is the lower bound for the decision variables.

In the figure below (Figure 18), the Pareto front is shown as a result of the *Paretosearch.m* algorithm. The x-axis represents the Objective 1, which is the overlapping area of the two fields of view, while the y-axis represents the Objective 2, that is the number of points inside the overlapping area and the z-axis is the Objective 3, which shows the  $y$  function.

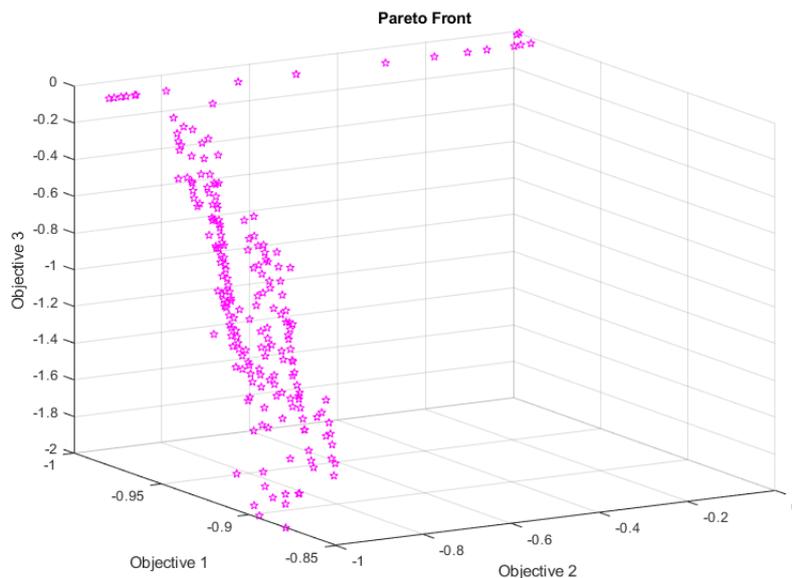


Figure 18: Pareto front

Each point in the figure (Figure 18) represents a possible configuration of  $\theta, \omega_1, \omega_2, r, D$  and the corresponding values of the objective functions  $f_1(\theta, \omega_1, \omega_2, r, D)$ ,  $f_2(\theta, \omega_1, \omega_2, r, D)$  and  $f_3(\theta, \omega_1, \omega_2, r, D)$ .

In case of three objective functions, the Pareto front turns out to be a surface. For any solution on the Pareto front, it is not possible to improve one objective without worsening the other. All the objectives are equally important and have the same weight and they are all maximised by the decision variables. For this reason, the utopia point is a reference point to get the optimal solution. The solutions are better or worse, depending on how close they are to the utopia point. The closer to the utopia point the solutions are, the better they are considered.

The Euclidean distances between the utopia point and all the points on the Pareto front have been evaluated to determine the trade-off solution among all the possible ones.

$$d_{P_{utopia}} = \min \left( \sqrt{(P_{utopia}(1) - (-f_1))^2 + (P_{utopia}(2) - (-f_2))^2 + (P_{utopia}(3) - f_3)^2} \right) \quad \text{Eq. (47)}$$

Where:

- $d_{P_{utopia}}$  is the minimum distance between the utopia point and the points on the Pareto front, defined by values of  $-f_1, -f_2$  and  $f_3$ .
- $-f_1, -f_2$  and  $f_3$  are the vectors of the objective functions.

The following figure (Figure 19) represents the Pareto front, the utopia point and the optimal solution or trade-off solution.

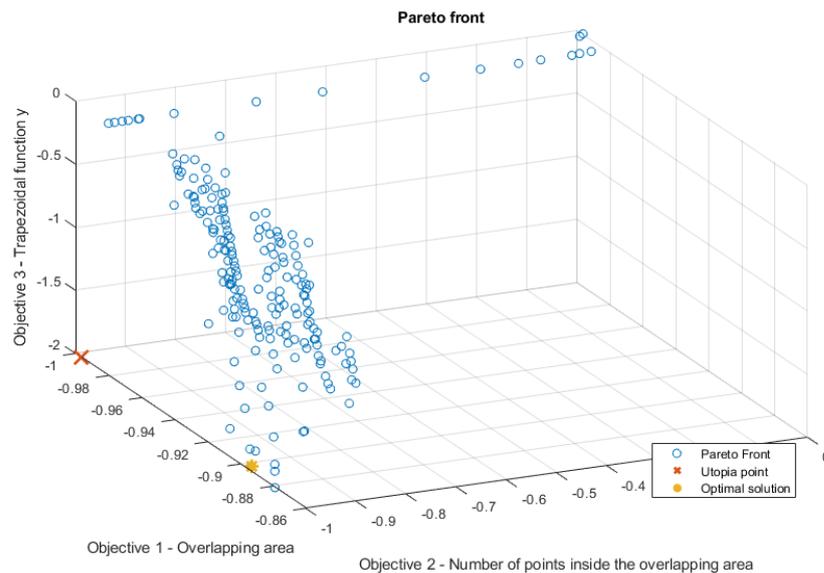


Figure 19: Pareto front, utopia point and trade-off solution

Finally, the solution is identified by the point on Pareto front, which has the minimum distance from the utopia point. This point gives information on the decision variable values ( $x^*$ ) and the objective function values ( $f_1(x^*), f_2(x^*), f_3(x^*)$ ).

$$x^* = [78.05^\circ \quad 265.22^\circ \quad 267.19^\circ \quad 153.30 \text{ m} \quad 34.30 \text{ m}] \quad \text{Eq. (48)}$$

$$F = [-f_1(x^*) \quad -f_2(x^*) \quad f_3(x^*)] = [-91.02 \quad -4759 \quad -2] \quad \text{Eq. (49)}$$

Where:

- $f_1(x^*) = 91.02 \%$  is the percentage of the overlapping area.
- $f_2(x^*) = 4759$  is the number of fragments (out of 5000) that lie in the overlapping area.
- $f_3(x^*) = -2$  is the value of the function  $y$  (Eq. (37)) for  $a = \frac{d_{max}}{D}$  that belongs to the range between 2 and 7, resulting in  $D = 34.30 \text{ m}$ .

Once determined the optimal values of  $\theta, \omega_1, \omega_2, r$  and  $D$ , that optimize the area on the ground that is visible from both cameras simultaneously, the number of fragments contained by this area, and the value of  $a$ , it is possible to place the cameras.

The figure below (Figure 20) shows the projection of the fragments on the ground, as well as the angle of view of each camera and the intersection polygon, whose border is highlighted in green. It represents the optimal position and orientation of the cameras with respect to the fragments.

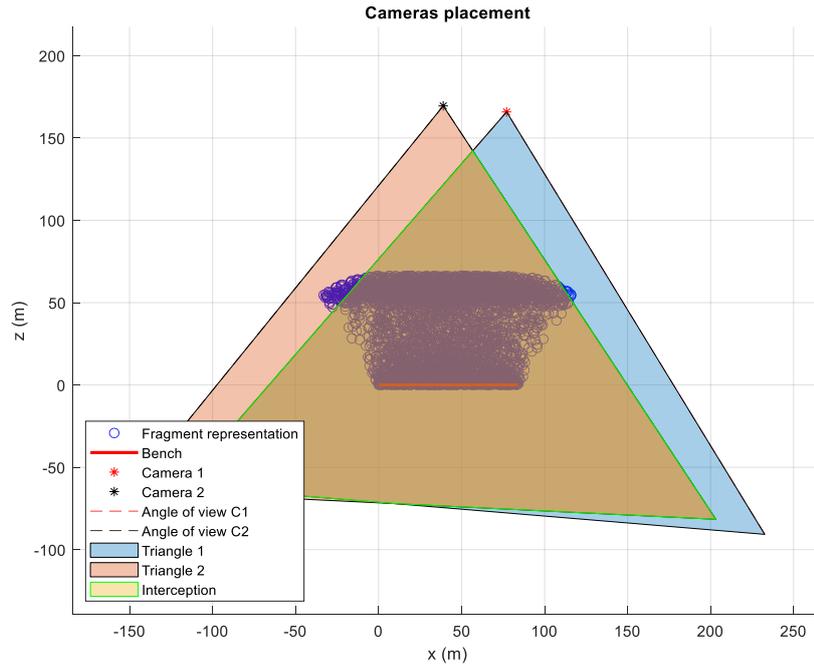


Figure 20: Optimal position and orientation of the cameras

In the figure above (Figure 20), the cameras are represented with asterisks, and positioned at different angles. The position of the *Camera 1* (in red) is defined by  $\vartheta = 78.05^\circ$  and the location of *Camera 2* (in black) is determined by  $(\beta + \vartheta) = 91.04^\circ$ , away from the first one of 34.30 m. Moreover, the distance between the centre of the bench and the cameras is about 153.30 m.

The two triangles have been used to determine the angle of view of each camera and its orientation with respect to the North. The orientation of the first camera is about  $265.22^\circ$  ( $\omega_1$ ) and the one of the second camera is  $267.19^\circ$  ( $\omega_2$ ). The triangles are the representation of the area on the ground that is visible from each camera. The intersection of these triangles results in a third polygon, which represents the area on the ground that is visible from both cameras. It represents one of the maximum possible overlapping areas, selected with the criteria of the utopia point, that also contains the greatest number of fragments.

#### 4.1.4 Camera selection

The camera selection process ends with the selection of the cameras, which involves the analysis of the existing relationship between the results obtained with the motion analysis of the blast face (fragment size and placement of the cameras) and the features of the cameras (focal length and pixel size, which depends on the sensor type).

In the figure below (Figure 21), a simplified scheme that combines the cameras features and the values coming from the motion analysis, as for example the fragment size and the distance between the farthest fragment and the cameras, is shown, where:

- $\phi$  is the diameter of the fragment (m).
- $d_{max}$  is the distance of the camera from the fragment (m).
- $\alpha$  is the angle between the radius of the fragment and the focal point ( $^{\circ}$ ).
- $f$  is the focal length of the camera (mm).
- $x$  is the pixel size able to capture the fragment ( $\mu\text{m}$ )

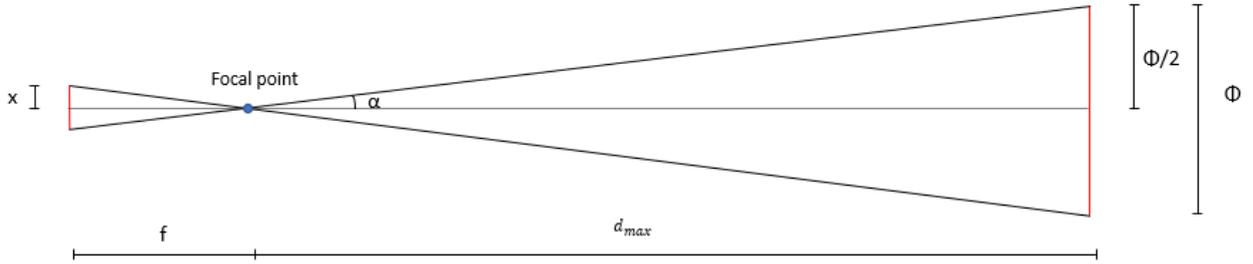


Figure 21: Graphical representation of the relationship between fragment size-max distance and pixel size-focal length

According to the geometrical relationships among the elements in the figure, the following equations can be used to finally select the proper camera:

$$\tan(\alpha) = \frac{x}{f} = \frac{r}{d_{max}} \quad \text{Eq. (50)}$$

$$x = \tan(\alpha) \cdot f \quad \text{Eq. (51)}$$

$$px = \frac{x}{\text{pixel size}(\text{sensor})} \quad \text{Eq. (52)}$$

From these equations, all the parameters are known, apart from the pixel size able to capture a fragment.

The following table (Table 7) represents a summary of all the characteristics used to identify the best camera. As it is possible to notice the cameras' features and the parameters coming from the design are combined in the Eq. (50-52) to get the number of pixels able to capture a fragment with a mean diameter of 58.9 cm, if the cameras are 153.3 m far from the bench.

Table 7: Relation between cameras features and geometrical parameters

CAMERAS FEATURES					
Type		Pixel size in camera sensor	Focal length	Shooting time with HFR mode	Num. of pixel to capture a fragment with $\phi = 58$ mm
		$\mu\text{m}$	mm		px
Sony	DSC-RX100 VII	1,12	9	approx. 6/7 s	31
Sony	ZV-1	1,12	9,4	approx. 3 s	32
Sony	RX10 III	1,12	8,8	approx. 4 s	30

GEOMETRICAL PARAMETERS					
Diameter $\phi$ (m)	Radius r (m)	Max distance $d_{max}$ (m)	$\tan(\alpha)$ rad	$\alpha$ °	Duration of blast s
0,5788	0,2894	153,2982	0,0019	0,108	3,82

The number of pixels has been evaluated for all the remaining cameras coming from Table 5, to make a comparison among the different types and to choose the best one. It has been assumed to put the cameras all at the same distance (153.3 m), pointing towards the same fragment of a size equal to 58 cm.

All the three different types of cameras listed have reached a valuable final result because the order of magnitude of the number of pixels is the same. However, the camera that requires the lowest amount of pixels to detect the same fragment from the same distance is the Sony ZV-1, with 32 px.

Since one of the main objectives is to record video with HFR, the maximum time that a video can last in high frame rate mode is therefore an important parameter to be considered. To be sure that the shooting time available from these cameras is long enough to record the event, the time required by the fragments to reach the ground has been evaluated. The blast will last 3.82 s, which involves that the Sony ZV-1 needs to be discarded because its shooting time is about 3 s.

Although the remaining two cameras (Sony DSC-RX100 VII and Sony RX10 III) are able to record the event, it is still preferable to choose the camera that guarantees the highest shooting time values.

All things considered, the best choice turns out to be the Sony DSC-RX100 VII. It is able to capture a fragment with 31 px, and its shooting time with HFR mode is much higher (6/7 s versus 4 s of the Sony RX10 III).

## **4.2 Photogrammetric analysis**

The reconstruction of the 3D trajectory of an object is based on calculating its coordinates in real space at different time instants. This is accomplished by using the photogrammetric principle of collinearity, which states that the object points in the images, the object point in space, and the projection centre of the cameras must lie on the same line. By ensuring the intersection of the optical axes of the two photographs, the three-dimensional position of the object may be determined (Ref.). Prior to solving the collinearity equations, a stereo calibration process is conducted to determine the orientation parameters of the cameras, including translation and rotation. This calibration step is crucial as it provides the necessary information to account for the number of unknowns in the collinearity equations system, enabling a solution to be obtained. Additionally, the calibration process determines the distortion coefficients of the lenses, which are then used to correct the captured images.

Once the orientation parameters of the cameras are established, the object is recorded in high frame rate (HFR) mode using the two cameras, which are kept in the same position used during the capture of the pattern images for calibration. The object trajectory is obtained through frame analysis of the recorded videos, which capture different positions of the object along its entire path. Each specific position is identified by two homologous frames—one from each camera—corresponding to the same instant. Utilizing MATLAB's multiple objects tracking function, the object is tracked, resulting in the coordinates, measured in pixels, of its centroid at each instant. From the 2D coordinates of the object in each frame, it has been possible to reconstruct the 3D trajectory of the object in real space. This procedure provides valuable insights into the object's movement and position within the 3D environment.

### **4.2.1 Stereo camera calibration**

Calibration is an essential step to accurately convert 2D picture points to their corresponding 3D world coordinates. The stereo-calibration relies on detecting the corners of a known-sized pattern in all the images of a stereophotogrammetric set (Cabrera-Quirós et al., 2013). By identifying the same real points in both images of a pair, the algorithm evaluates the projections of these points from the world coordinates, defined by the pattern, into the image coordinates using the stereo parameters. Moreover, the algorithm aims to minimize the distances between the detected points and their reprojected counterparts. With the calibration it is possible to obtain the relative positions and the orientations of the two cameras in a stereo camera system.

It involves determining the internal and external orientations of cameras used to record object movements and correcting lens distortion. It is a one-time process since the cameras remain stationary during recording.

The internal orientation of the cameras, also known as intrinsic parameters, refers to obtaining the intrinsic parameters of the camera, which describe its internal functioning. These parameters are the constants in the camera, and they include the position of the main point of the image (where the optical axis intersects the camera sensor), the focal length (the distance between the optical centre and the sensor) and the lens distortion coefficients (both tangential and radial).

The external orientation, or extrinsic parameters, defines the position of the camera and its orientation in relation to the real world. In this case, it involves establishing the relative orientation of one camera (e.g. right camera) with respect to another (e.g. left camera), using the left camera as a reference system for the reconstruction of the object motion. In other words, the left camera represents the origin of the reference system of the object. These extrinsic parameters consist of a relative translation with respect to the reference system on which the reconstruction of the object trajectory is to be performed and the three camera angles:

- The pitch ( $\varphi$ ), which is the angle between the horizontal and the optical axis of the camera. Its positive value indicates that the camera is pointing downwards.
- The roll ( $\omega$ ), that represents the rotation around the optical axis, with positive values for counter-clockwise tilt.
- The yaw ( $\kappa$ ), which defines the orientation of the optical axis in the horizontal plane.

The figure below (Figure 22) shows a visual interpretation of the cameras angles.

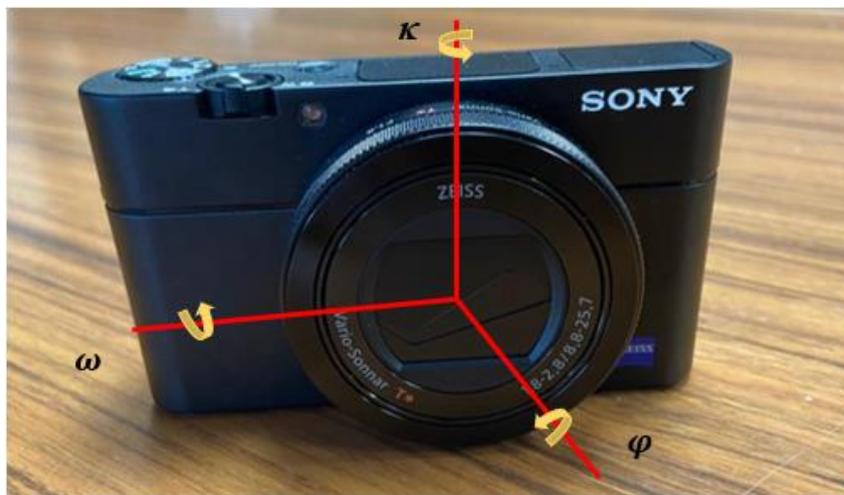


Figure 22: The three rotation angles of the right camera with respect to the left camera

The calibration has been performed with MATLAB, based on the images extracted from the recordings of the chessboard patterns of the left camera and the right one. From each frame obtained by the recorded video of the left camera, there is a corresponding frame recorded by the right camera. This pair of frames corresponds to a specific instant in the video.

The applied calibration methodology can be subdivided into different stages:

- Detection of checkerboard patterns: A set of stereo photos are analysed to find checkerboard patterns. The image points (corner coordinates) of the detected checkerboard corners are identified in this stage.
- Generation of checkerboard key-points: The algorithm generates the world coordinates of the checkerboard key-points once the checkerboard patterns have been identified. These key-points specify the spatial placements of the checkerboard corners in a real-world coordinate system.
- Calibration process: The calibration procedure uses the world points that correspond to the detected image points to determine the cameras' parameters. Both intrinsic (camera matrix and distortion coefficients) and extrinsic (rotation and translation) characteristics are included in this. These camera parameters are key parameters for precise depth measurement and 3D reconstruction.
- Reprojection error analysis: The code assesses the calibration's accuracy by looking at the reprojection errors. Reprojection errors quantify the separation between the projected image points (using the estimated camera parameters) and the corresponding detected image points. A good calibration gives lower reprojection errors.
- Visualization of extrinsic parameters: The code allows to visualize the extrinsic parameters to understand the relative positions and orientations of the cameras in the stereo camera system. This visualization shows how the cameras are orientated and placed in relation to the global/world coordinate system.
- Parameter estimation errors: The code also shows the parameter estimation errors, suggesting how accurately the camera parameters were estimated. These errors provide the level of uncertainty of the calibration process and can be used to find problems or suggest changes.

By identifying checkerboard patterns, determining camera characteristics, and analysing the relative positions and orientations of the cameras through the algorithm, a robust 3D reconstruction of the motion of the object is possible.

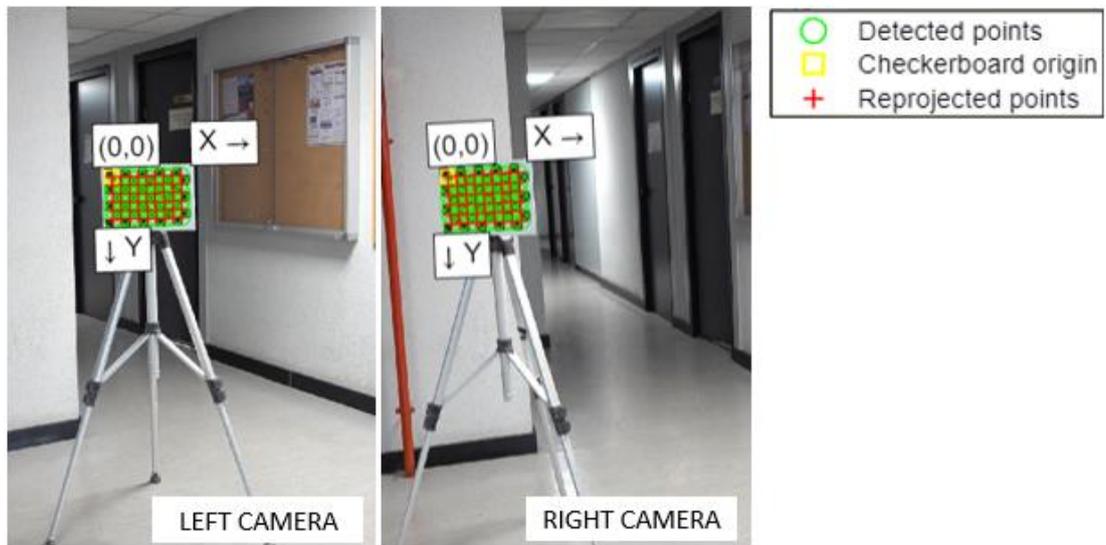


Figure 23: Example of a calibrated stereopair with Camera Calibrator App from MATLAB

In the figure above (Figure 23) an example of the stereo calibration arrangement is represented. It involves a calibration pattern 10-by-7 chessboard with 28 mm side squares. On the chessboard the key points (green circles), the reprojected points (red cross) and the origin of the chessboard reference system are represented.

The following figure (Figure 24) shows a zoom in on the same chessboards, seen from the two cameras.

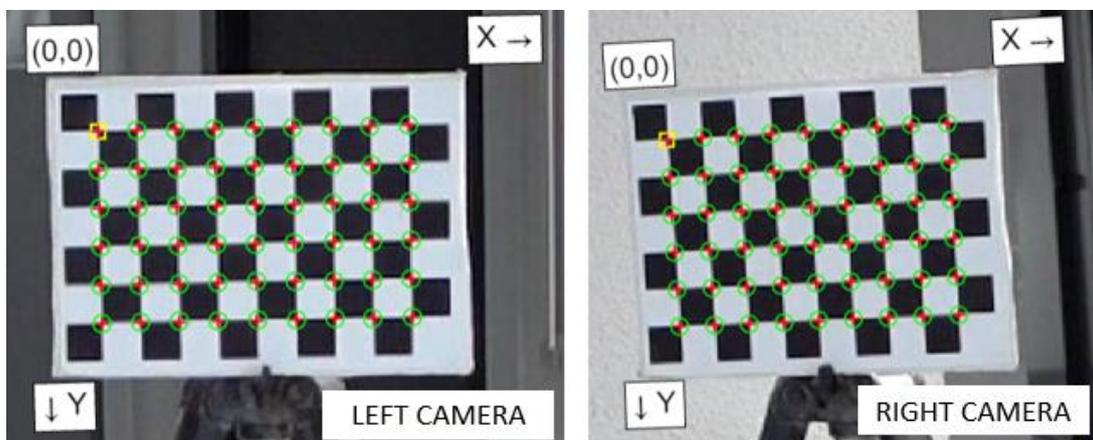


Figure 24: Zoom in of the calibrated chessboard patterns

### **4.2.2 Object tracking**

To detect moving objects in the videos, they are processed frame by frame using a motion-based multi-objects tracking example on MATLAB. Each frame of the video is read, and the motion detection is performed.

The tracking algorithm firstly divides the foreground from the background of the video by masking it through morphological operations, in order to remove noises and be able to detect what we are interested in.

Secondly, the blob analysis is performed on the foreground mask to identify and analyse related areas, or blobs, in an image. A blob is a collection of pixels with similar properties, such as texture, colour, or intensity.

Then, the code extracts information like areas and centroids of the objects detected, that are used to visualize the fragment position in each frame. To distinguish among the paths of each object in each frame and to screen undesired points or outliers, a filter is applied. The method applied performs point selection on the distance and the derivative criteria.

Overall, a motion-based multi-object tracking has been performed using foreground detection. It demonstrates the detection and tracking of moving objects in a video, as well as visualization and analysis of the centroid coordinates. The additional calculations add functionality for point selection or tracking based on distance and angle conditions.

### **4.3 Trajectory reconstruction**

The reconstruction of the 3D positions of the object by applying collinearity equations to each video frame is based on the works of (Wolf et al., 2014) and (Alsadik, 2014).

By analysing the same frame in both videos, it is possible to derive distinct 3D positions of the object for each frame pair, representing different time instants. These positions provide the reconstruction of a three-dimensional trajectory, as shown in the following figure (Figure 25).

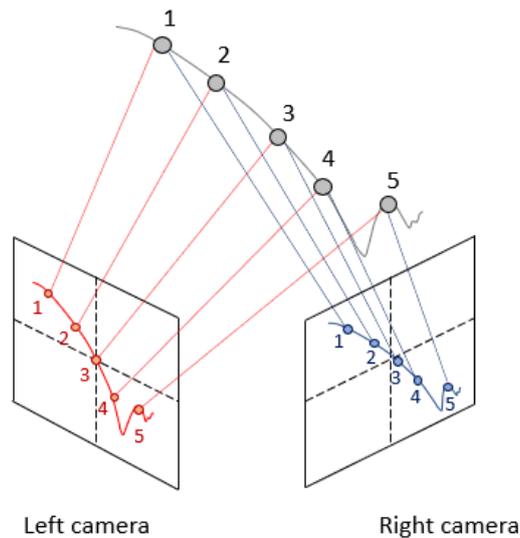


Figure 25: Trajectory seen from the two cameras. Each position (1-5) forms a stereoscopic pair.

An essential and highly useful geometric principle in photogrammetry is the collinearity criterion, as described by (Wolf et al., 2014). The concept of collinearity condition refers to a fundamental notion concerning the geometry of aerial or terrestrial photography and the related 3D objects being shot. The collinearity requirement is essential for photogrammetric operations because it enables the inference of three-dimensional objects locations from two-dimensional images. Correspondences between the image points and the object points must be established in order to precisely reproduce the geometry and dimensions of the object. The collinearity condition states that the optical centre of the camera, the image point in the 2D frame and the corresponding object point in 3D space must all lie on a straight line in a three-dimensional space. This condition ensures that the two photographs of the same object, taken at the same time, satisfy the collinearity condition.

The collinearity criterion is mathematically described by two equations, related to the x and y coordinates of a photo. These formulas express the correspondence between the image coordinates on the photograph and the spatial coordinates of the object point. The intrinsic and extrinsic camera parameters are connected to the 3D object coordinates (X, Y, Z) of the corresponding point in the scene as well as the image coordinates (x, y) of a point in the image. The external camera parameters provide the position and orientation of the camera in space, while the internal camera parameters describe the focus length, principal point coordinates, and distortion coefficients.

If the collinearity condition is satisfied, the intersection of the two homologous rays on the ground can be determined. This intersection point represents the three-dimensional position of the object point.

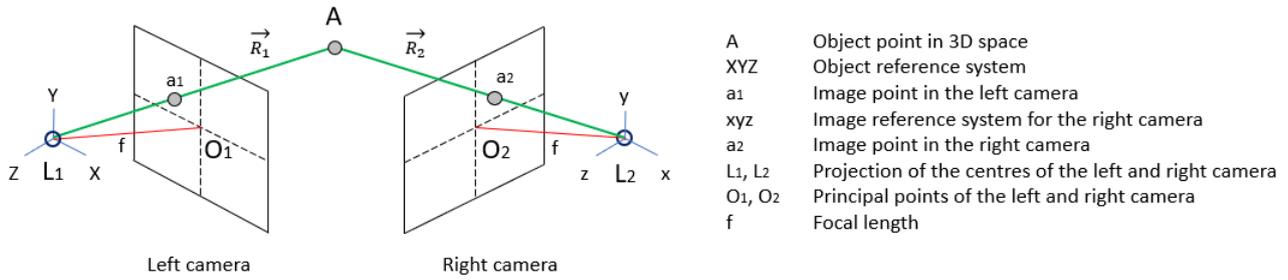


Figure 26: Collinearity condition for the two cameras

In the figure above (Figure 26), the collinearity condition for both cameras has been graphically represented. The object  $A$  lies in a 3D space and its coordinates are described by the  $XYZ$  reference system. The image system of the left camera coincides with the object reference system. Indeed, the photo-coordinates of  $a_1$  on the left camera are given by the same reference system of the object  $A$ , while the object point  $a_2$  on the right camera is described by an instrumental reference system (image system)  $xyz$ .  $L_1$  and  $L_2$  are respectively the projections of the centres of the left and the right camera.  $O_1$  and  $O_2$  are the principal point of the cameras and  $f$  is the focal length.

To get the 3D coordinates of the object in each frame, the MATLAB function *triangulate.m* has been adopted. This function involves the application of the collinearity equations to perform the 3D triangulation of points.

Triangulation is the method of determining the 3D position of a point in space, by intersecting the projection beams from many camera viewpoints (Hartley & Sturm, 1995). The fundamental idea is that if a point is visible from two or more camera field of views, it is possible to calculate its 3D position by locating the intersection of the projection rays originating from those fields of view.

### Collinearity equations and triangulation: general rules

The collinearity equations are presented below, according to (Wolf et al., 2014):

$$x_{ai} = x_{0i} - f \cdot \left( \frac{r_{11i}(X_A - X_{Li}) + r_{12i}(Y_A - Y_{Li}) + r_{13i}(Z_A - Z_{Li})}{r_{31i}(X_A - X_{Li}) + r_{32i}(Y_A - Y_{Li}) + r_{33i}(Z_A - Z_{Li})} \right) \quad i = 1,2 \quad \text{Eq. (53)}$$

$$y_{ai} = y_{0i} - f \cdot \left( \frac{r_{11i}(X_A - X_{Li}) + r_{12i}(Y_A - Y_{Li}) + r_{13i}(Z_A - Z_{Li})}{r_{31i}(X_A - X_{Li}) + r_{32i}(Y_A - Y_{Li}) + r_{33i}(Z_A - Z_{Li})} \right) \quad i = 1,2 \quad \text{Eq. (54)}$$

Where:

- $i$  is the index which identifies one of the two images of the stereo-pairs.
- $x_a, y_a$  are the photo coordinates of the image point  $a$ .
- $x_0, y_0$  are the coordinates of the principal points, known from the camera calibration.
- $r_{11}, r_{12}, r_{13}, r_{31}, r_{32}, r_{33}$  are the components of rotation matrix of the camera, knowing its intrinsic and extrinsic parameters.
- $X_a, Y_a, Z_a$  are the space coordinates of the object point  $A$ . They are the unknowns of these equations.
- $X_L, Y_L, Z_L$  are the space coordinates of the centre of projection  $L$ .
- $f$  is the focal length of the cameras.

For each stereo-pair (one for each instant) there are 4 non-linear equations (2 for each photograph), with 3 unknowns ( $X_A, Y_A, Z_A$ ) and 1 degree of freedom. These equations should be applied for each frame extracted from the video.

These equations make an overdetermined system without an exact solution. To obtain a result that can almost satisfy the equations, approximations of this system have been realized through the least squares method, applied on the linearised equations. For this reason, if the images are captured by more cameras (more than 2), the approximations will be more accurate, and the system will have more degree of freedom (Handayani, 2010).

Actually, the significance of the stereo calibration procedure is evident because without the stereo calibration procedure, the orientation parameters would likewise be unknowns. In this situation the equation system will be characterized by 21 unknowns:

- 9 orientation parameters for each camera ( $X_L, Y_L, Z_L, \varphi, \omega, k, x_0, y_0, f$ ).
- 3 object coordinates ( $X_A, Y_A, Z_A$ ).

Since the least squares method would require at least 22 equations to approximate the solution, this problem would be impossible to resolve.

Due to lens distortions, the measured photo-coordinates may contain slight errors. The Brown-Conrady distortion model (Brown, 1966) is used to take these systemic mistakes into account.

$$x_a = x'_a + (x'_a - x_0) \cdot (q_1 \cdot r^2 + q_2 \cdot r^4) + (p_1 \cdot (r^2 + 2 \cdot (x'_a - x_0)^2) + 2 \cdot p_2 \cdot (x'_a - x_0) \cdot (y'_a - y_0)) \quad \text{Eq. (55)}$$

$$y_a = y'_a + (y'_a - y_0) \cdot (q_1 \cdot r^2 + q_2 \cdot r^4) + (2 \cdot p_1 \cdot (x'_a - x_0) \cdot (y'_a - y_0) + p_2 \cdot (r^2 + 2 \cdot (y'_a - y_0)^2)) \quad \text{Eq. (56)}$$

$$r = \sqrt{(x'_a - x_0)^2 + (y'_a - y_0)^2} \quad \text{Eq. (57)}$$

Where:

- $x_a, y_a$  are the undistorted coordinates of the object.
- $x'_a, y'_a$  are the distorted coordinates of the object.
- $q_1, q_2$  are the radial distortion coefficients.
- $p_1, p_2$  are the tangential distortion coefficients.

The non-linear collinearity equations are linearised as follows through the Taylor's theorem:

$$F_i = x_{0i} - f \cdot \frac{r_i}{q_i} = x_{ai} \quad \text{Eq. (58)}$$

$$G_i = y_{0i} - f \cdot \frac{s_i}{q_i} = y_{ai} \quad \text{Eq. (59)}$$

Where:

$$q_i = r_{31i}(X_A - X_{Li}) + r_{32i}(Y_A - Y_{Li}) + r_{33i}(Z_A - Z_{Li}) \quad \text{Eq. (60)}$$

$$r_i = r_{11i}(X_A - X_{Li}) + r_{12i}(Y_A - Y_{Li}) + r_{13i}(Z_A - Z_{Li}) \quad \text{Eq. (61)}$$

$$s_i = r_{21i}(X_A - X_{Li}) + r_{22i}(Y_A - Y_{Li}) + r_{23i}(Z_A - Z_{Li}) \quad \text{Eq. (62)}$$

To simplify the collinearity system, the Taylor's theorem has been applied to approximate the Equation (58) up to the first order of the partial derivatives. It can be used to get an approximation of the polynomial-based collinearity problems for a given point.

$$x_a = F_0 + \left(\frac{\partial F}{\partial \omega}\right)_0 d\omega + \left(\frac{\partial F}{\partial \phi}\right)_0 d\phi + \left(\frac{\partial F}{\partial \kappa}\right)_0 d\kappa + \left(\frac{\partial F}{\partial X_L}\right)_0 dX_L + \left(\frac{\partial F}{\partial Y_L}\right)_0 dY_L + \left(\frac{\partial F}{\partial Z_L}\right)_0 dZ_L + \left(\frac{\partial F}{\partial X_A}\right)_0 dX_A + \left(\frac{\partial F}{\partial Y_A}\right)_0 dY_A + \left(\frac{\partial F}{\partial Z_A}\right)_0 dZ_A \quad \text{Eq. (63)}$$

$$y_a = G_0 + \left(\frac{\partial G}{\partial \omega}\right)_0 d\omega + \left(\frac{\partial G}{\partial \phi}\right)_0 d\phi + \left(\frac{\partial G}{\partial \kappa}\right)_0 d\kappa + \left(\frac{\partial G}{\partial X_L}\right)_0 dX_L + \left(\frac{\partial G}{\partial Y_L}\right)_0 dY_L + \left(\frac{\partial G}{\partial Z_L}\right)_0 dZ_L + \left(\frac{\partial G}{\partial X_A}\right)_0 dX_A + \left(\frac{\partial G}{\partial Y_A}\right)_0 dY_A + \left(\frac{\partial G}{\partial Z_A}\right)_0 dZ_A \quad \text{Eq. (64)}$$

This point  $A (X_A^0, Y_A^0, Z_A^0)$  is the one determined by the intersection of the optical lines of each camera, based on the method proposed by (Instituto Geografico Nacional, 2008).

Through the stereo calibration process, the parameters describing the external orientation of the cameras (the rotation matrix  $R_C$  and the translation vector  $T_C$ ) are known. The unknowns in the Equation (64) are  $dX_A$ ,  $dY_A$  and  $dZ_A$ .

The linearised collinearity equations are the following:

$$b_{J14i} dX_A + b_{J15i} dY_A + b_{J16i} dZ_A = J_i \quad \text{Eq. (65)}$$

$$b_{J24i} dX_A + b_{J25i} dY_A + b_{J26i} dZ_A = K_i \quad \text{Eq. (66)}$$

Where:

- $b_j$  coefficients are the partial derivatives of the functions  $F$  and  $G$ , with respect to the indicated unknowns evaluated in the initial approximation.
- $J$  and  $K$  are equal to  $x_a - F_0$  and  $y_a - G_0$ , respectively.

The linear equation system defined by Equation (65) can be written in matrix form as follows:

$$\mathbf{B}_J \cdot \mathbf{X} = \mathbf{I} \quad \text{Eq. (67)}$$

Where:

$$\mathbf{B}_J = \begin{pmatrix} b_{J14_1} & b_{J15_1} & b_{J16_1} \\ b_{J24_1} & b_{J25_1} & b_{J26_1} \\ b_{J14_2} & b_{J15_2} & b_{J16_2} \\ b_{J24_2} & b_{J25_2} & b_{J26_2} \end{pmatrix} \quad \text{Eq. (68)}$$

$$\mathbf{X} = \begin{pmatrix} dX_A \\ dY_A \\ dZ_A \end{pmatrix} \quad \text{Eq. (69)}$$

$$\mathbf{I} = \begin{pmatrix} J_1 \\ K_1 \\ J_2 \\ K_2 \end{pmatrix} \quad \text{Eq. (70)}$$

As stated before, the system above is an overdetermined system of linear equations, made by 4 equations and 3 unknowns. This means that it does not give an exact solution for  $\mathbf{X}$ , so that a global least squares adjustment is required. The application of the least squares to solve  $\mathbf{X}$  means minimizing the sum of the squares of the components of the residual vector. The residuals are determined using:

$$\mathbf{V}_R = \mathbf{B}_J \cdot \mathbf{X} - \mathbf{I} \quad \text{Eq. (71)}$$

Applying Otero and Sevilla's (1989) criterion of minimization of the sum of squared residuals:

$$\mathbf{B}_J^T \mathbf{B}_J \cdot \mathbf{X} = \mathbf{B}_J^T \cdot \mathbf{I} \quad \text{Eq. (72)}$$

$$\mathbf{X} = (\mathbf{B}_J^T \cdot \mathbf{B}_J)^{-1} \cdot \mathbf{B}_J^T \cdot \mathbf{I} \quad \text{Eq. (73)}$$

The  $\mathbf{X}$  matrix of unknowns adjusts the  $X_A^0$ ,  $Y_A^0$ , and  $Z_A^0$  initial values. The same method has been repeated until the value of the elements of  $\mathbf{X}$  is negligible in order to build an iterative least square process, which improves the estimation of the coordinates of  $A$ .

## **5. Laboratory test**

In lab environments, photogrammetric analysis is essential for validating and improving the accuracy and the reliability of the methodology proposed. Laboratory experiments offer controlled settings, where accurate measurements can be made, enabling an exhaustive analysis of the photogrammetric methodology and the evaluation of its effectiveness.

The main goal of lab test is to assess the accuracy, precision, and difficulties of the procedure under investigation. The outcomes of these tests help in the establishment of best practices, the improvement of workflows and the development of photogrammetric technologies, resulting in more precise and consistent results in real-world settings.

In conclusion, the laboratory test offers a supervised environment to assess the precision and effectiveness of the proposed methodology. It is essential for testing the method, comprehending its limitations, and enhancing its use for accurate three-dimensional measurements.

### **5.1 Equipment**

As widely in the previous chapters, the choice and use of proper tools are crucial for the effective execution of photogrammetric analysis. The cameras, tripods, object to be tracked, calibration patterns, the triggering light, and two computers are among the crucial elements. Together, they create a complete setting for carrying out a trustworthy photogrammetric analysis.

#### *Cameras*

The cameras employed for this investigation are Sony DSC-RX100 VII (Sony, 2016). To be able to detect the 3D motion of an object, two identical cameras are necessary. These cameras provide great image quality and clarity thanks to their image sensors, ensuring the gathering of accurate and detailed data. They are suitable for on-site applications due to their lightweight and small form, which makes for simple transportation. Below, a photo (Figure 27) of the camera is represented.



*Figure 27: Camera Sony DSC-RX100 VII*

### *Tripods*

Two tripods (one for each camera) are used to securely hold the Sony cameras during data collecting in order to guarantee stability and precise positioning. Tripods offer a reliable support structure that helps to reduce unintentional camera movements and guarantee consistent imagery throughout the analysis process. The adaptability and adjustability of the tripods provide flexible positioning, enabling optimal camera angles and scene coverage.

The chosen tripod is TSL08CN00 Slim Tripod from Benro (Benro EN-GBP), made of carbon fiber to minimize its weight. It can support 4 kg and includes a ball head and a quick release plate. They are ideal for on-site applications because they are light (1.0 kg) and compact, so that they can be easily carried and stored. The tripods height ranges between 510 mm, when folded, to its maximum value of 1463 mm. Its body is made by 4-sections legs, that can be independently spread to get the wanted height. Thanks to a primary locking knob, placed in the head, it is possible to control the orientation of the camera, according to specific needs. Moreover, the presence of a built-in bubble level is an important component for precise camera orientation. It aids to keep the horizons level and the images stable, consistent, and aligned precisely. In addition, the tripods are equipped with rubber feet, which guarantee a greater stability on various surfaces. In the following figure (Figure 22), the TSL08CN00 Slim Tripod is shown.



*Figure 28: TSL08CN00 Slim Tripod*

### *Object*

To track and examine the item's motion, a simple ball is chosen as the object of interest. It is perfect for tracking test because of its highly recognisable features, as its rounded shape. The cameras will capture the ball's movements inside the scene, allowing for the later extraction of its 3D trajectory and motion traits. The following figure (Figure 23) represents the ball to be tracked.

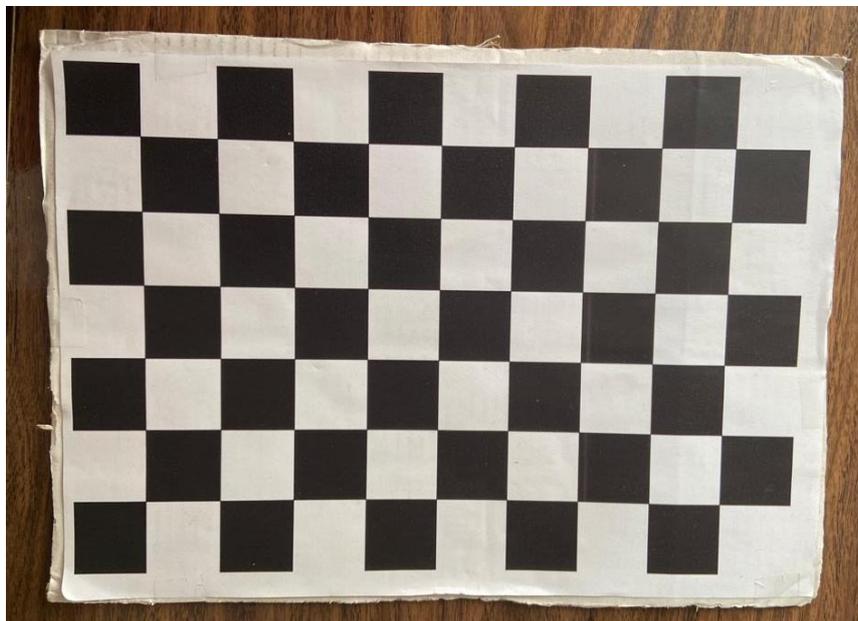


*Figure 29: Ball as tracked object*

### *Calibration pattern*

The Sony cameras are calibrated using a chessboard-shaped calibration pattern, according to (Cabrera-Quirós et al., 2013). The size of the chessboard employed is 10-by-7 squares of 28 mm side each. In order to make it easier to estimate camera parameters like focus length, lens distortion, and spatial orientation, the checkerboard design offers a clearly defined and instantly recognizable pattern of reference points. The 3D coordinates of the object are precisely measured and reconstructed thanks to calibration.

The figure below (Figure 30) shows the tool employed for the calibration process.



*Figure 30: Chessboard pattern*

### *Triggering light*

A triggering light is used to identify when the object starts moving and to synchronize the recordings made by the two cameras. The triggering light acts as a visual signal of the time zero of the video, which means the instant when the motion starts. The reference time set provided by the light enables the synchronization with the captured video frames. With this synchronization it is possible to conduct a reliable time-based analysis and to ensure an accurate matching between the object's trajectory and the recorded data. In the case under investigation, the triggering light was made with the flash of a smartphone.

### *Personal computers*

Two computers are connected to the cameras as part of the configuration/setup to perform the photogrammetric analysis. During the data collecting process, these computers are essential for remotely controlling the cameras because, for safety reasons, it is not possible for the user to be physically near the cameras.

The remote-control capability is facilitated through the use of the TeamViewer app, which enables seamless control and monitoring of the camera settings and operations from a smartphone. Users who have the software (the TeamViewer app) installed on their smartphone can remotely access and operate the linked PCs, making it possible to manage the cameras from a portable and approachable interface. The capacity to control computers remotely makes it unnecessary to physically touch them while collecting data, resulting in a more efficient and safety process.

This set of tools provides data collection, making it easier to analyse and reconstruct the 3D trajectory of the object and motion characteristics.

The two cameras are kept stationary throughout the recording, looking either at the moving object for trajectory reconstruction or at the calibration pattern provided for stereo calibration.

Below, the arrangement of the whole equipment is represented in (*Figure 31*).



*Figure 31: Setup of the equipment*

## 5.2 Realization of the photogrammetric sets

The laboratory trial starts with the data acquisition. It is necessary to record and extract frames from the videos of calibration and moving object.

### *Photogrammetric set for calibration*

The videos of the calibration pattern are taken by placing the chessboard at different positions within the field of view of the camera and by slightly rotating the pattern in various orientations while the cameras are kept fixed. The videos are recorded with the same HFR that is going to be used to record the object's motion. After many tests with different values of HFR (250 fps, 500 fps and 1000 fps), a HFR of 250 fps is considered appropriate because it guarantees enough number of frames and an adequate processing time.

From the video of the calibration pattern recorded from the left camera, the frames are extracted randomly. Since each frame is associated to timestamps, it is possible to extract frames from the video, made by the right camera, at precisely the same moments as in the first video. All the frames are then saved as individual images and a correction of distortion due to the lenses is applied on them.

### *Photogrammetric set for moving object detection*

The videos are recorded with a HFR value of 250 fps (same value used for the calibration videos). The frames are extracted from the recorded videos from both the left and the right cameras. Among all the frames extracted, only some frames are chosen. From the point of the lighting signal onward, the frames of interest are selected. The lighting signal acts a triggering mark, indicating the starting point from which the frame must be considered for subsequent processing. Then, lens distortion is removed from the chosen frames to ensure that they have correct and undistorted representations of the captured scene. Moreover, a new video sequence is realised by merging together the unaltered frames in time sequence. After that, a multi-object tracking analysis is applied to the generated video. To do this, the ball motion within the movie is identified and tracked.

By following these stages, the frames coming from both calibration and moving object videos are accurately extracted and corrected from lens distortion to guarantee a better quality of the analysis.

### 5.3 Lens distortion correction

Lens distortion is a phenomenon where objects captured by cameras appear to have distorted shapes. This is typically caused by the features of the camera lens.

As stated by (Wolf et al., 2014) and (Wang, 1990), lens distortions can be categorized as decentring or symmetric radial distortions. Both occur if light rays are bent or shift directions, preventing them from emerging parallel to their incoming directions after passing through the lens.

Symmetric radial distortion results in the distortion of imaged points along radial lines from the optical axis. While decentring distortion refers to an off-centre distortion pattern results, which contains both tangential and asymmetric radial components. Distortion is an unavoidable effect of lens manufacturing, but its effects can be greatly diminished with careful design. Traditional camera calibration techniques only supplied data on the symmetric radial component.

The correction of the lens distortion is necessary to:

- Improve the quality of the images: Images that have been deformed can be stretched, appear fuzzy, or have other anomalies that affect the image's overall perception and visual appeal. Lens distortion can be corrected to return things to their original size and shape for accurate and appealing photographs.
- Obtain more accurate measurements: Lens distortion can cause uncertainties, especially in cases where a recognition and reconstruction of the motion of an object is performed. Lens distortion correction creates outcomes/results that are more precise and trustworthy, by ensuring that measurements and object detection algorithms are implemented appropriately.

More in detail, the radial distortion value shows the radial displacement between the collimator cross's ideal location and its actual image. The method for calculating the values of radial lens distortion coming from (Wolf et al., 2014) involves fitting a polynomial curve to a plot of the displacements vs radial distances.

Based on lens design theory, the polynomial has the following structure:

$$\Delta r = k_1 \cdot r^1 + k_2 \cdot r^2 + k_3 \cdot r^5 + k_4 \cdot r^7 \quad \text{Eq. (74)}$$

Where:

- $\Delta r$  is the value of the radial distortion (mm).
- $r$  is the radial distance from the principal point (mm).
- $k_1, k_2, k_3, k_4$  are the coefficients of the polynomial. The distortion data from the calibration are used to solve them using the least squares method.

The distance  $r$  from an image point to the primary point is calculated and used to compute the value of  $r$  from Eq. (74) in order to correct the  $x$  and  $y$  position of the image point. To achieve this, firstly the  $x$  and  $y$  coordinates have been converted into  $\bar{x}$ ,  $\bar{y}$  coordinates, which describe the relative position of the image point with respect to the principal point through the following equations:

$$\bar{x} = x - x_p \quad \text{Eq. (75)}$$

$$\bar{y} = y - y_p \quad \text{Eq. (76)}$$

Where:

- $\bar{x}$ ,  $\bar{y}$  are the relative coordinates of the image point (with respect to the principal point).
- $x$ ,  $y$  are the coordinates of the image point.
- $x_p$ ,  $y_p$  are the coordinates of the principal point.

Then, the radial distance from the principal point has been computed:

$$r = \sqrt{\bar{x}^2 + \bar{y}^2} \quad \text{Eq. (77)}$$

Finally, the radial distortion ( $\Delta r$ ) has been evaluated. The corrections  $\delta x$  and  $\delta y$  (related to  $x$ ,  $y$  components) have been obtained by using the similar-triangle relationship, as shown in the figure below (Figure 32).

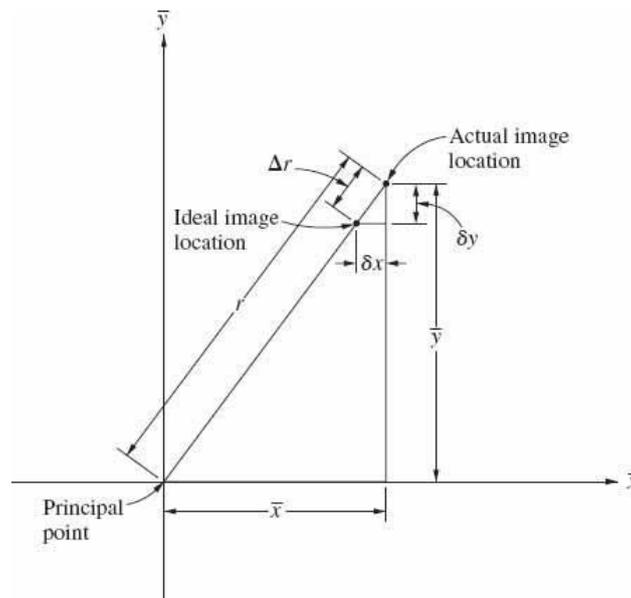


Figure 32: Relationship between radial lens distortion and corrections to  $x$  and  $y$  coordinates from (Wolf, 2014)

Based on the similarity between triangles, the following relationships are valid:

$$\frac{\Delta r}{r} = \frac{\delta x}{\bar{x}} = \frac{\delta y}{\bar{y}} \quad \text{Eq. (78)}$$

From which:

$$\delta x = \bar{x} \cdot \frac{\Delta r}{r} \quad \text{Eq. (79)}$$

$$\delta y = \bar{y} \cdot \frac{\Delta r}{r} \quad \text{Eq. (80)}$$

Through these correction values, it has been possible to get the corrected coordinates  $x_c$  and  $y_c$ :

$$x_c = \bar{x} - \delta x \quad \text{Eq. (81)}$$

$$y_c = \bar{y} - \delta y \quad \text{Eq. (82)}$$

Each frame (coming from whether the calibration recordings nor the moving object videos) has gone through the aforementioned process applying a built-in MATLAB function *undistortImage.m*, which takes as input the stereo parameters coming from the calibration.

## 5.4 Result and discussion

All the explained methodology has been applied to the recorded videos from the left and the right cameras.

In the figure below (Figure 33), the results obtained after the calibration process are shown. In blue the left camera is represented while the red camera is in red. The coloured squares in the figure are the depiction of the calibration patterns in the real-world coordinates.

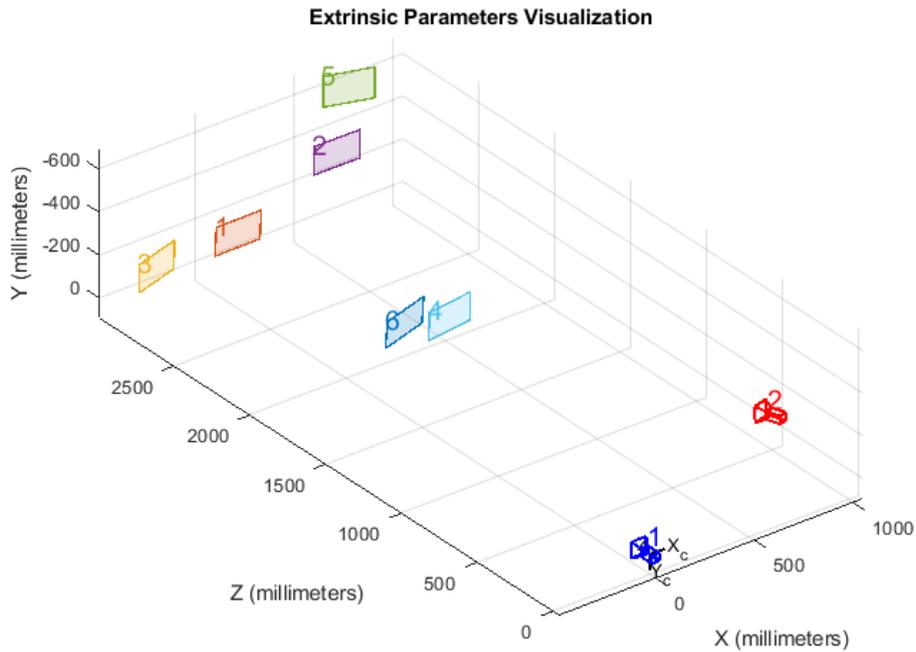


Figure 33: Extrinsic parameters obtained with calibration

In the following table (Table 8), the intrinsic and the extrinsic parameters coming from the calibration process are listed.

Table 8: Intrinsic and extrinsic parameters coming from the calibration

CAMERA PARAMETERS				
INTRINSIC				
	Camera Left		Camera Right	
Focal length	1,67E+03	1,66E+03	1,66E+03	1,65E+03
Principal point	981,89	515,43	930,14	563,58
Image size	1080	1920	1080	1920
Radial distortion	0,07	-0,22	-0,07	0,21
Tangential distortion	0	0	0	0

CAMERA PARAMETERS		
EXTRINSICS		
	Camera Left	Camera Right
Rotation matrix	$R = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$R = \begin{bmatrix} 0.96 & 0.01 & 0.28 \\ 0.002 & 0.99 & -0.04 \\ 0.28 & 0.04 & 0.96 \end{bmatrix}$
Translation vector	$T = [0 \ 0 \ 0]$	$T = [-1122 \ 76.60 \ -192.48]$

After, the accuracy of the calibration has been evaluated by examining the reprojection errors, which measure the distance between the projected image points and the corresponding detected image points.

As shown in the following figure (Figure 34), the mean reprojection error is around 0.16 pixels. This value is low enough to state that the calibration process is robust and reliable.

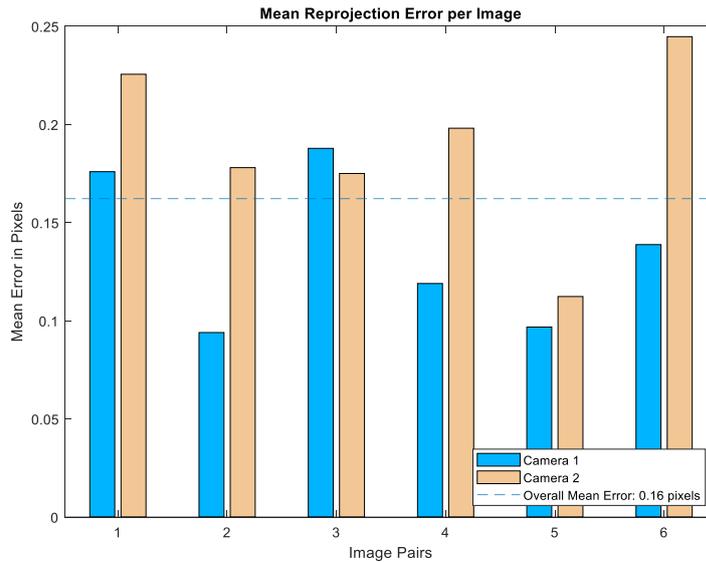


Figure 34: Mean reprojection errors per image

Once the calibration has been performed, the analysis of the videos takes place. Through the motion based multi-objects tracking, it is possible to recognise the path of the object in the video, as represented in the figure below (Figure 35).

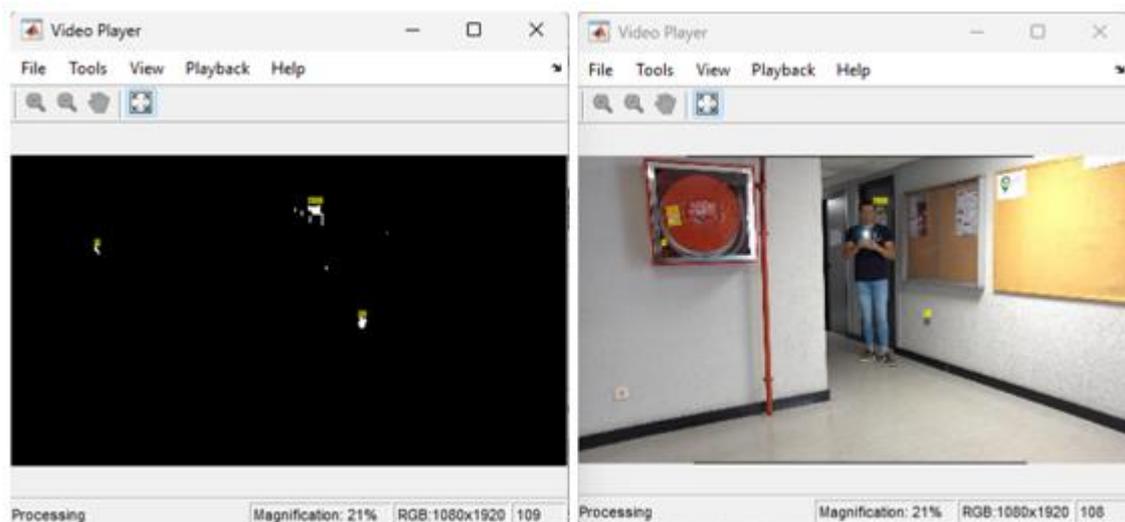


Figure 35: Application of the motion based multi-object algorithm in frame 108 (Left camera)

The figure above (Figure 35) shows the masked video on the left and the normal video on the right. In the masked video it is depicted in a clear way the difference between the foreground and the

background. Through the *Blobanalysis.m*, the x-y coordinates of the centroid of the object have been identified in each frame.

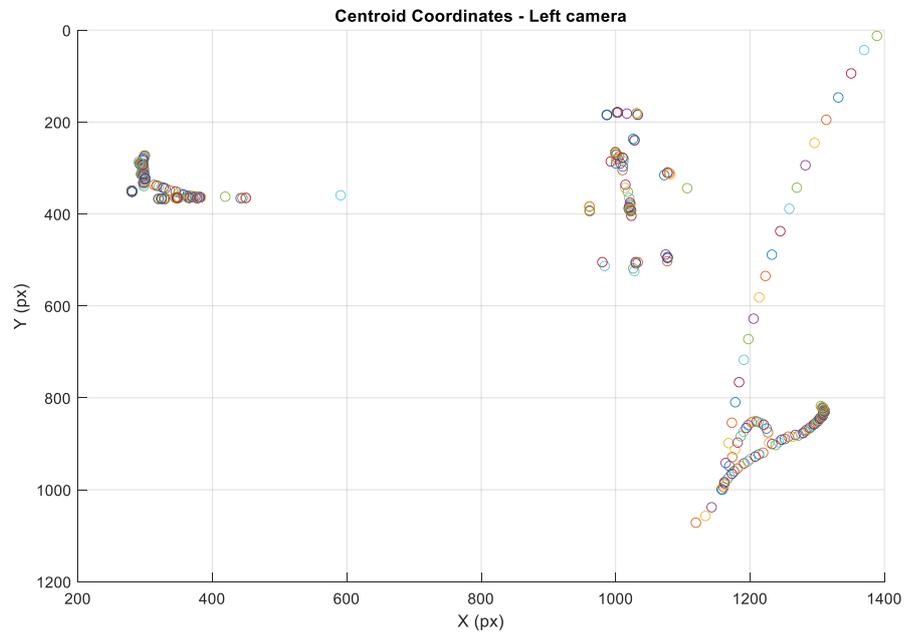


Figure 36: 2D representation of the centroids coordinates - Left camera

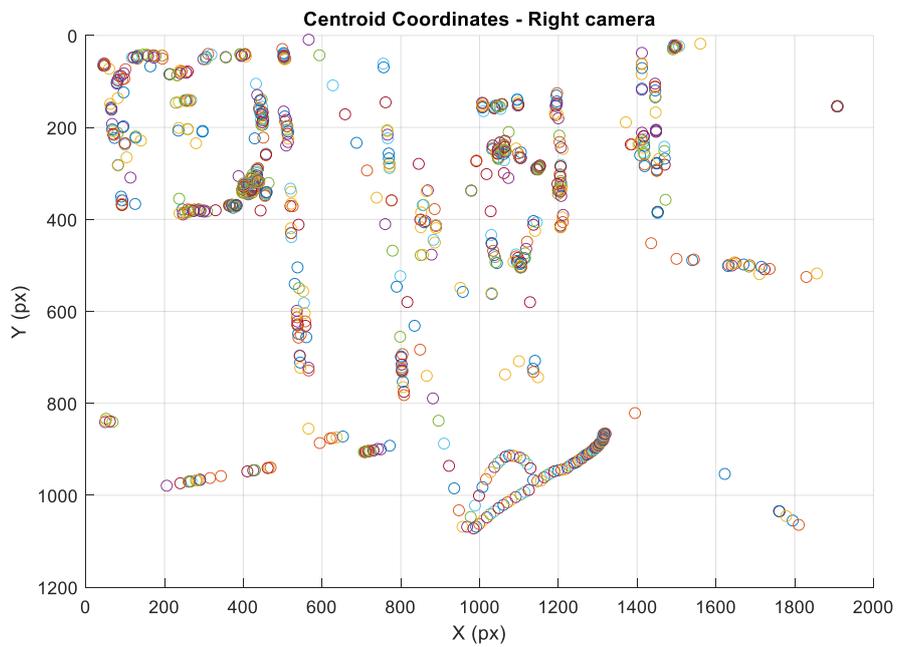


Figure 37: 2D representation of the centroids coordinates - Right camera

In the figures above (Figure 36 and Figure 37), the representation of the coordinates of the centroids is shown. Each circle represents the position of the object, through the x-y coordinates (in pixels) of its centroid, in each frame.

One may notice a well recognisable trajectory in 2D appears in the images together with some other points that can be considered as outliers (more evident in Figure 36). To represent only the path in which we are interested in, the derivative filter has been applied. It consists of a function able to filter the undesired points. Once defined a starting point (which belong to the path), considered as a reference point, the distance between it and the centroids in each frame has been computed. Moreover, the centroids' derivative in relation to the starting location has been calculated. With this procedure, it is possible to determine whether a point belongs or not to the trajectory by comparing the distances and the derivatives with predetermined thresholds. Finally, based on the chosen point and centroid of the current frame, the reference point is updated and all the method is iterated through the frames.

The figures below (Figure 38 and Figure 39) represent only the points that belong the trajectory of the object.

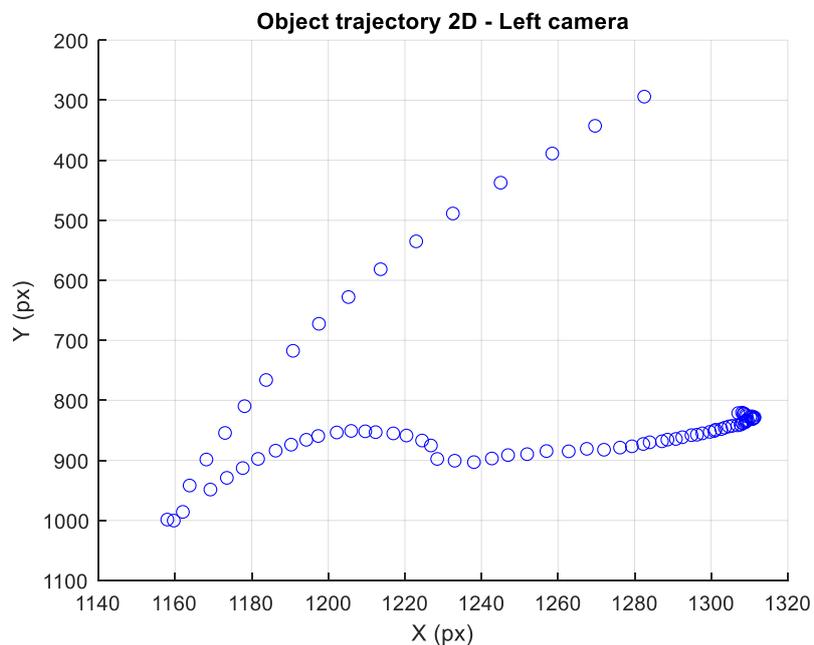


Figure 38: Object trajectory in 2D by the left camera

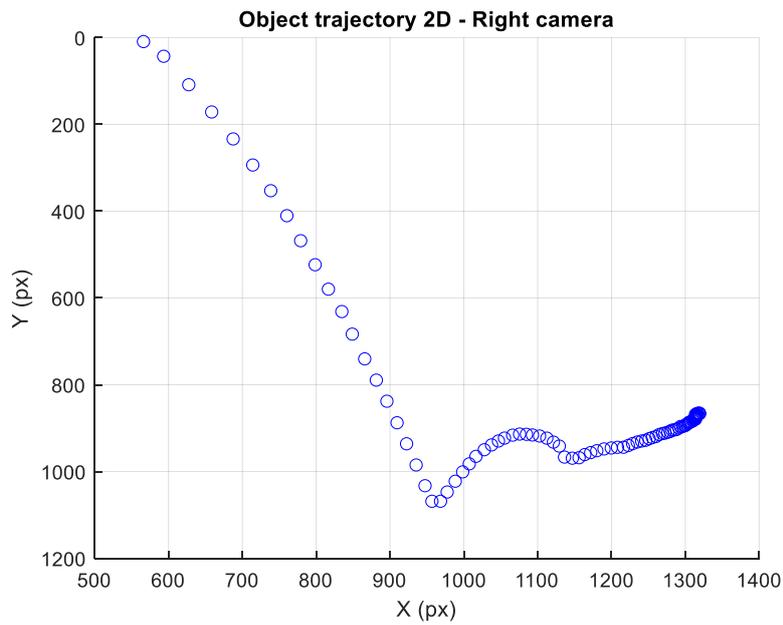


Figure 39: Object trajectory in 2D by the right camera

At the end, through the information coming from calibration process and the *triangulate* MATLAB function, the reconstruction of the trajectory has been possible in the real coordinate system. In the figure below (Figure 40), the object trajectory in 3D has been plotted, together with the position of the cameras.

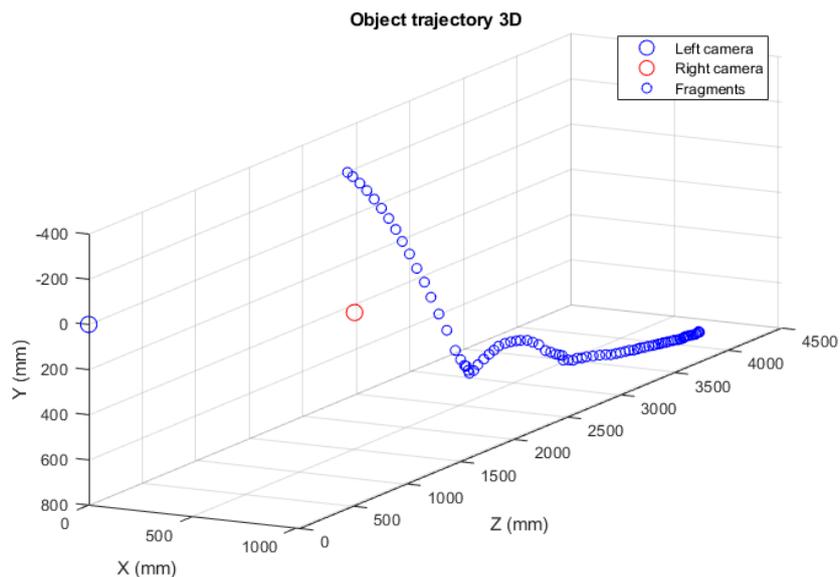


Figure 40: Trajectory of the object in 3D

To be sure that the 3D coordinates obtained by the analysis are correct, the first image point of each image used to calibrate the cameras has been represented in the real space. If this reprojected point coincides with its real position in the calibration patterns, it means that the method used to reconstruct the real coordinates of an object works well.

Moreover, the trajectory has been also represented in following figure (Figure 41), using the extrinsic parameters visualization, coming from the calibration process, to visualize the position of the cameras and the calibration patterns together with the trajectory of the object.

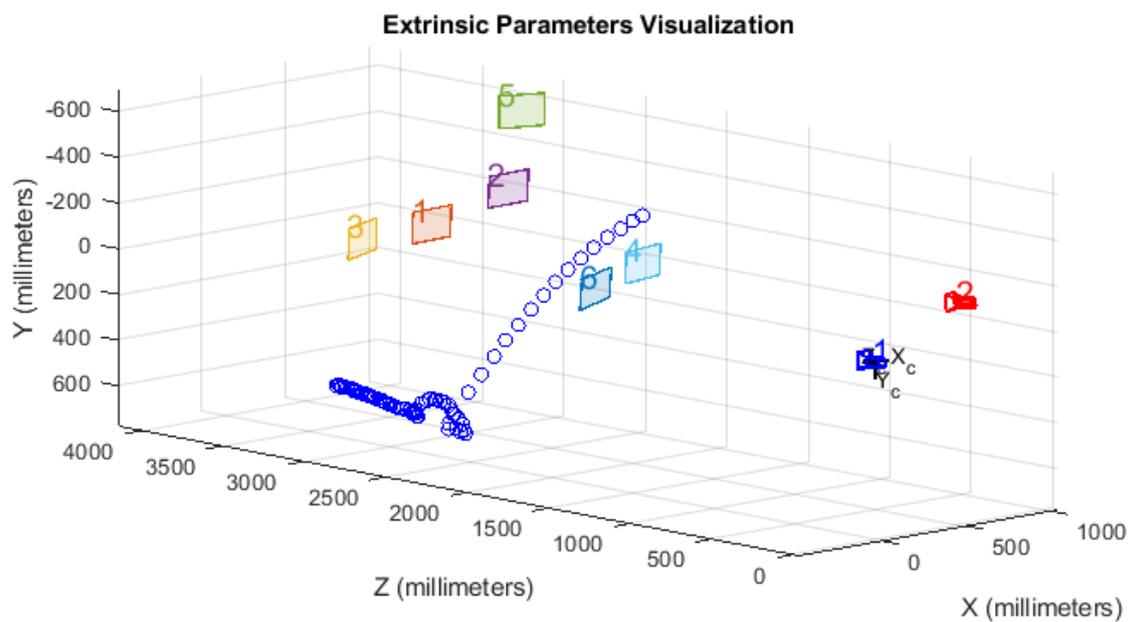


Figure 41: 3D trajectory of the object with cameras and chessboards locations

## 6. Field trial

After having done all the trials in laboratory, the analysed methodology has been tested in the field, in Valdilecha Quarry, to see if the proposed approach is effective also in a real environment. In contrast to laboratory tests, which are conducted under carefully controlled conditions, field trials require taking videos in real environments, which may provide a number of difficulties and unknowns due to illumination, moving objects and complex geometries. The main goal of a field trial is to assess the precision, quality, and viability of the method in practical settings.

Field tests also offer a chance to detect and solve practical and technical issues found during the data collection and processing phases, as problems with image quality, image distortion, and the effects of environmental factors. In this sense field trials provide essential information on the advantages, disadvantages, and potential areas for improvement, which helps to advance in the methodology. In conclusion, field tests help to improve the approaches, workflows, and tools used in photogrammetric analysis by highlighting the difficulties and constraints that can occur in complicated real contexts.

### 6.1 Features of the bench

The bench under analysis is the number 3 and in the following table (Table 9) its geometrical parameters and blast data are listed.

Table 9: Geometrical parameters of the bench under investigation

<b>Bench 3 - Blast from 12.06.23</b>		
<b>Parameters</b>		<b>Units</b>
Number of holes	26	-
Number of rows	4	-
Number of holes per row	6-7	-
Height of the bench	13	m
Length of the bench	49	m
Burden	5	m
Spacing	7	m
Stemming	3	m
In-row delay	42	ms
Among-rows delay	67	ms

The type of explosive employed is RIOFLEX, a bulk water gel based on ammonium nitrate, produced by MAXAM. Its density ranges between  $0.6 \text{ g/cm}^3$  and  $1.35 \text{ g/cm}^3$ . Thanks to its

variable density, it is considered very flexible and adaptable to many applications and different environments. The detonation has been performed through electronic detonators with an in-row delay of 42 ms and among-rows delay of 67 ms (Maxam, 2023).

The figure below (Figure 42) represents the design of the blast together with the location of a geophone, highlighted with a red circle.

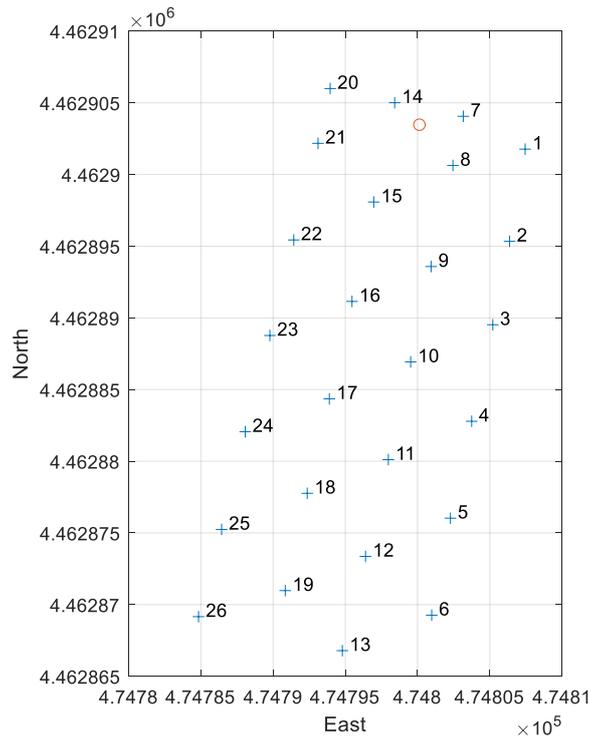


Figure 42: Design of the blast of the bench n.3

## 6.2 Placement of the cameras

According to the methodology proposed in this work, the cameras has been placed in the optimal position, as shown the figure below (Figure 43).

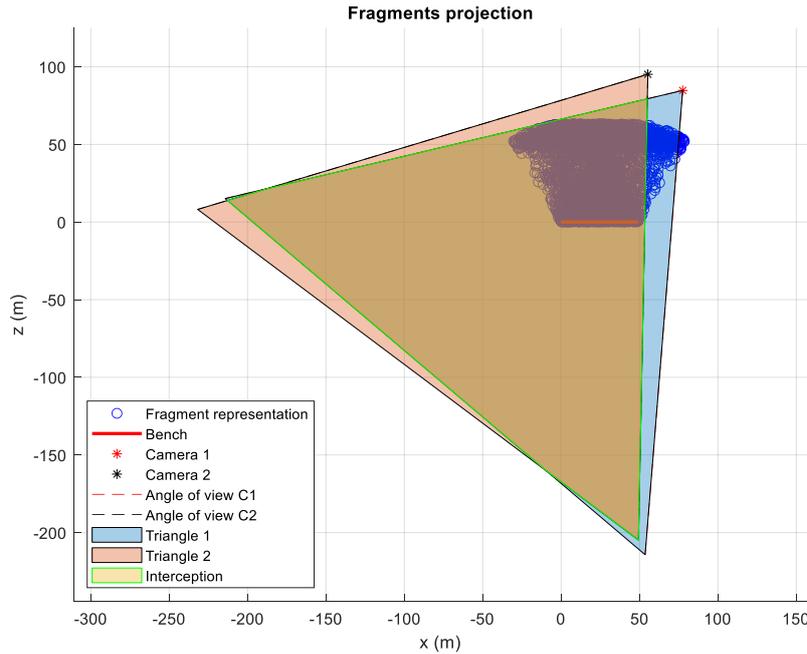


Figure 43: Optimal placement of the cameras in the field

The figure above (Figure 43) shows the cameras, the bench, the fragments, and the fields of view of the cameras are represented by two triangles.

The *Camera 1* is depicted in red and located at an angle  $\vartheta = 58^\circ$ . The second camera (*Camera 2*) is shown in black and positioned at an angle  $(\beta + \vartheta) = 72^\circ$ , with a distance of 24.55 m from the first camera. The distance between the centre of the bench and the cameras is about 100 m. This value is the minimum distance (lower bound) at which the cameras should be placed.

To determine the angle of view and orientation of each camera with respect to the horizontal plane, two triangles are used. The first camera has an orientation of about  $229.42^\circ$  ( $\omega_1$ ), while the second camera has an orientation of  $232.86^\circ$  ( $\omega_2$ ). These orientations indicate the direction in which the cameras are pointing.

The triangles represent the visible area on the ground from each camera's perspective. When these triangles intersect, they form a third polygon. This polygon depicts the area on the ground that is visible from both the cameras. It has five vertices and represents the maximum possible overlapping area, encompassing the largest number of fragments.

A preliminary inspection of the site was conducted on 25<sup>th</sup> of May (2023) to understand how the configuration of the place was and where the equipment could be placed to obtain the best result.

In the figures below (Figure 44), an overview of the part of the quarry under investigation is represented and on the left side of the image it is possible to distinguish the bench.

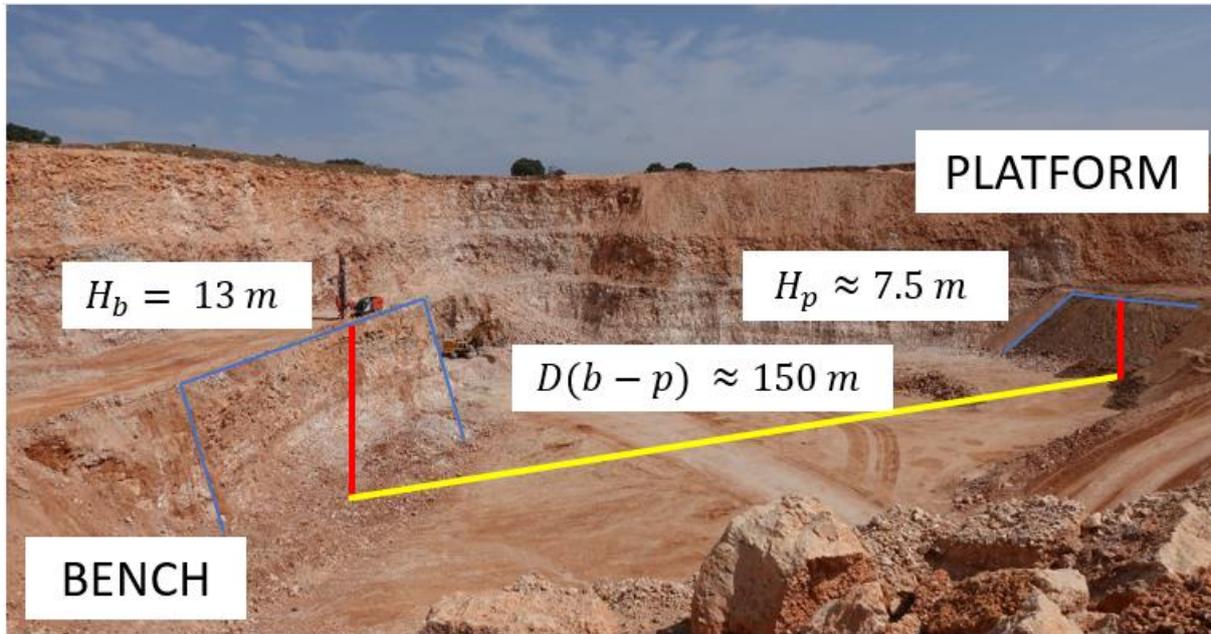


Figure 44: Overview of the site

As it is possible to notice in Figure 44, in front of the bench there is a platform with an altitude of 7.5 m, where the cameras can be easily placed. The distance between the bench and the plateau is about 150 m. It has been preferable to settle the cameras on the plateau and not on the ground for safety reasons to prevent damages to the cameras.

In the images below (Figure 45 and Figure 46), the setup of the equipment on the platform and a zoom in of the cameras are shown.



Figure 45: Setup of the equipment



Figure 46: Zoom in of the location of the cameras

The cameras have been placed at a distance of about 150 m from the bench, 20 m far one from the other about. These values are different from the ones found theoretically. The difference is due to the configuration of the land. Moreover, the orientations of the cameras were not perfectly equal to the ones found with the software, because of the difficulty of reaching those precise measurements ( $\omega_1$  and  $\omega_2$ ). In the table below (Table 10), the different values are reported.

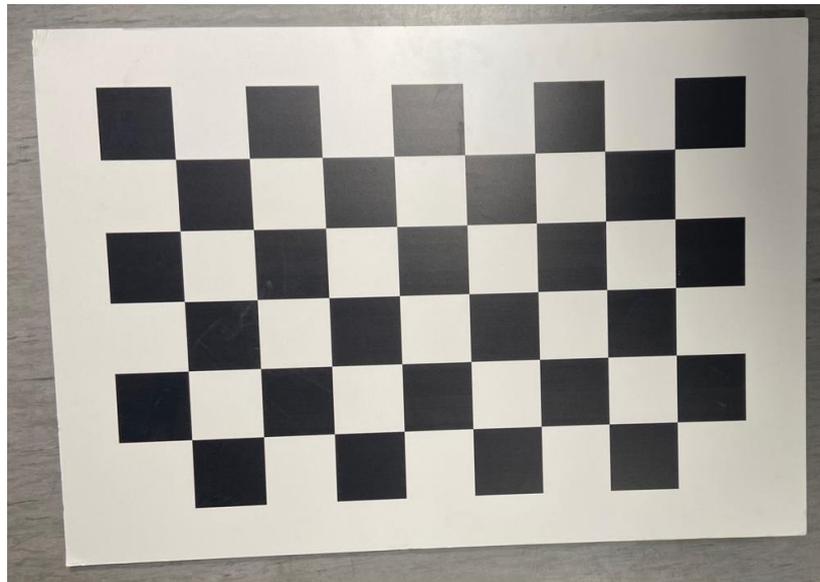
Table 10: Comparison between the theoretical and in-field values

	Theoretical values	Value used in the field
$\omega_1$	229.42°	225°
$\omega_2$	232.86°	230°
$D_{C1-C2}$	24.55 m	20 m
$D_{Bench-C}$	100 m	150 m

### 6.3 Calibration process

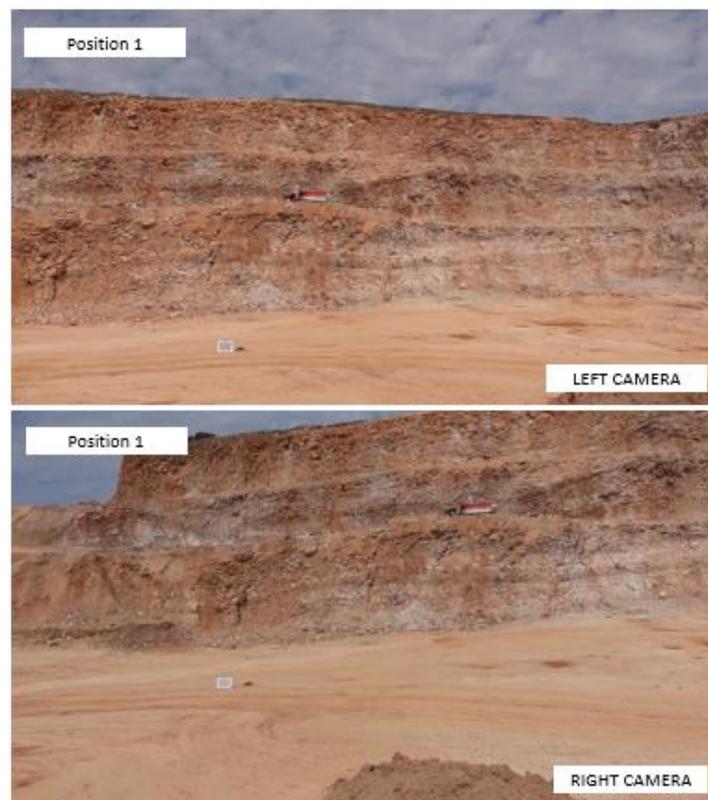
Since the distances in the field are much greater than those used in the laboratory, we decided to adopt a larger chessboard for the calibration so that it could be more easily seen from both cameras.

As shown in figure below (Figure 47), the calibration pattern is a 9-by-6 chessboard with 78 mm side squares.



*Figure 47: Calibration pattern used in the field*

To perform the calibration, the chessboard has been placed in 10 different positions. The following images (Figures 48 and Figures 49) depict examples of different locations of the calibration pattern.



*Figure 48: Example of calibration, chessboard in position 1*

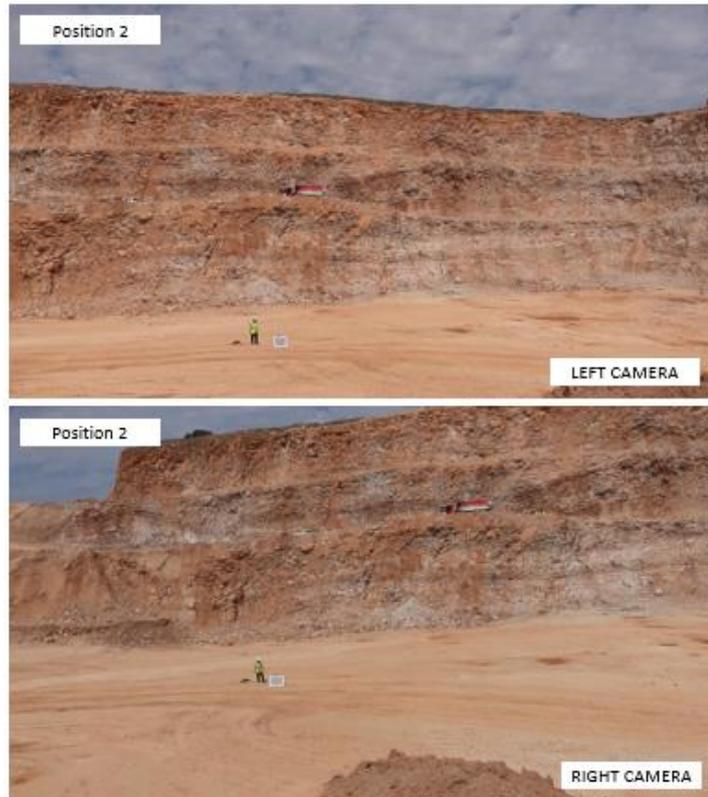


Figure 49: Example of calibration, chessboard in position 2

Through the above-mentioned process, the calibration was done as shown in the figures below (Figure 50 and Figure 51).

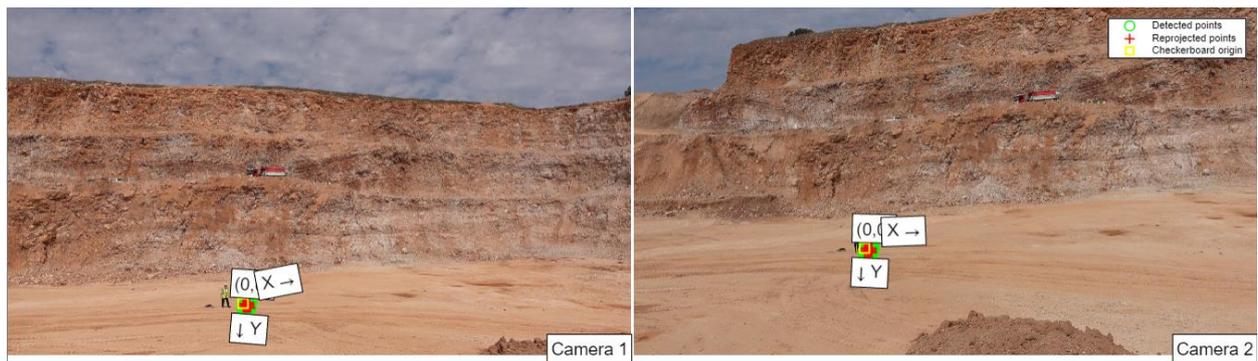
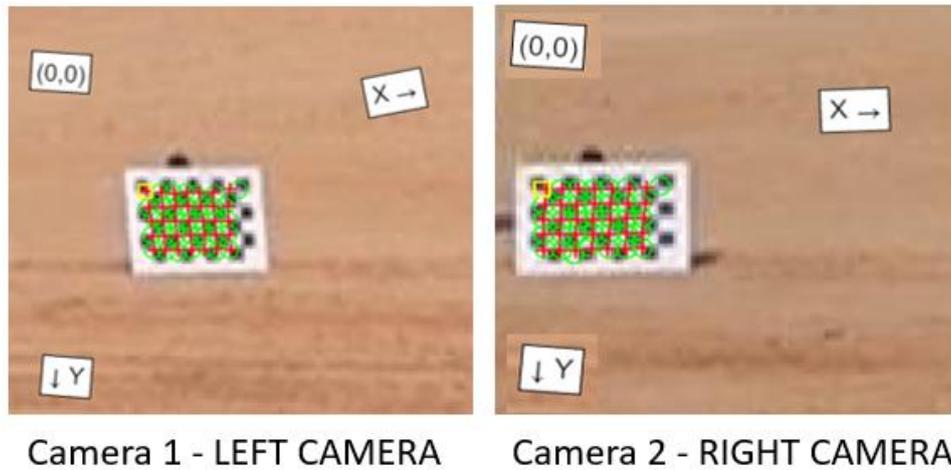


Figure 50: Stereo calibration result, with the chessboard in position 2



*Figure 51: Zoom in of the stereo calibration result*

As it is possible to see, the stereo calibration is not very accurate. Not all the corners of the squares are detected probably because the distance between the pattern and the cameras was too high, and the image results to be pixelated and blurred.

#### **6.4 Blast recordings**

Once the calibration videos were recorded and the holes were charged, the blast took place. It lasted 4 seconds. In the following figures (Figure 52 and Figure 53), some representative frames of the blast motion are shown.

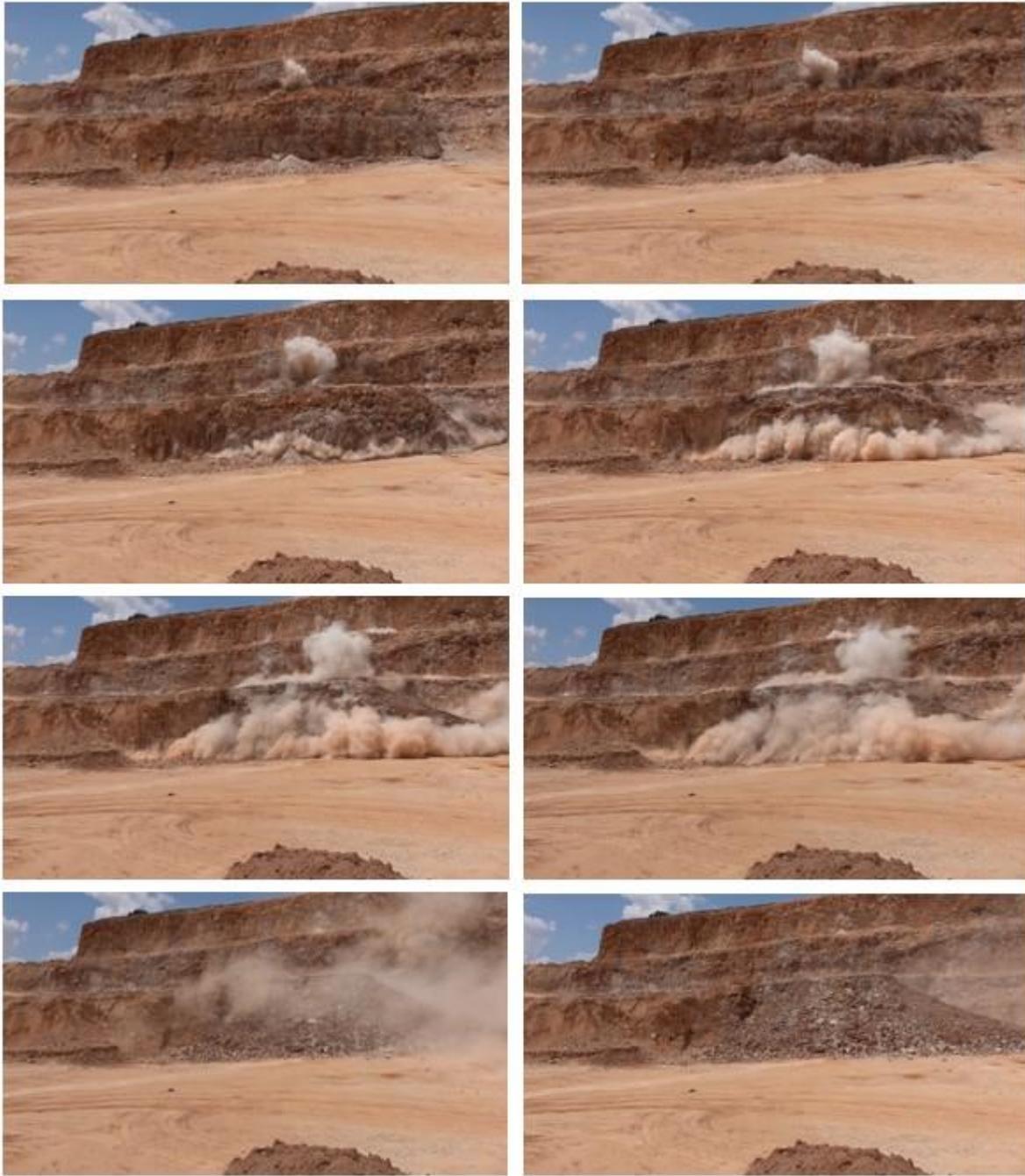
As it is possible to notice from the images, the blast produced a great amount of fumes and dust, which made it challenging to identify the trajectory of the individual fragments from that distance, as sometimes they were hidden by the dust. However, the motion based multi-object algorithm has been applied in both recordings and, as it has been predicted, all the fragments were seen by a unique moving cloud and no single paths were recognisable.

LEFT CAMERA



*Figure 52: Example of representative frames of the blast (Left camera)*

RIGHT CAMERA



*Figure 53: Example of representative frames of the blast (Right camera)*

## 7. Discussion

In this first part of this work a criterion is proposed to choose two identical cameras for the reconstruction of the three-dimensional movement of the blast face fragments. This is done through a motion analysis of the blast face, whose aim is to determine whether a moving object may be captured by the cameras at a specific distance from its source and where the cameras have to be placed to get the optimal results.

Employing equations of motion, the final position of each fragment on the ground and its distance from the bench have been evaluated starting from the geometry of the site. Then, the optimal location of the cameras is defined through the multi-objective optimization. Both the cameras are placed in front of the bench, oriented according to specific angles, and distant from each other of a certain value.

In this optimal configuration, both cameras can record the motion of the greatest amount of fragments at the same time.

At the end, the existing link between the outcomes from the design of the site and the features of the cameras has been carried out in detail. The fragment size, the duration of the blast and the distance between the cameras and the bench (which depends on the position of the cameras on the ground) have been combined together with the focal length and the pixel size of the cameras. By the comparison of the performances of the cameras analysed viz. Sony DSC-RX100 VII, Sony ZV-1, Sony RX10 III, Canon G5X Mark II, the most suitable camera for our research has been chosen.

In conclusion, the camera Sony DSC-RX100 VII is preferable to the other cameras considered, because it has:

- The highest shooting time, able to record the whole blast.
- A reasonable number of pixels able to capture a fragment with a diameter of about 59 cm, from a distance of 153.3 m.
- High frame rate values in HFR mode, necessary to have a record with good quality.

The following part of the work is focused on the use of photogrammetric analysis to be able to reconstruct the motion of the fragment in a real-world coordinate system. The proposed methodology has been initially tested in laboratory trials showing effective results, and subsequently applied in field, where numerous issues and challenges has been encountered. As it usually happens, when you move from theory (in laboratory) to practice (on the field) there could be different results between what has been predicted and what really happens for many reasons.

## 7.1 Issues

The placement of the cameras was not properly the one defined theoretically because of the influence of the shape of the site. The most relevant feature to underline is the distance between the bench and the cameras. For safety reasons, the cameras have been placed in front of the bench, on a platform 150 *m* far from the bench. The choice not to place the cameras on the ground (on the bottom of the quarry) has affected the result of the analysis both of the calibration and the fragments videos.

### *Calibration issues*

The calibration pattern was too far from the cameras because it appears to be blurred and pixelated, so that the recognition of the corners of the squares is difficult to identify and sometimes not possible, as shown in the Figure 54, from the light blue circle. The calibration pattern was a 9-by-6 squares chessboard and in this case, the board size evaluated with the algorithm has been instead 8-by-6 squares, resulting in a decrease in the accuracy of the method. Moreover, the detected points, indicating the corners of the squares and that should lie on the same line, appear to be misaligned. This can affect the radial distortion values, which order of magnitude is bigger than what was expected ( $10^3$  instead of  $10^{-2}/10^{-1}$ ).

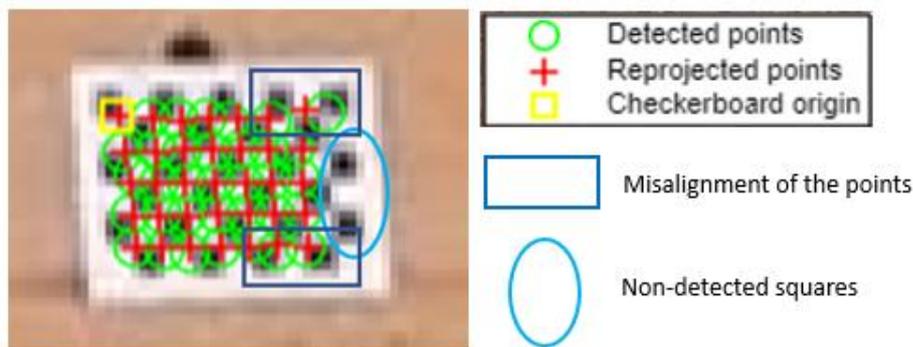


Figure 54: Example of calibration issues

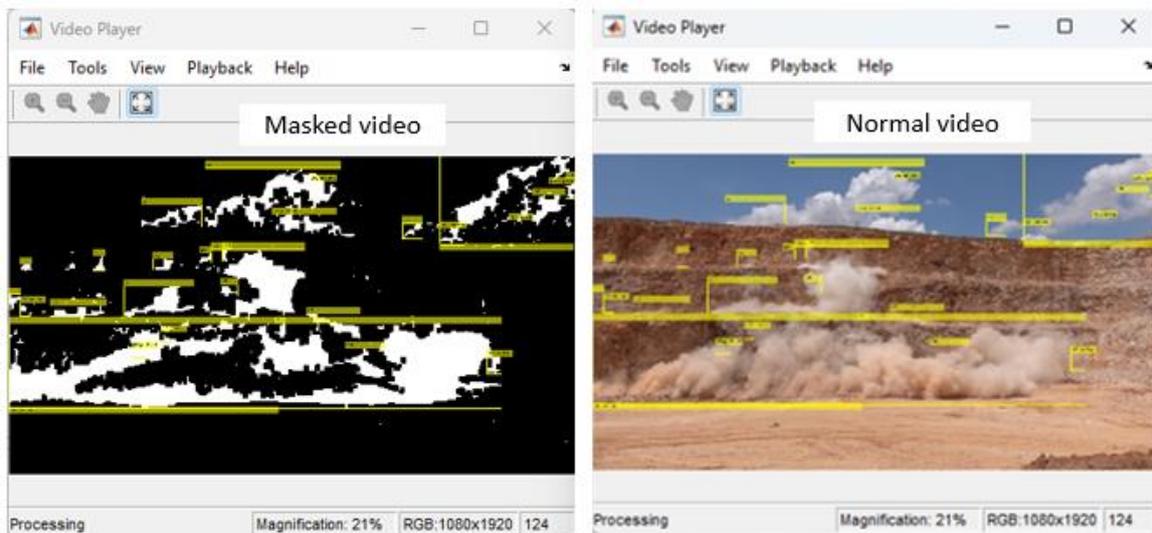
Furthermore, problems in calibration have also affected the correction due to lens distortion. The MATLAB function *undistortImage.m* takes as input the stereo parameters (as for example the rotation matrixes, the translation vector, the focal lengths of the cameras and the radial distortion values) coming from the calibration, returning as output the modified undistorted image. For this reason, if the variables from the calibration are inaccurate, the error also affects the lens distortion correction.

### *Trajectory recognition issues*

After the calibration, the next step is the recognition of the motion of the fragments (represented by centroids) in the videos, by analysing the coordinates of the centroids in each frame. The procedure proposed in the laboratory test has been applied to the recordings taken on the field, but unfortunately in the latter case the multi-object motion-based algorithm was not able to detect the path of individual fragments. All the fragments have been seen as a unique moving cloud. The figures below (Figure 55 and Figure 56) show examples of the application of the motion based multi-object algorithm applied on two frames.

As it is noticeable in the first image (Figure 55), most of the fragments are enclosed in the same box, which is indicating the location (x and y coordinates in pixels) of the object identified by the algorithm. For this reason, the individual recognition of a single fragment trajectory was not possible under the conditions employed. Moreover, the amount of dust produced by the blast was quite high, resulting in covering the motion of the rock fragments.

In the second figure (Figure 56), the influence of the dust is even more evident as it is possible to see from the yellow boxes around the clouds of smoke, which indicate their movement.



*Figure 55: Application of the motion based multi-object algorithm in frame 124 (Left camera)*

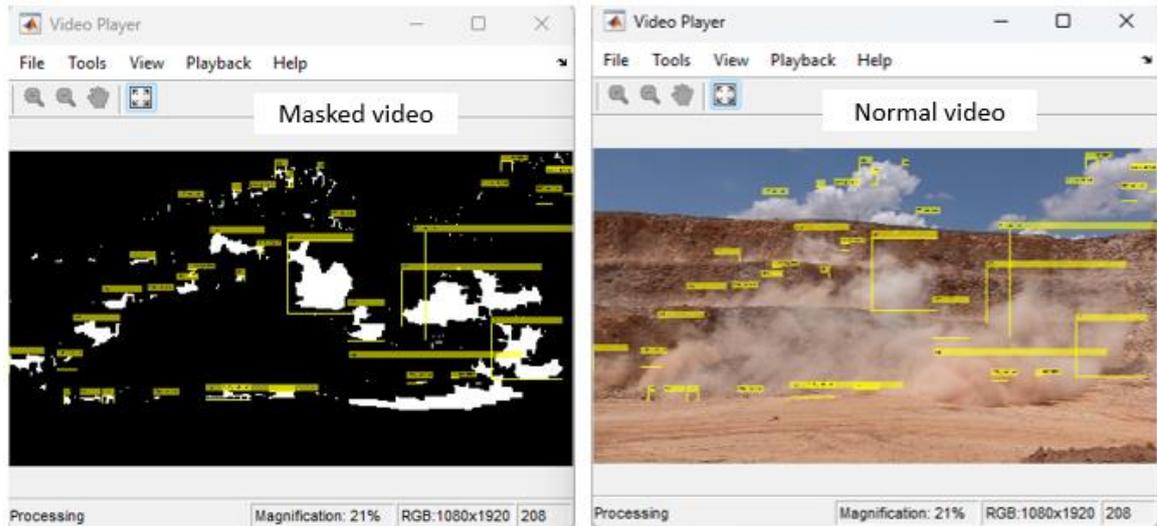


Figure 56: Application of the motion based multi-object algorithm in frame 208 (Left camera)

## 7.2 Recommendations

To try to avoid these problems, it is necessary to make some changes and improvements to the procedure.

### *Calibration improvements*

To improve the calibration results, there could be some enhancements and changes to do: modifying the placement of the cameras or moving the calibration pattern closer to them.

If the cameras are kept in the position described, maybe a larger calibration pattern is needed. However, a larger pattern may cause problems in transport and handling, despite the improvements that could be achieved in calibration results. For this reason, there could be two possibilities to better the experiments:

1. to approach the calibration pattern to the cameras or
2. to approach the cameras to the chessboard.

In this way it is possible to avoid pixelated and blurred images of the chessboard, so that the corners of each square will be well visible.

### *Trajectory recognition improvements*

To reconstruct the trajectory of the fragments through the use of the motion based multi-object algorithm, a solution can be to focus only on a part of the whole bench and to put the camera as close as possible to the bench, obviously always within the limits imposed by safety regulations. This implies that the cameras will not be placed on the plateau in front of the bench, but on the

ground. By placing the cameras on the ground, the distance can be decided based on the specific needs and requirements, rather than being imposed by the site's morphology.

Although the fragments detection can be easier, placing the cameras closer to objects of interest increases the potential damage. In this case, it might be appropriate to take some precautions, as for example covering them with some kind of protection, if possible. Moreover, a manual recognition system can be used. It involves the manually detection of the objects frame by frame, instead of the application of an automatic algorithm.

Furthermore, the videos can be recorded by cameras settled on two small drones (instead of the ground), that should remain fixed at a certain height for all the duration of the blast. This new setup can maybe help in the recognition of the fragments' motion, offering new branches of investigation.

By applying these changes into practice, the calibration procedure will achieve greater accuracy and the motion-based multi-object tracking system will improve the trajectory reconstruction, enhancing the performance and yielding better outcomes.

## **8. Conclusion**

As a development of the TRL 3 (*Concept validation*), this works aims to reach the TRL 4 (*Experimental pilot*). However, the final result coming from the analysis of the field videos has proved to be different from what had been predicted. The change from laboratory analysis to field test and condition has shown some problematic features to be considered.

Environmental changes, unforeseen variables, and practical limitations can all have an impact on how well the operation works and how it turns out. Despite certain challenges, these situations provide invaluable insights and chances for improvement. These enhancements might involve adjusting the experimental design, resolving technical problems, improving the data collection and analysis procedures to better fit real-world circumstances.

Notwithstanding the unsuccessful results of the field trial, the work has significantly improved a number of crucial areas, as for example camera location, camera selection for object tracking, automatic fragment object movement recognition, calibration, and 3D reconstruction of object trajectories.

The suggested strategy for introducing cameras in a blast provides a way to choose the best cameras and their optimal placement, based on statistical models as the use of the multi-objectives optimization. The camera selection process results to be affected by the object tracking method, a crucial component of the proposed methodology. A proper detection algorithm makes sure that the

chosen cameras can record the blast, ensuring the extraction of essential data for an accurate object tracking and trajectory analysis. The adopted technology uses computer vision algorithms to automatically identify and track moving objects in video frames, greatly affecting (positively in the lab tests and negatively in the field) the efficiency and precision of trajectory analysis.

Moreover, a calibration methodology has been a necessary step to reconstruct the 3D path of a moving object. By calibrating the cameras, an accurate connection between the image plane and the real coordinate system is achieved. According to the accuracy of this process, it is possible to rebuild the trajectory of a moving object in three dimensions, giving an exhaustive comprehension of fragment movement.

Overall, the passage between field testing and laboratory trials marks a significant development and validation milestone for the method. The difficulties encountered during this shift are important teaching moments that help iteratively improve the process for subsequent deployments. Based on the methodology proposed in this work, the process can be further refined to produce more accurate and dependable outcomes in real-world circumstances through ongoing improvement and adaptation.

## Bibliography and references

- Adil, E., Mikou, M., & Mouhsen, A. (2022). A novel algorithm for distance measurement using stereo camera. *The Institution of Engineering and Technology*.
- Afum, B., & Temeng, V. (2015). Reducing Drill and Blast Cost through Blast Optimisation - A Case Study. *Ghana Mining Journal*, 15(2), 50–57.
- Alsadik, B. S. A. (2014). *Guided close range photogrammetry for 3D modelling of cultural heritage sites. Dissertation*.
- Ascher, S., & Pincus, E. (1999). *The Filmmaker's Handbook: A Comprehensive Guide for the Digital Age* (3rd ed.). A Plume Book.
- Balch, K. (1999). High Frame Rate Electronic Imaging. *Motion Video Products*.
- Benro EN-GBP. (n.d.). *Benro Slim Tripod Kit - TSL08CN00* .
- Blair, B. E. (1960). Use of high-speed camera in blasting studies. *US Department of the Interior, Bureau of Mines*, 5584.
- Bowa, V. M. (2015). Optimization of Blasting Design Parameters on Open Pit Bench. A Case Study of Nchanga Open Pits. *Int. J SciTechnolRes*, 4(9).
- Božić, B. (1998). Control of fragmentation by blasting. *Rudarsko-Geološko-Naftni Zbornik*, 10(1).
- Brown, D. (1966). *Photogrammetric Engineering - Decentering distortion of lenses* (Vol. 32).
- Brunner, D. (n.d.). *Frame Rate: A Beginner's Guide*.
- Cabrera-Quirós, L., Campos-Gómez, R., & Castro-Godínez, J. (2013). *Critical steps in camera pose estimation: an evaluation using LTI-LIB2 library*.
- Canon. (2019). *PowerShot G5X Mark III Advanced User Guide*. Canon Inc.
- Chiappetta, R. F., & Mammale, M. E. (1987). Analytical high-speed photography to evaluate air decks, stemming retention and gas confinement in presplitting, reclamation and gross motion applications. *In Proc. of the 2 Nd Int. Symp. on Rock Fragmentation by Blasting*.
- Custodio, A. L., Madeira, J. F. A., Vaz, A. I. F., & Vicente, L. N. (2011). Direct multisearch for multiobjective optimization. *SIAM Journal on Optimization*.
- D'Errico, J. (2006). *Inhull MATLAB function*.
- European Technology Platform on Sustainable Mineral Resources (ETP SMR). (2016). *VERAM – Vision and roadmap for European raw materials*.
- European Union. (2021). *101003750-DIGIECOQUARRY-Part B*.
- Franco, M. P. (2022). *Video analysis of the blast face at the El Aljibe Quarry (Master Thesis)*. Universidad Politecnica de Madrid.

- Fryer, J., & McIntosh, K. (2001). Enhancement of Image Resolution in Digital Photogrammetry. *PHOTOGRAMMETRIC ENGINEERING & REMOTE SENSING*.
- Fujifilm. (n.d.). *Digital Camera X-T4 Owner's Manual*. Fujifilm Corporation.
- Handayani, H. (2010). Space intersection by collinearity. In *Geoid Volume* (Vols. 05, No.02). Institut Teknologi Sepuluh Nopember.
- Hanson Hispania. (2020). *Participating quarry information - The quarry life award*.
- Hartley, R. I., & Sturm, P. (1995). Triangulation. In V. Hlaváč & R. Šára (Eds.), *Computer Analysis of Images and Patterns. CAIP 1995. Lecture Notes in Computer Science, vol 970*. Springer.
- Héder, M. (2017). From NASA to EU: the evolution of the TRL scale in Public Sector Innovation. *The Innovation Journal: The Public Sector Innovation Journal*, 22(2).
- Hunter, R., Chu, M., Hoang, T., Hyatt, J., & Baloch, G. A. (2022). CAMERA TRIGGERING AND MULTI - CAMERA PHOTOGRAMMETRY . *Patent Application Publication*.
- Hwang, C. L. (1979). *Multiple objective decision making, methods and applications: a state-of-the-art survey*. Berlin ; New York : Springer-Verlag.
- Instituto Geografico Nacional. (2008). Grupo A2 - Tema 28: Fotogrametria y Teledeteccion. *Temario Basico y Bibliografia Recomendada Para La Preparacion de La Oposicion al Cuerpo de Ingenieros Geografos*, 67–85.
- Jiménez, J. H. (2022). *Methodologies for the reconstruction of fragment trajectories ejected from blasting. 2D and 3D approaches* [Proyecto fin de Grado]. Universidad Politecnica de Madrid.
- Joseph, C. (2022). *What is HFR (High Frame Rate) and why should you care?*
- Karim, A. (2000). *Multi-Objective Optimization Techniques*.
- Maxam. (2023). *RIOFLEX MX 10000, Mechanically sensitized Cross-linked Bulk Watergel Explosive*.
- Moser, P., & Gaich, A. (2007). 3D Stereophotogrammetry for blast design and control. In *Vienna Conference Proceedings 2007*, 3–11.
- Navarro, J., Segarra, P., & Castedo, R. (2018). *High speed video recording to determine rock movement*.
- Nikon Inc., Berkenfeld, D., Black, D., Corrado, M., & Silverman, L. (2023). *Understanding Focal Length*.
- Onederra, I., & Esen, S. (2003). Selection of inter-hole and inter-row timing for surface blasting—an approach based on burden relief analysis. *Proceedings of the 2nd World Conference on Explosives and Blasting Technique*, 10–12.
- Over, J.-S. R., Ritchie, A. C., Kranenburg, C. J., Brown, J. A., Buscombe, D., Noble, T., Sherwood, C. R., Warrick, J. A., & Wernette, P. A. (2021). *Processing Coastal Imagery With Agisoft*

*Metashape Professional Edition, Version 1.6— Structure From Motion Workflow Documentation.*

- Ozkahraman, H. T. (2005). Fragmentation assessment and design of blast pattern at Goltas Limestone Quarry, Turkey. *International Journal of Rock Mechanics and Mining Sciences* (2006), 43, 628–633.
- Palangio, T. W., Palangio, T. C., & Maerz, N. (2005). *Advanced automatic optical blast fragmentation sizing and tracking.*
- Segarra, P. (2004). Chapter 4: Measurement and analysis of the bench face movement . In *Análisis experimental de la fragmentación, vibraciones y movimiento de la roca en voladuras a cielo abierto.* (pp. 24–47).
- Sony. (2016). *Digital Still Camera DSC-RX10M3 How to Use.* Sony Corporation.
- Sony. (2020). *Interchangeable Lens Digital Camera ILCE-α7SIII Help Guide* Sony. Sony Corporation.
- Sony. (2022). *Interchangeable Lens Digital Camera ILME-FX30 Help Guide.* Sony Corporation.
- Stojadinović, S., Pantović, R., & Žikić, M. (2011). Prediction of flyrock trajectories for forensic applications using ballistic flight equations. *International Journal of Rock Mechanics and Mining Sciences*, 48(7), 1086–1094.
- TEC-SHS. (2008). *TECHNOLOGY READINESS LEVELS HANDBOOK FOR SPACE APPLICATIONS* (1st ed.). ESA.
- Versus. (2016). *Pixel size (main camera).*
- Wang, Z. (1990). Principles of photogrammetry (with remote sensing). In *Principles of photogrammetry (with remote sensing).* Publishing House of Wuhan Technical University of Surveying and Mapping.
- Wolf, P. R., Dewitt, B. A., & Wilkinson, B. E. (2014). *Elements of photogrammetry with applications in GIS* (McGraw-Hill Education, Ed.; 4th ed.).

## Appendix A

### % Location of the cameras

```
clc
clear all
close all

%% Data from BANCO 2E - Blast of 01.07.22

nhtot = 24;      % Tot num of holes per blast
nrows = 2;      % num of rows
h1r = 12;       % num of holes per row
B = 5.75;       % Burden (m)
Sp = 7;         % Spacing (m)
H = 14.2;       % Height (m)
T = 3;         % Stemming (m)
W = h1r*Sp;     % Width (m)
tir=25;        % Inter-row delay (ms)

passing = [0.2 0.25 0.3 0.35 0.4 0.45 0.5 0.55 0.6 0.65 0.7 0.75 0.8 0.85 0.9 0.95 1];
% in %
frag_size = [50 60 100 140 160 200 250 270 320 350 500 600 700 900 1050 2000 4000];
% in mm
s = 50:.001:4000;
length(s);
pp = pchip(frag_size,passing,s);

xholes = ((Sp/2):Sp:(W-(Sp/2)))';
yholes = H;
L=length(xholes);

figure(1)
rectangle('Position', [0 0 W H])
hold on
xline(xholes)
hold on
% scatter(xc,yc,'r')
grid on
xlabel('Width (m)')
ylabel('Height (m)')
title('Bench representation')
hold off

% passing=[0 passing];
% frag_size=[0 frag_size];
nfrag=5000;
rsizes=[];
ppr = @(x) pchip(passing,frag_size,x);
while numel(rsizes)~=nfrag
    fragpass=rand(1,1);
    if fragpass>=0.2
        rsizes=[rsizes ppr(fragpass)]; % mm
    end
end
f_radius=rsizes'/2000;
```

```

% Coordinate of a generic fragment on the bench
ry = rand(1,nfrag)*H;
rx = rand(1,nfrag)*W;
[mdis,pos] = min(abs(rx-xholes),[],1);
tdis1=(rx-xholes);
tdis2=zeros(1,nfrag);
for jj=1:nfrag
    tdis2(jj)=tdis1(pos(jj),jj);
end
nabs = atan(B./mdis);
n2 = atan(B./tdis2);
dir=sign(n2);
Bf = sqrt(mdis.^2+B^2);
mazim = mean(nabs);
Mean_radius=mean(f_radius);

figure(2)
semilogx(frag_size,passing,'o')
% hold on
% plot(fragpass, rsize, '-')
xlabel('Size (mm)')
ylabel('Passing (%)')
title('Grain size distribution')
grid on
hold off

% Constants and equations known for each instant and fragment
g = 9.81; % Gravity m/s2
ad = 1.1614; % Air density kg/m3
S = pi*f_radius.^2; % Surface of fragment m2.
m = 2721*(4/3)*pi*f_radius.^3; % Mass of fragment
mu = 18.46e-6; % Dynamic air viscosity kg/(m*s)

t0 = 0;
tstep = 1e-2;

% Models: Response time coeff
t_resp = 23.063*exp(-1.013*Bf./ry);

% Initial velocity
maxv=25;
v0 = maxv*(8.7153*(ry/H).^4-17.139*(ry/H).^3+8.3769*(ry/H).^2+0.9364*(ry/H)-0.005);

% Ejection angle
alfa = 59.551*exp(-1.151*(Bf./ry));
v0x = v0.*cosd(alfa);
v0y = v0.*sind(alfa);

POS=cell(1,nfrag);
TIME=cell(1,nfrag);

P=zeros(numel(f_radius),3);
for ii=1:nfrag
    i=1; %Iteration.
    x=[];
    x=0;
    y=[];
    y=ry(ii);
    vx=[];
    vy=[];

```

```

ax=[];
ay=[];
v=[];
t=[];
t=t0+t_resp(ii)+tir*(pos(ii)-1);
t=t/1000;

while y(end)>0
    % i
    if i==1
        vx(i) = v0x(ii);
        vy(i) = v0y(ii);
        v(i) = v0(ii);

        Re = 2*ad.*f_radius(ii).*v(i)./mu;
        Cd = (24./Re)+((2.6.*(Re./5))./(1+(Re./5).^1.52))+((0.411.*(Re./26300).^(-
7.94))./(1+(Re./263000).^(-8)))+(Re.^0.08)./461000);

        fr = (0.5*ad.*Cd.*S(ii))./m(ii);           % Drag force
        ax(i) = -fr*vx(i)*v(i);
        ay(i) = -g-fr*vy(i)*v(i);

    else
        t(i)=t(i-1)+tstep;
        vx(i) = vx(i-1) + ax(i-1)*(t(i)-t(i-1));
        vy(i) = vy(i-1) + ay(i-1)*(t(i)-t(i-1));
        v(i) = sqrt(vx(i).^2+vy(i).^2);
        Re = 2*ad.*f_radius(ii).*v(i)./mu;
        Cd = (24./Re)+((2.6.*(Re./5))./(1+(Re./5).^1.52))+((0.411.*(Re./26300).^(-
7.94))./(1+(Re./263000).^(-8)))+(Re.^0.08)./461000);

        fr = (0.5.*ad.*Cd*S(ii))./m(ii);
        ax(i) = -fr*vx(i).*v(i);
        ay(i) = -g-fr.*vy(i).*v(i);

        x(i) = x(i-1)+vx(i).*(t(i)-t(i-1));
        y(i) = y(i-1)+vy(i).*(t(i)-t(i-1));

    end
    POS{ii}(i,:)=[rx(ii)+real(x(i))*cos(nabs(ii))*dir(ii) real(y(i))
real(x(i))*sin(nabs(ii))];
    i=i+1;
end
X(ii)=real(x(i-1));
Y(ii)=real(y(i-1));
T(ii)=t(end);
TIME{ii}=t;
P(ii,:)=[rx(ii)+X(ii)*cos(nabs(ii))*dir(ii) 0 X(ii)*sin(nabs(ii))];
end

XX=[0 W W 0];
YY=[0 0 0 0];
ZZ=[0 0 H H];

range=X;

[tmax,int]=max(T);
[m,inm] = min(f_radius);
tt=0:0.01:tmax;

```

```

Col=flip(jet(numel(tt)));
CT = interp1(tt,Col,T,'pchip');

figure(3)
hold on
for ii=1:10
    plot3(POS{ii}(:,1),POS{ii}(:,3),POS{ii}(:,2))
end
fill3(XX,YY,ZZ,'y','FaceAlpha',0.3)
%scatter3(P(:,1),P(:,3),P(:,2),10*radius*2,CT,'LineWidth',2)
xlabel('x (m)')
ylabel('z (m)')
zlabel('y (m)')
grid on
title ('Fragments trajectory')
xlim([-20 104])
ylim([0 100])
zlim([0 25])
set(gca, 'YDir', 'reverse' )

figure(4)
scatter3(P(:,1),P(:,3),P(:,2),10*f_radius*2,CT,'LineWidth',2)
set(gca, 'YDir', 'reverse' )
hold on
fill3(XX,YY,ZZ,'y','FaceAlpha',0.3)
% plot3(XX,YY,ZZ,'r')
scatter(rx,zeros(1,numel(rx)))
% scatter3(rx,0,ry)
% for jj=1:nfrag
%     plot([rx(jj) P(jj,1)],[0 P(jj,3)])
% end
grid on
%axis([0 250 0 500])
title ('Fragments projection')
xlabel('x (m)')
ylabel('z (m)')
xlim([-20 104])
ylim([0 90])
zlim([0 20])
daspect([1 1 1])

figure(5)
scatter(P(:,1),P(:,3))
set(gca, 'YDir', 'reverse' )
hold on
% scatter(rx,zeros(1,numel(rx)))
% for jj=1:nfrag
%     plot([rx(jj) P(jj,1)],[0 P(jj,3)])
% end
plot([0 W],[0 0],'r','LineWidth',2)
grid on
title ('Fragments projection')
xlabel('x (m)')
ylabel('z (m)')
legend('Fragment representation','Bench','Location','northeast')

[mu,sigma,muCI,sigmaCI] = normfit(P(:,3));

figure(6)
PD=fitdist(P(:,3),"Kernel")

```

```

xgrid = linspace(0,70,100)';
PDF_Est = pdf(PD,xgrid);
plot(xgrid,PDF_Est,'k-','LineWidth',2)
title ('Kernel Distribution model')
xlabel('Travelled distance in z-direction (m)')
ylabel('Probability')

% CF = cdf(PD,xgrid);

data = [P(:,1) P(:,3)];
[bandwidth,density,X,Y]=kde2d(data);

% plot the data and the density estimation
figure(7)
contour3(X,Y,density,50)
hold on
plot(data(:,1),data(:,2),'r.','MarkerSize',5)
plot([0 W],[0 0],'k','LineWidth',2)
title ('Bivariate Kernel density estimation')
set(gca, 'XDir', 'reverse' )
xlabel('Travelled distance (m)')
ylabel('Travelled distance (m)')

% Max distance at 100% - distance camera-bench
max_dist=prctile(range,100);

figure(8)
ecdf(range,'Bounds','on')
hold on
yline(1,'r','--')
xline(max_dist,'r')
plot(max_dist,0,'b*')
xlabel('Total travelled distance (m)')
ylabel('Cumulative probability')
title('Empirical Cumulative Distribution Function')
legend('CDF','Lower Confidence Bound','Upper Confidence Bound',"AutoUpdate","off",'Location','northwest')

DATA = [data f_radius];

%% Multi-objective optimization
ub = [pi/2, 2*pi, 2*pi, 250, 50];
lb = [0, 0, 0, 100, 1];

% Define the number of variables
nvars = 5;

% Define the options for the genetic algorithm
options = optimoptions('paretosearch','PlotFcn','psplotparetof','ParetoSetSize',200);

% Run the genetic algorithm
[XXX, fval] =
paretosearch(@(x)multiobjective2(x,DATA),nvars,[],[],[],[],lb,ub,[],options);

% Find the utopia point
utopia = [min(fval(:,1)) min(fval(:,2)) min(fval(:,3))];

% Calculate the distances between the utopia point and all points on the Pareto front

```

```

distances = pdist2(utopia,fval,'euclidean');

% Find the index of the point on the Pareto front with the minimum distance to the
utopia point
[min_distance, min_index] = min(distances);

% Get the decision variable values for the point on the Pareto front with the minimum
distance to the utopia point
best_p = fval(min_index,:);
dec_var = XXX(min_index,:);

figure(11)
plot3(fval(:,1),fval(:,2),fval(:,3),'o')
hold on
grid on
scatter3(utopia(1), utopia(2), utopia(3), 200, 'x', 'LineWidth', 2);
scatter3(best_p(1), best_p(2), best_p(3), 100, '*', 'LineWidth', 2);
xlabel('Objective 1 - Overlapping area');
ylabel('Objective 2 - Number of points inside the overlapping area');
zlabel('Objective 3 - Trapezoidal function y')
legend('Pareto Front','Utopia point','Optimal solution','Location','southeast')
title ('Pareto front')

%% Application of the multi-objective optimization
theta = dec_var(1);
w1 = dec_var(2);
w2 = dec_var(3);
radius = dec_var(4);
D = dec_var(5);

Cx = W/2;
Cz = 0;

R = 300;
beta = 2*asin(D/(2*radius));
delta = 72*pi/180;
arc_x = radius*cos(theta)+Cx;
arc_z = radius*sin(theta)+Cz;

data1 = [DATA(:,1) DATA(:,2)];
nfragg=size(data1,1);

% Coordinates of C1
C1_x = radius*cos(theta)+Cx;
C1_z = radius*sin(theta)+Cz;

% Coordinates of C2
C2_x = radius*cos(theta+beta)+Cx;
C2_z = radius*sin(theta+beta)+Cz;

% Triangle 1
X1_t1 = C1_x + R*cos(w1-(delta/2));
Z1_t1 = C1_z + R*sin(w1-(delta/2));

X2_t1 = C1_x + R*cos(w1+(delta/2));
Z2_t1 = C1_z + R*sin(w1+(delta/2));

% Triangle 2

```

```

X1_t2 = C2_x + R*cos(w2-(delta/2));
Z1_t2 = C2_z + R*sin(w2-(delta/2));

X2_t2 = C2_x + R*cos(w2+(delta/2));
Z2_t2 = C2_z + R*sin(w2+(delta/2));

V_t1 = [X2_t1 Z2_t1; C1_x C1_z; X1_t1 Z1_t1];
V_t2 = [X2_t2 Z2_t2; C2_x C2_z; X1_t2 Z1_t2];

mp_x = (C1_x+C2_x)/2;
mp_z = (C1_z+C2_z)/2;

pol1 = polyshape(V_t1(:,1), V_t1(:,2));
A1 = polyarea(V_t1(:,1), V_t1(:,2)); % m2
pol2 = polyshape(V_t2(:,1), V_t2(:,2));
A2 = polyarea(V_t2(:,1), V_t2(:,2)); % m2

% Define the vertices of the intersection polygon
int = intersect(pol1,pol2);
V_int = int.Vertices;

A_int = polyarea(V_int(:,1), V_int(:,2)); % m2
overlap = (A_int/A1); % percentage (%)
in = inhull(data1, V_int); % determine which points are inside the convex hull

% Extract the coordinates of the points that lie inside the intersection area
points_inside = DATA(in, :);

%Evaluation of dmax
mp_x = (C1_x+C2_x)/2;
mp_z = (C1_z+C2_z)/2;

%Evaluation of the distances between C1 and all the points inside the
%intersection area
dC1_f = sqrt((C1_x - points_inside(:,1)).^2 + (C1_z - points_inside(:,2)).^2);

%Evaluation of the distances between C2 and all the points inside the
%intersection area
dC2_f = sqrt((C2_x - points_inside(:,1)).^2 + (C2_z - points_inside(:,2)).^2);

dmax = max(sqrt((mp_x - points_inside(:,1)).^2 + (mp_z - points_inside(:,2)).^2));

a = dmax/D;

figure(12)
scatter(P(:,1),P(:,3), 'bo')
hold on
plot([0 W],[0 0], 'r', 'LineWidth', 2)
plot(arc_x, arc_z)
plot(C1_x,C1_z, 'r*')
plot(C2_x, C2_z, 'k*')
plot(V_t1(:,1),V_t1(:,2), '--r')
plot(V_t2(:,1),V_t2(:,2), '--k')
plot(pol1)
plot(pol2)
plot(int, 'EdgeColor','g','LineStyle','-')
grid on
title('Fragments projection')
xlabel('x (m)')

```

```

ylabel('z (m)')
legend('Fragment representation','Bench','','Camera 1','Camera 2','Angle of view
C1','Angle of view C2','Triangle 1','Triangle 2','Interception','Location','southwest')
axis equal

```

## % Kernel density function

```

function [bandwidth,density,X,Y]=kde2d(data,n,MIN_XY,MAX_XY)
global N A2 I
if nargin<2
    n=2^8;
end
n=2^ceil(log2(n)); % round up n to the next power of 2;
N=size(data,1);
if nargin<3
    MAX=max(data,[],1); MIN=min(data,[],1); Range=MAX-MIN;
    MAX_XY=MAX+Range/2; MIN_XY=MIN-Range/2;
end
scaling=MAX_XY-MIN_XY;
if N<=size(data,2)
    error('data has to be an N by 2 array where each row represents a two dimensional
observation')
end
transformed_data=(data-repmat(MIN_XY,N,1))./repmat(scaling,N,1);
%bin the data uniformly using regular grid;
initial_data=ndhist(transformed_data,n);
% discrete cosine transform of initial data
a= dct2d(initial_data);
% now compute the optimal bandwidth^2
I=(0:n-1).^2; A2=a.^2;
t_star=root(@(t)(t-evolve(t)),N);
p_02=func([0,2],t_star);p_20=func([2,0],t_star); p_11=func([1,1],t_star);
t_y=(p_02^(3/4)/(4*pi*N*p_20^(3/4)*(p_11+sqrt(p_20*p_02))))^(1/3);
t_x=(p_20^(3/4)/(4*pi*N*p_02^(3/4)*(p_11+sqrt(p_20*p_02))))^(1/3);
% smooth the discrete cosine transform of initial data using t_star
a_t=exp(-(0:n-1).^2*pi^2*t_x/2)*exp(-(0:n-1).^2*pi^2*t_y/2).*a;
% now apply the inverse discrete cosine transform
if nargin>1
    density=idct2d(a_t)*(numel(a_t)/prod(scaling));
    density(density<0)=eps; % remove any negative density values
    [X,Y]=meshgrid(MIN_XY(1):scaling(1)/(n-1):MAX_XY(1),MIN_XY(2):scaling(2)/(n-
1):MAX_XY(2));
end
bandwidth=sqrt([t_x,t_y]).*scaling;
end
#####
function [out,time]=evolve(t)
global N
Sum_func = func([0,2],t) + func([2,0],t) + 2*func([1,1],t);
time=(2*pi*N*Sum_func)^(-1/3);
out=(t-time)/time;
end
#####
function out=func(s,t)
global N
if sum(s)<=4
    Sum_func=func([s(1)+1,s(2)],t)+func([s(1),s(2)+1],t); const=(1+1/2^(sum(s)+1))/3;
    time=(-2*const*K(s(1))*K(s(2))/N/Sum_func)^(1/(2+sum(s)));
    out=psi(s,time);

```

```

else
    out=psi(s,t);
end

end

#####
function out=psi(s,Time)
global I A2
% s is a vector
w=exp(-I*pi^2*Time).*[1,.5*ones(1,length(I)-1)];
wx=w.*(I.^s(1));
wy=w.*(I.^s(2));
out=(-1)^sum(s)*(wy*A2*wx')*pi^(2*sum(s));
end
#####
function out=K(s)
out=(-1)^s*prod((1:2:2*s-1))/sqrt(2*pi);
end
#####
function data=dct2d(data)
% computes the 2 dimensional discrete cosine transform of data
% data is an nd cube
[nrows,ncols]= size(data);
if nrows~=ncols
    error('data is not a square array!')
end
% Compute weights to multiply DFT coefficients
w = [1;2*(exp(-i*(1:nrows-1)*pi/(2*nrows))).'];
weight=w(:,ones(1,ncols));
data=dct1d(dct1d(data)')';
    function transform1d=dct1d(x)

        % Re-order the elements of the columns of x
        x = [ x(1:2:end,:); x(end:-2:2,:) ];

        % Multiply FFT by weights:
        transform1d = real(weight.* fft(x));
    end
end
#####
function data = idct2d(data)
% computes the 2 dimensional inverse discrete cosine transform
[nrows,ncols]=size(data);
% Compute wieghts
w = exp(i*(0:nrows-1)*pi/(2*nrows)).';
weights=w(:,ones(1,ncols));
data=idct1d(idct1d(data)');
    function out=idct1d(x)
        y = real(ifft(weights.*x));
        out = zeros(nrows,ncols);
        out(1:2:nrows,:) = y(1:nrows/2,:);
        out(2:2:nrows,:) = y(nrows:-1:nrows/2+1,:);
    end
end
#####
function binned_data=ndhist(data,M)
% this function computes the histogram
% of an n-dimensional data set;
% 'data' is nrows by n columns
% M is the number of bins used in each dimension

```

```

% so that 'binned_data' is a hypercube with
% size length equal to M;
[nrows,ncols]=size(data);
bins=zeros(nrows,ncols);
for i=1:ncols
    [dum,bins(:,i)] = histc(data(:,i),[0:1/M:1],1);
    bins(:,i) = min(bins(:,i),M);
end
% Combine the vectors of 1D bin counts into a grid of nD bin
% counts.
binned_data = accumarray(bins(all(bins>0,2),:),1/nrows,M(ones(1,ncols)));
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function t=root(f,N)
% try to find smallest root whenever there is more than one
N=50*(N<=50)+1050*(N>=1050)+N*((N<1050)&(N>50));
tol=10^-12+0.01*(N-50)/1000;
flag=0;
while flag==0
    try
        t=fzero(f,[0,tol]);
        flag=1;
    catch
        tol=min(tol*2,.1); % double search interval
    end
    if tol==.1 % if all else fails
        t=fminbnd(@(x)abs(f(x)),0,.1); flag=1;
    end
end
end
end

```

## % Multi-objective optimization function

```
function F = multiobjective2(x,DATA)
```

```

theta = x(1);
w1 = x(2);
w2 = x(3);
radius = x(4);
D = x(5);

```

```

Cx = 84/2;      % W
Cz = 0;
R = 500;
beta = 2*asin(D./(2*radius));
delta = 72*pi/180;
arc_x = radius.*cos(theta)+Cx;
arc_z = radius.*sin(theta)+Cz;

```

```

data1 = [DATA(:,1) DATA(:,2)];
nfragg=size(data1,1);

```

### % Coordinates of C1

```

C1_x = radius.*cos(theta)+Cx;
C1_z = radius.*sin(theta)+Cz;

```

### % Coordinates of C2

```
C2_x = radius.*cos(theta+beta)+Cx;
```

```

C2_z = radius.*sin(theta+beta)+Cz;

% Triangle 1
X1_t1 = C1_x + R*cos(w1-(delta/2));
Z1_t1 = C1_z + R*sin(w1-(delta/2));

X2_t1 = C1_x + R*cos(w1+(delta/2));
Z2_t1 = C1_z + R*sin(w1+(delta/2));

% Triangle 2
X1_t2 = C2_x + R*cos(w2-(delta/2));
Z1_t2 = C2_z + R*sin(w2-(delta/2));

X2_t2 = C2_x + R*cos(w2+(delta/2));
Z2_t2 = C2_z + R*sin(w2+(delta/2));

V_t1 = [X2_t1 Z2_t1; C1_x C1_z; X1_t1 Z1_t1];
V_t2 = [X2_t2 Z2_t2; C2_x C2_z; X1_t2 Z1_t2];
mp_x = (C1_x+C2_x)/2;
mp_z = (C1_z+C2_z)/2;
pol1 = polyshape(V_t1(:,1), V_t1(:,2));
A1 = polyarea(V_t1(:,1), V_t1(:,2)); % m2
pol2 = polyshape(V_t2(:,1), V_t2(:,2));
A2 = polyarea(V_t2(:,1), V_t2(:,2)); % m2
% Define the vertices of the intersection polygon
int = intersect(pol1,pol2);
V_int = int.Vertices;

if isempty(int.Vertices)==0
    A_int = polyarea(V_int(:,1), V_int(:,2)); % m2
    overlap = (A_int/A1); % percentage (%)
    in = inhull(data1, V_int); % determine which points are inside the convex
hull

    F1 = -overlap;
    F2 = -sum(in)/nfragg;

    %Extract the coordinates of the points that lie inside the
    %intersection area
    points_inside = DATA(in, :);

    if isempty(points_inside)==0

        mp_x = (C1_x+C2_x)/2;
        mp_z = (C1_z+C2_z)/2;

        %Evaluation of the distances between C1 and all the points inside the
        %intersection area
        dC1_f = sqrt((C1_x - points_inside(:,1)).^2 + (C1_z -
points_inside(:,2)).^2);

        %Evaluation of the distances between C2 and all the points inside the
        %intersection area
        dC2_f = sqrt((C2_x - points_inside(:,1)).^2 + (C2_z -
points_inside(:,2)).^2);

        dmax = max(sqrt((mp_x - points_inside(:,1)).^2 + (mp_z -
points_inside(:,2)).^2));

        a = dmax/D;

```

```

        F3 = (a<2).*0 + ((a>=2)&&(a<4)).*(-a+2) + ((a>=4)&&(a<5)).*-2 +
((a>=5)&&(a<7)).*(a-7) + (a>=7).*0;
        else
        F3 = 0;
        end

    else
        F1=0;
        F2=0;
        F3=0;
    end

F=[F1; F2; F3];
end

%-----

clc
clear all
close all

%% FRAMES EXTRACTION - CAMERA LEFT
% Frames extraction from the calibration video
% import the video file
cd 'C:\Users\pagon\OneDrive\Desktop\MUMS\THESIS\Video_Processing\Left_1206'

% import the video file
obj1 = VideoReader('20230612115401.MTS');
vidHeight1 = obj1.Height;
vidWidth1 = obj1.Width;
%

% set the time interval to extract frames from (in seconds)
startTime1 = 54; % start at 30 seconds
endTime1 = 221; % end at 54 seconds

% convert the time interval to frame indices
startFrame1 = floor(startTime1 * obj1.FrameRate) + 1;
endFrame1 = floor(endTime1 * obj1.FrameRate);

ST1 = '.jpg';
% reading and writing the frames
for i1 = startFrame1 : endFrame1
    % read the current frame
    frame_cal = read(obj1, i1);

    % convert integer to string
    Sx1 = num2str(i1);

    % concatenate the filename with the file format
    Strc1 = strcat(Sx1, ST1);

    % set the directory to save the frames in
    cd frames_calib_2

    % export the frame
    imwrite(frame_cal, Strc1);

```

```

    cd ..
end

%%-----

% Frames extraction from the falling object video
% import the video file
cd 'C:\Users\pagon\OneDrive\Desktop\MUMS\THESIS\Video_Processing\Left_1206'

% import the video file
obj = VideoReader('20230612140854.MTS');
vidHeight = obj.Height;
vidWidth = obj.Width;

% set the time interval to extract frames from (in seconds)
startTime = 552; % start at 30 seconds
endTime = 669; % end at 54 seconds

% convert the time interval to frame indices
startFrame = floor(startTime * obj.FrameRate) + 1;
endFrame = floor(endTime * obj.FrameRate);

ST='.jpg';
% reading and writing the frames
for i = startFrame : endFrame
    % read the current frame
    frame_obj = read(obj, i);

    % convert integer to string
    Sx = num2str(i);

    % concatenate the filename with the file format
    Strc = strcat(Sx, ST);

    % set the directory to save the frames in
    cd frames_object

    % export the frame
    imwrite(frame_obj, Strc);

    cd ..
end

%%-----

clc
clear all
close all

%% FRAMES EXTRACTION - CAMERA RIGHT
% Frames extraction from the calibration video
% import the video file
cd 'C:\Users\pagon\OneDrive\Desktop\MUMS\THESIS\Video_Processing\Right_1206'

% import the video file
obj1 = VideoReader('20230612115242.MTS');
vidHeight1 = obj1.Height;
vidWidth1 = obj1.Width;

```

```

% set the time interval to extract frames from (in seconds)
startTime1 = 115; % start at 30 seconds
endTime1 = 305; % end at 54 seconds

% convert the time interval to frame indices
startFrame1 = floor(startTime1 * obj1.FrameRate) + 1;
endFrame1 = floor(endTime1 * obj1.FrameRate);

ST1 = '.jpg';
% reading and writing the frames
for i1 = startFrame1 : endFrame1
    % read the current frame
    frame_cal = read(obj1, i1);

    % convert integer to string
    Sx1 = num2str(i1);

    % concatenate the filename with the file format
    Strc1 = strcat(Sx1, ST1);

    % set the directory to save the frames in
    cd frames_calib2

    % export the frame
    imwrite(frame_cal, Strc1);

    cd ..
end

%%-----

% Frames extraction from the falling object video
% import the video file
cd 'C:\Users\pagon\OneDrive\Desktop\MUMS\THESIS\Video_Processing\Right_1206'

% import the video file
obj = VideoReader('20230612140805.MTS');
vidHeight = obj.Height;
vidWidth = obj.Width;

% set the time interval to extract frames from (in seconds)
startTime = 593.4; % start at 30 seconds
endTime = 710.4; % end at 54 seconds

% convert the time interval to frame indices
startFrame = floor(startTime * obj.FrameRate) + 1;
endFrame = floor(endTime * obj.FrameRate);

ST='.jpg';
% reading and writing the frames
for i = startFrame : endFrame
    % read the current frame
    frame_obj = read(obj, i);

    % convert integer to string
    Sx = num2str(i);

    % concatenate the filename with the file format
    Strc = strcat(Sx, ST);

```

```

% set the directory to save the frames in
cd frames_object

% export the frame
imwrite(frame_obj, Strc);

cd ..
end

```

## % Video processing

```

%% CAMERA CALIBRATION
% Define images to process
imageFileNames1 =
{'C:\Users\pagon\OneDrive\Desktop\MUMS\THESIS\Video_Processing\Left_2305\frames_calibra
tion\frame_200.jpg',
'C:\Users\pagon\OneDrive\Desktop\MUMS\THESIS\Video_Processing\Left_2305\frames_calibra
tion\frame_600.jpg',
'C:\Users\pagon\OneDrive\Desktop\MUMS\THESIS\Video_Processing\Left_2305\frames_calibra
tion\frame_1000.jpg',
'C:\Users\pagon\OneDrive\Desktop\MUMS\THESIS\Video_Processing\Left_2305\frames_calibra
tion\frame_1400.jpg',
'C:\Users\pagon\OneDrive\Desktop\MUMS\THESIS\Video_Processing\Left_2305\frames_calibra
tion\frame_2650.jpg',
'C:\Users\pagon\OneDrive\Desktop\MUMS\THESIS\Video_Processing\Left_2305\frames_calibra
tion\frame_4300.jpg'}; % Left

imageFileNames2 =
{'C:\Users\pagon\OneDrive\Desktop\MUMS\THESIS\Video_Processing\Right_2305\frames_calibra
tion\frame_100.jpg',
'C:\Users\pagon\OneDrive\Desktop\MUMS\THESIS\Video_Processing\Right_2305\frames_calibra
tion\frame_600.jpg',
'C:\Users\pagon\OneDrive\Desktop\MUMS\THESIS\Video_Processing\Right_2305\frames_calibra
tion\frame_1050.jpg',
'C:\Users\pagon\OneDrive\Desktop\MUMS\THESIS\Video_Processing\Right_2305\frames_calibra
tion\frame_1350.jpg',
'C:\Users\pagon\OneDrive\Desktop\MUMS\THESIS\Video_Processing\Right_2305\frames_calibra
tion\frame_2450.jpg',
'C:\Users\pagon\OneDrive\Desktop\MUMS\THESIS\Video_Processing\Right_2305\frames_calibra
tion\frame_4300.jpg'}; % Right

% Detect checkboards in images
[imagePoints,boardSize,pairsUsed] = detectCheckerboardPoints(imageFileNames1,
imageFileNames2);
[imagePoints, boardSize, imagesUsed] = detectCheckerboardPoints(imageFileNames1,
imageFileNames2);

% Generate world coordinates of the checkboard keypoints
squareSize = 28; % in mm
worldPoints = generateCheckerboardPoints(boardSize, squareSize);

% Read one of the images from the first stereo pair
I1 = imread(imageFileNames1{1});
[mrows, ncols, ~] = size(I1);

% Calibrate the camera
[stereoParams,pairsUsed,estimationErrors] =
estimateCameraParameters(imagePoints,worldPoints, ...

```

```

    'EstimateSkew',false,'EstimateTangentialDistortion',false, ...
    'NumRadialDistortionCoefficients', 2,'WorldUnits','millimeters', ...
    'InitialIntrinsicMatrix', [], 'InitialRadialDistortion', [], ...
    'ImageSize', [mrows, ncols]);

% View reprojection errors
h1 = figure;
showReprojectionErrors(stereoParams);

% Visualize pattern locations
h2 = figure;
showExtrinsics(stereoParams, 'CameraCentric');

% Display parameter estimation errors
displayErrors(estimationErrors, stereoParams);

% Camera 1 - Left
r1 = eye(3); % Rotation matrix
w1 = (pi/180)*(0);
p1 = (pi/180)*(0);
k1 = (pi/180)*(0);
Tx1 = 0; Ty1 = 0; Tz1 = 0; % Translation matrix
T1 = [Tx1 Ty1 Tz1];
angles1 = rad2deg([w1 p1 k1]);
mw1 = [1 0 0; 0 cos(w1) sin(w1); 0 -sin(w1) cos(w1)];
mp1 = [cos(p1) 0 -sin(p1); 0 1 0; sin(p1) 0 cos(p1)];
mk1 = [cos(k1) sin(k1) 0; -sin(k1) cos(k1) 0; 0 0 1];
m1 = mw1*mp1*mk1;

% Camera 2 - Right
r2 = stereoParams.PoseCamera2.R; %stereoParams.RotationOfCamera2;
eulXYZ = rotm2eul(r2, 'XYZ');
w2 = -eulXYZ(1);
p2 = -eulXYZ(2);
k2 = -eulXYZ(3);
Tx2 = -stereoParams.TranslationOfCamera2(1);
Ty2 = -stereoParams.TranslationOfCamera2(2);
Tz2 = -stereoParams.TranslationOfCamera2(3);
T2 = [Tx2 Ty2 Tz2];
angles2 = rad2deg([w2 p2 k2]);
mw2 = [1 0 0; 0 cos(w2) sin(w2); 0 -sin(w2) cos(w2)];
mp2 = [cos(p2) 0 -sin(p2); 0 1 0; sin(p2) 0 cos(p2)];
mk2 = [cos(k2) sin(k2) 0; -sin(k2) cos(k2) 0; 0 0 1];
m2 = mw2*mp2*mk2;

%% -----

f = 8.8; % Focal length 8.8-25.7
% Camera 1 - Left camera focal length (pixels)
f1 = norm(stereoParams.CameraParameters1.FocalLength);
% Camera 2 - Right camera focal length (pixels)
f2 = norm(stereoParams.CameraParameters2.FocalLength);
psize = [f/f1, f/f2];

%% -----

% LEFT CAMERA - Correction from lens distortion
pathname_left = 'C:\Users\pagon\OneDrive\Desktop\MUMS\THESIS\Video_Processing\camera
left\frames_object';
file_list_left = dir(fullfile(pathname_left, '*.jpg'));

```

```

image_left = cell(length(file_list_left), 1);
und_image_left = cell(length(file_list_left), 1);

for i = 1:length(file_list_left)
    current_file_name_left = file_list_left(i).name;
    current_file_path_left = fullfile(pathname_left, current_file_name_left);
    current_image_left = imread(current_file_path_left);

    image_left{i} = current_image_left;

    cameraParams_left = {stereoParams.CameraParameters1
stereoParams.CameraParameters2};
    und_image_left{i} = undistortImage(image_left{i}, cameraParams_left{1});
end

for i = 1:length(und_image_left)
    output_file_name_left = sprintf('image_%04d.jpg', i);
    output_file_path_left = fullfile('corrected_images', output_file_name_left);

    imwrite(und_image_left{i}, output_file_path_left);
end

% CREATION OF THE VIDEO WITH THE CORRECTED IMAGES - LEFT CAMERA
% Setup
workingDir = 'C:\Users\pagon\OneDrive\Desktop\MUMS\THESIS\Video_Processing\camera
left\corrected_images';
imageNames = dir(fullfile(workingDir, '*.jpg')); %'image_%04d',
imageNames = {imageNames.name}';

% Creation of a new video with the image sequence
outputVideo = VideoWriter(fullfile(workingDir, 'moving_obj_left1.avi'));
% outputVideo.FrameRate = outputVideo.FrameRate;
open(outputVideo)
for ii = 1:length(imageNames)
    img = imread(fullfile(workingDir, imageNames{ii}));
    writeVideo(outputVideo, img)
end
close(outputVideo)

% View the new video
corr_left_Video_Avi = VideoReader(fullfile(workingDir, 'moving_obj_left1.avi'));
ii = 1;
while hasFrame(corr_left_Video_Avi)
    mov(ii) = im2frame(readFrame(corr_left_Video_Avi));
    ii = ii+1;
end

figure
imshow(mov(1).cdata, 'Border', 'tight')

movie(mov,1,corr_left_Video_Avi.FrameRate)

%%-----
%% Camera left - Tracking object

cd 'C:\Users\pagon\OneDrive\Desktop\MUMS\THESIS\Video_Processing\Left_2305'
obj_left=FUNC_MotionBasedMultiObjectTrackingExample('moving_obj_left.avi');

```

```

v_left = VideoReader('moving_obj_left.avi');
areas_left = {};
centroids_left = {};
bbox_left = {};
areas_all_frames_left = {};
bbox_all_frames_left = {};
centroids_all_frames_left = {}; % initialize empty cell array
tracks_left = []; % initialize empty array for tracks

while hasFrame(v_left)
    frame_left = readFrame(v_left);
    % Detect foreground.
    mask_left = obj_left.detector.step(frame_left);

    % Apply morphological operations to remove noise and fill in holes.
    mask_left = imopen(mask_left, strel('rectangle', [3,3]));
    mask_left = imclose(mask_left, strel('rectangle', [15, 15]));
    mask_left = imfill(mask_left, 'holes');

    detector_left = vision.ForegroundDetector('NumGaussians', 5, ...
        'NumTrainingFrames', 40, 'MinimumBackgroundRatio', 0.7);
    blobAnalysis_left = vision.BlobAnalysis('BoundingBoxOutputPort', true, ...
        'AreaOutputPort', true, 'CentroidOutputPort', true, ...
        'MinimumBlobArea', 200 );
    % Moving object detection - Perform blob analysis to find connected components.
    [areas_left, centroids_left, bbox_left] =
obj_left.blobAnalyser.step(mask_left);
    % Store centroids in cell array
    centroids_all_frames_left{end+1} = centroids_left;
    areas_all_frames_left{end+1} = areas_left;
    bbox_all_frames_left{end+1} = bbox_left;

end

centroidX_l = {};
centroidY_l = {};
for i = 1:length(centroids_all_frames_left)
    centroidX_l{i} = centroids_all_frames_left{i}(:, 1);
    centroidY_l{i} = centroids_all_frames_left{i}(:, 2);
end

figure(10)
hold on % Add subsequent scatter plots to the same figure
for i = 1:length(centroidX_l) %400:length(centroidX_l)
    scatter(centroidX_l{i}, centroidY_l{i}); % centroidX_l{i}, centroidY_l{i}
    % text(centroidX_l{i}, centroidY_l{i}, num2str(i), 'Color', 'red', 'FontSize',5);
end
title('Centroid Coordinates - Left camera');
xlabel('X (px)');
ylabel('Y (px)');
grid on
set(gca, 'Ydir', 'reverse')

aux_l1=centroids_all_frames_left(102:200);
distances_l1 = cell(numel(aux_l1), 1);
stp_l1=[];
idx_l1=1;
points_l1=cell(numel(aux_l1),1);

```

```

centroids_l1=cell(numel(aux_l1), 1);
deriv_l1=cell(numel(aux_l1), 1);
thresh_l1=100;
threshd_l1= 0.1; %0.1;

for ii=1:numel(aux_l1)
    if ii==1
        stp_l1=aux_l1{ii}(idx_l1,:);
        points_l1{ii}=stp_l1;
        centroids_l1{ii} = aux_l1{ii};
    else
        centroids_l1{ii}=aux_l1{ii};
        distances_l1{ii} = pdist2(stp_l1,centroids_l1{ii});
        deriv_l1{ii} = atan2(centroids_l1{ii}(:,2)-stp_l1(2),centroids_l1{ii}(:,1)-
stp_l1(1));
        if ii==2
            deriv_l1{ii-1}=deriv_l1{ii};
        end
        if all(distances_l1{ii} > thresh_l1) & all(abs(deriv_l1{ii} - deriv_l1{ii -
1}(idx_l1,:))' > threshd_l1) % && (idx_l1 <= numel(deriv_l1{ii - 1}))
            points_l1{ii}=[];
        else
            %[~,idx_l1] = min(distances_l1{ii});
            [~,idx_l1]=min(distances_l1{ii}.*abs(deriv_l1{ii}-deriv_l1{ii-
1}(idx_l1,:))');
            points_l1{ii}=centroids_l1{ii}(idx_l1,:);
            stp_l1=centroids_l1{ii}(idx_l1,:);
        end
    end
end
end

figure(11)
hold on
for ii=1:numel(points_l1)
    if isempty(points_l1{ii})
        continue; % Skip empty cells
    end
    plot(points_l1{ii}(1),points_l1{ii}(2),'ob')
end
title('Object trajectory 2D - Left camera');
xlabel('X (px)');
ylabel('Y (px)');
grid on
set(gca, 'Ydir', 'reverse')

%%-----

% RIGHT CAMERA - Correction from lens distortion
pathname_right = 'C:\Users\pagon\OneDrive\Desktop\MUMS\THESIS\Video_Processing\camera
right\frames_object';
file_list_right = dir(fullfile(pathname_right, '*.jpg'));
image_right = cell(length(file_list_right), 1);
und_image_right = cell(length(file_list_right), 1);

for i = 1:length(file_list_right)

    current_file_name_right = file_list_right(i).name;

```

```

current_file_path_right = fullfile(pathname_right, current_file_name_right);
current_image_right = imread(current_file_path_right);

image_right{i} = current_image_right;

cameraParams_right = {stereoParams.CameraParameters1
stereoParams.CameraParameters2};
und_image_right{i} = undistortImage(image_right{i}, cameraParams_right{2});
end

for i = 1:length(und_image_right)
    output_file_name_right = sprintf('image_%04d.jpg', i);
    output_file_path_right = fullfile('corrected_images', output_file_name_right);

    imwrite(und_image_right{i}, output_file_path_right);
end

% CREATION OF THE VIDEO WITH THE CORRECTED IMAGES - RIGHT CAMERA
% Setup
workingDir_r = 'C:\Users\pagon\OneDrive\Desktop\MUMS\THESIS\Video_Processing\camera
right\corrected_images';
imageNames_r = dir(fullfile(workingDir_r, '*.jpg')); %'image_%04d',
imageNames_r = {imageNames_r.name}';

% Creation of a new video with the image sequence
outputVideo_r = VideoWriter(fullfile(workingDir_r, 'moving_obj_right1.avi'));
% outputVideo.FrameRate = outputVideo.FrameRate;
open(outputVideo_r)
for ii = 1:length(imageNames_r)
    img_r = imread(fullfile(workingDir_r, imageNames_r{ii}));
    writeVideo(outputVideo_r, img_r)
end
close(outputVideo_r)

% View the new video
corr_right_Video_Avi = VideoReader(fullfile(workingDir_r, 'moving_obj_right1.avi'));
ii = 1;
while hasFrame(corr_right_Video_Avi)
    mov_r(ii) = im2frame(readFrame(corr_right_Video_Avi));
    ii = ii+1;
end

figure
imshow(mov_r(1).cdata, 'Border', 'tight')

movie(mov_r, 1, corr_right_Video_Avi.FrameRate)

cd 'C:\Users\pagon\OneDrive\Desktop\MUMS\THESIS\Video_Processing\Right_2305'
obj_right = FUNC_MotionBasedMultiObjectTrackingExample('moving_obj_right.avi');

v_right = VideoReader('moving_obj_right.avi');
areas_right = {};
centroids_right = {};
bbox_right = {};
areas_all_frames_right = {};
bbox_all_frames_right = {};
centroids_all_frames_right = {}; % initialize empty cell array
while hasFrame(v_right)

```

```

frame_right = readFrame(v_right);
% Detect foreground.
mask_right = obj_right.detector.step(frame_right);

% Apply morphological operations to remove noise and fill in holes.
mask_right = imopen(mask_right, strel('rectangle', [3,3]));
mask_right = imclose(mask_right, strel('rectangle', [15, 15]));
mask_right = imfill(mask_right, 'holes');

detector_right = vision.ForegroundDetector('NumGaussians', 5, ...
    'NumTrainingFrames', 40, 'MinimumBackgroundRatio', 0.7);
blobAnalysis_left = vision.BlobAnalysis('BoundingBoxOutputPort', true, ...
    'AreaOutputPort', true, 'CentroidOutputPort', true, ...
    'MinimumBlobArea', 200 );
% Moving object detection - Perform blob analysis to find connected components.
[areas_right, centroids_right, bbox_right] =
obj_right.blobAnalyser.step(mask_right);
% Store centroids in cell array
centroids_all_frames_right{end+1} = centroids_right;
areas_all_frames_right{end+1} = areas_right;
bbox_all_frames_right{end+1} = bbox_right;
end

centroidX_r = {};
centroidY_r = {};
for i = 1:length(centroids_all_frames_right)
    centroidX_r{i} = centroids_all_frames_right{i}(:, 1);
    centroidY_r{i} = centroids_all_frames_right{i}(:, 2);
end

figure(12)
hold on % Add subsequent scatter plots to the same figure
for i = 1:length(centroidX_r) % 389:length(centroidX_r)
    scatter(centroidX_r{i}, centroidY_r{i}); %centroidX_r{i}, centroidY_r{i}
    % text(centroidX_r{i}, centroidY_r{i}, num2str(i), 'Color', 'red', 'FontSize', 5);
end
title('Centroid Coordinates - Right camera');
xlabel('X (px)');
ylabel('Y (px)');
grid on
set(gca, 'Ydir', 'reverse')

aux_r1=centroids_all_frames_right(102:200);
distances_r1 = cell(numel(aux_r1),1);
stp_r1=[];
idx_r1=2;
points_r1=cell(numel(aux_r1),1);
centroids_r1 = cell(numel(aux_r1),1);
deriv_r1=cell(numel(aux_r1), 1);
thresh_r1=1000;
threshd_r1=0.05; %0.1

for ii=1:numel(aux_r1)
    if ii==1
        stp_r1=aux_r1{ii}(idx_r1,:);
        points_r1{ii}=stp_r1;
        centroids_r1{ii} = aux_r1{ii};
    else
        centroids_r1{ii}=aux_r1{ii};
    end
end

```

```

        distances_r1{ii} = pdist2(stp_r1,centroids_r1{ii});
        deriv_r1{ii} = atan2(centroids_r1{ii}(:,2)-stp_r1(2),centroids_r1{ii}(:,1)-
stp_r1(1));
        if ii==2
            deriv_r1{ii-1}=deriv_r1{ii};
        end
        if all(distances_r1{ii} > thresh_r1) & all(abs(deriv_r1{ii}-deriv_r1{ii-
1}(idx_r1)) > threshd_r1)
            points_r1{ii}=[];
        else
%           [~,idx_r1] = min(distances_r1{ii});
            [~,idx_r1]=min(distances_r1{ii}.*abs(deriv_r1{ii}-deriv_r1{ii-1}(idx_r1)));
            points_r1{ii}=centroids_r1{ii}(idx_r1,:);
            stp_r1=centroids_r1{ii}(idx_r1,:);
        end
    end
end

figure(13)
hold on
for ii=1:numel(points_r1)
    if isempty(points_r1{ii})
        continue; % Skip empty cells
    end
    plot(points_r1{ii}(1),points_r1{ii}(2),'ob')
end
title('Object trajectory 2D - Right camera');
xlabel('X (px)');
ylabel('Y (px)');
grid on
set(gca, 'Ydir', 'reverse')

%%-----

% Triangulation
points_l1_c = points_l1(1:84);
points_r1_c = points_r1(1:84);
p_l1 = zeros(length(points_l1_c),2);
p_r1 = zeros(length(points_r1_c),2);
for i=1:84
    p_l1(i,:) = points_l1_c{i};
    p_r1(i,:) = points_r1_c{i};
end

% Compute the distance from camera 1 to the face.
point3d = zeros(length(p_l1),3);
distanceInMeters = [];
for i = 1:length(p_l1)
    point3d(i,:) = triangulate(p_l1(i,:), p_r1(i,:), stereoParams);
    distanceInMeters(i)= norm(point3d(i,:))/1000;
end

h2 = figure;
showExtrinsics(stereoParams, 'CameraCentric');
hold on
for i=1:length(point3d)
    scatter3(point3d(i,1), point3d(i,3), point3d(i,2), 'ob')
end

```

```

%% Result presentation
T1 = [Tx1; Ty1; Tz1]; % r1
T2 = [Tx2; Ty2; Tz2]; % r2
C1 = -r1' * T1;
C2 = r2' * T2;

figure (14)
hold on;
% set(gca, 'Xdir', 'reverse')
set(gca, 'Zdir', 'reverse')
cam1 = plot3(C1(1), C1(3), C1(2), 'bo', 'MarkerSize', 10); % Camera 1 position
cam2 = plot3(C2(1), C2(3), C2(2), 'ro', 'MarkerSize', 10); % Camera 2 position
grid on;

for i=1:length(point3d)
    frag = scatter3(point3d(i,1), point3d(i,3), point3d(i,2), 'ob');
    hold on
    grid on
end

title('Object trajectory 3D');
xlabel('X (mm)')
ylabel('Z (mm)')
zlabel('Y (mm)')
lgd = legend([cam1, cam2, frag], 'Left camera', 'Right camera', 'Fragments');
daspect([1 1 1])

```