

Development of a Residual Stress Measuring System Based on the Contour Method

Carlos Alexandre Dias Contente

Master in Mechanical Engineering Dissertation

Supervisors: Dr. Pedro Moreira, Dr. Paulo J. S. Tavares,
Dr. Daniel Braga



Master in Mechanical Engineering
September 2020

Abstract

This work describes the development of a contour acquisition system designed to determine residual stress fields through the Contour Method. The system components, its possible configurations and precautions to be taken during its operation are explained. Moreover, the precautions to be taken in the image acquisition concerning the camera calibration, the speed calibration and, finally, the formation of the point clouds are also described.

Throughout this dissertation, the transformations to be applied to the acquired point clouds are listed and justified, so that they can be assigned as valid displacement boundary conditions, when applying the Contour Method.

Two different specimens were tested and the results agreed with expectations. In both cases it is found that most of the surface points presented valid residual stress values, however, a considerable part of them invalidated the results by surpassing the maximum error allowed.

Some sources of error, which are detrimental to the point clouds shapes, were eliminated and a method to correct the rigid rotations of the acquired clouds was applied. However, there were identified limitations that remained to be overcome, namely the accuracy of the scanning speed value and mainly, the system's resolution, which did not reach the minimum required. Finally, as future work, suggestions are made to overcome each one of the mentioned limitations.

Resumo

Este trabalho visa o desenvolvimento de um sistema de aquisição de contorno, com vista a determinação de tensões residuais através do Método Contour. São descritas as partes constituintes do sistema, bem como as possíveis configurações deste, cuidados a ter na sua operação e processo de calibração. São também descritos os cuidados a ter na aquisição de imagens respeitantes à calibração da câmara, à calibração da velocidade e, por fim, respeitantes à formação das nuvens de pontos.

Ao longo desta dissertação são também enumeradas e justificadas as transformações a aplicar às nuvens de pontos adquiridas, para que estas possam constituir condições de fronteira de deslocamentos válidas, na aplicação do Método Contour.

Foram testados dois provetes diferentes e os resultados mostraram-se coerentes com o expectável. Em ambos os casos se verifica que a maior parte dos pontos estudados apresentam resultados válidos de tensão residual, contudo, uma ainda considerável parte deles invalida os resultados, pelo facto de apresentarem valores de erro superiores ao máximo aceitável.

Foram eliminadas algumas fontes de erro prejudiciais à forma das nuvens adquiridas, e foi aplicado um método que corrige as rotações rígidas destas, contudo, houve limitações identificadas que ficaram por ultrapassar, nomeadamente a precisão do valor determinado para a velocidade de varrimento e, principalmente, a resolução do sistema de medição, que não atingiu o valor mínimo requerido. Finalmente, como trabalhos futuros, são apresentadas sugestões para colmatar cada uma das limitações mencionadas.

Acknowledgments

First, I would like to thank Dr. Pedro Miguel Guimarães Pires Moreira, Dr. Paulo José da Silva Tavares and Dr. Daniel Filipe Oliveira Braga, supervisors of this project, for all the guidance provided during all stages of this work.

I would also like to thank Dr. Behzad Vasheghani Farahani, Eng. Pedro José Silva Carvalho Pereira Sousa and Eng. Francisco Barros for all the help they gave me in optics and in understanding the MATLAB algorithms involved, emphasizing their patience and disponibility during all the project.

Moreover, I must extend my acknowledgments to Eng. Nuno Viriato and Eng. Tiago Domingues, who provided me all the necessary help from a logistic point of view, suggesting me several constructive solutions for the system, for both fixing and alignment purposes.

Last but not least, I would like to thank Eng. Lucas Dourado Azevedo for his cooperation during the final stage of this project, giving me access to Quantal S.A.'s specimen and the respective CMM measurements.

*"Ya é trippy believe me,
Por isso fico a esculpir aqui
A minha vida, a minha pedra, vou sair dela David."*

Mário Cotrim

Contents

Abbreviations	xi
1 Introduction	1
1.1 Motivation, Problem Statement and Goals	1
1.2 Dissertation Synopsis	2
2 Introduction to Residual Stress	3
2.1 Origin and Types of Residual Stresses	3
2.2 Measurement Methods of Residual Stresses	5
2.2.1 Curvature	7
2.2.2 Sectioning method	7
2.2.3 Hole-drilling method	9
2.2.4 Deep-hole method	11
2.2.5 X-ray and neutron diffraction	12
2.2.6 Contour method	14
3 Introduction to Camera Calibration	19
3.1 Pinhole Camera Model	19
3.2 Absolute Conic	20
3.3 Lens Distortion	22
3.4 Closed-form Solution	23
3.5 Least-squares Minimization	25
4 Contour Measuring System	26
4.1 Resolution	26
4.1.1 Welded Plate Specimen	27
4.1.2 Aluminum specimen	30
4.2 System Parts	32
4.2.1 Camera	33
4.2.2 LED Line Projector	35
4.2.3 Linear Positioning System	36
4.3 System Configuration	38
4.4 System Calibration	39
4.4.1 Camera Calibration	40
4.4.2 LED Plane Calibration	42
4.4.3 Speed Calibration	43
4.4.4 3D Point Clouds	45

5	Results	48
5.1	Actual Depth Resolution	48
5.2	CT specimen	49
5.2.1	3D Point Clouds	49
5.2.2	Residual Stress Field Comparison	60
5.3	AISI D2 Plate	60
5.3.1	3D Point Clouds	61
5.3.2	Residual Stress Field Comparison	62
6	Conclusion	64
6.1	Concluding Remarks	64
6.2	Future Works	65
	Bibliography	66
A	Calibration Algorithm	69
B	Welded Plates specimen	83
C	Correction of the rigid rotations	99
D	CT specimen	104

Acronyms and Symbols

Acronyms

RS	Residual Stress
HS	High-Speed
EDM	Electrical Discharge Machining
wEDM	wire Electrical Discharge Machining
FE	Finite Element
CMM	Co-ordinate Measuring Machine
LSP	Laser Shock Peening
FEA	Finite Element Analysis
CT	Compact Tension
MP	Megapixel
FWD	Free Working Distance
DOF	Depth of Field
LOME	Optics and Experimental Mechanics Laboratory
fps	frames per second

Symbols

E	Young's modulus
ν	Poisson's ratio
σ	stress
l	beam length
h	beam thickness
g	beam deflection
$\varepsilon_1, \varepsilon_2, \varepsilon_3$	principle strains
$\bar{a}, \bar{b}$	geometric constants
$\sigma_{max}, \sigma_{min}$	maximum and minimum stresses
σ_{IS}	hole-drilling induced stress
d	distance between atomic planes
d_0	stress free spacing
θ	diffraction angle
λ_d	wavelength
t_f	flight time
σ_{zz}	surface normal stress
M	3D point
$\tilde{M}$	augmented M vector
$\tilde{m}$	image augmented vector

s	arbitrary scalar factor
A	camera intrinsic matrix
P	projection matrix
α, β	image axes scale factors
u_0, v_0	principal point coordinates
γ	image skew
R	rotation matrix
t	translation vector
Ω	absolute conic
ω	absolute conic image
$\check{x}, \check{y}$	distorted image coordinates
$\delta x, \delta y$	distortions
$\check{u}, \check{v}$	distorted image pixel coordinates
d_x, d_y	distances between adjacent pixels
K_i	radial distortion coefficients
P_i	decentering distortion coefficients
λ	arbitrary scalar
H	homography
vel_x, vel_y, vel_z	speed along the respective direction
f_{rate}	acquisition frame rate
ϕ	triangulation angle
ϕ_c	ϕ complementary angle
sz_{px}	pixel size
F	focal length (mm)
θ_{res}	angular resolution
s_x, s_y	number of pixels along the respective direction
res	depth resolution
σ_y	yield strength

List of Figures

2.1	In-depth of residual stress in 2014-T6 aluminum alloy produced by conventional (CS) and ultrasonic (US) shoot peening. (Mechanical Engineering, 2016)	4
2.2	Curvature methods of measuring residual stress: a) layer deposition; b) layer removal. (Withers and Bahdeshia, 2001)	7
2.3	Sectioning method. (Rossini et al., 2012)	8
2.4	Strain gauge rosette for the hole-drilling method a) and the ring-core method b). (Sargal and Mendal, 2013)	10
2.5	The 5 steps of the Deep-hole method. (George and Smith, 2005)	11
2.6	a) Diffraction from crystal lattice planes b) Diffraction peak. (Stresstech, 2020) .	12
2.7	Bueckner's superposition principle. (Weber et al., 2014)	15
2.8	Comparison between the results of a touch probe and a laser scanning system. (Prime et al., 2004)	18
3.1	Pinhole camera model. (Zhang et al., 2003)	20
3.2	Absolute conic and its image. (Zhang et al., 2003)	21
4.1	Flowchart describing the resolution <i>a priori</i> study.	26
4.2	Overall view of the Welded Plate ABAQUS model.	27
4.3	Residual stress distribution at mid thickness for the Z axis study of the Welded Plate specimen.	28
4.4	Residual stress distribution at mid thickness for the XY plane study of the Welded Plate specimen.	29
4.5	Cross-section view of the Aluminum specimen ABAQUS model.	30
4.6	Residual stress distribution at mid thickness for the Z axis study of the Aluminum specimen.	31
4.7	CMM data distribution over the Aluminum specimen cut surface.	32
4.8	UI337 uEye iDS Camera	33
4.9	Scheme of the lens and the calculated parameters	34
4.10	Scheme of the lens assembly	35
4.11	EFFI-Lase V2 LED Projector	35
4.12	Laser line (on the left) vs LED line (on the right)	36
4.13	UT100 Linear Table	37
4.14	TL78 Control Unit	38
4.15	Possible system's configurations. (MVTec)	38
4.16	System configuration	39
4.17	Image of ruler numbers under sharp focal conditions.	41
4.18	Sample of 4 images used to perform camera calibration.	41
4.19	Re-projection of the checkerboard intersection points	42

4.20	System extrinsic parameters visualization.	44
4.21	Sample of images captured during the speed calibration stage.	44
4.22	System displacements along X,Y and Z directions.	45
4.23	Sharp LED line contrasting with dark background.	46
4.24	Pixels intensity of a saturated LED line (A) and a not saturated LED line (B).	46
5.1	Triangulation scheme showing triangulation angle (ϕ) and its complementary (ϕ_c)	49
5.3	c1A point cloud.	50
5.2	CT specimen details.	51
5.4	Stage 2 of c1A point cloud.	51
5.5	c1A point cloud in stage 3 versus its respective CMM point cloud.	52
5.6	c1A point cloud in stage 4 versus its respective CMM point cloud.	52
5.7	Thorlabs standard bases.	53
5.8	Camera fastening.	54
5.9	c2A point cloud in stage 4 versus its respective CMM point cloud.	54
5.10	c2A and c2B point clouds in stage 5 versus their respective CMM point clouds.	55
5.11	Fastening of the linear tables.	56
5.12	Displacements registered along the 3 directions.	57
5.13	c3 point clouds in stage 5 versus their respective CMM point clouds.	57
5.14	c3 point clouds in stage 6 versus their respective CMM point clouds.	58
5.15	c3 point clouds in stage 6 versus their respective corrected CMM point clouds.	58
5.16	Displacement comparison along the cut plane.	59
5.17	Residual Stress comparison along the cut plane.	60
5.18	AISI D2 plate specimen details.	61
5.19	Scanned point clouds versus their respective corrected CMM point clouds.	62
5.20	Residual Stress distribution along the cut plane.	62
5.21	Residual Stress comparison along the cut plane.	63

List of Tables

2.1	Summary of the methods to measure residual stress.	6
4.1	Error determination for the decimal digit reduction along the Z axis of the Welded Plate specimen.	28
4.2	Error determination for the decimal digit reduction along the XY plane of the Welded Plate specimen.	29
4.3	Error determination for the decimal digit reduction along the Z axis of the Aluminum specimen.	30
4.4	uEye Camera main specifications	33
4.5	MachVis input parameters	34
4.6	Optical specifications of Effi-Lase V2	36
5.1	res value for different ϕ	48
5.2	CT specimen properties	50
5.3	AISI D2 properties	61

Chapter 1

Introduction

1.1 Motivation, Problem Statement and Goals

Residual stresses (RS) are caused by non-uniform deformation gradients in the material, created by manufacturing processes such as: machining, forging, extrusion, among others. These stresses remain in place after the relief of the load that caused them and can be advantageous or detrimental for the life of the component. Advantageous RS can be intentionally caused, for example, through laser shock peening (LSP), but in most cases its presence is unwanted, since safety indicators such as fatigue and rupture strength are negatively affected by it. Therefore, it is mandatory to know the RS field on a part/structure to optimize the manufacturing process.

Several techniques have been developed over the years to measure this type of stresses. The Contour Method, which is one of the most recently developed, stands out because it provides a 2D map of residual stresses along a surface, in a relatively fast and low cost way. It is a destructive technique that involves cutting the part to release the installed stress and acquiring the contour of the surface from which the 2D stress map is to be obtained, traditionally using a coordinate measure machine (CMM), which registers the 3D coordinates of each point by a laser or a touch probe. As its name suggests, the Optics and Experimental Mechanics Laboratory (LOME) dedicates part of its resources to study themes related to optics, and as such, has challenged students to develop an alternative contour measuring system using a Line Laser Scanning system. Therefore, the main tasks of this project are: the design and assembly of a measuring system for surface topography measurement, the adaptation of a MATLAB algorithm for measurements interpretation and calculation of the residual stress distribution and, finally, testing the entire system with experimental measurements, taking CMM measurements as benchmark.

The author of this work felt motivated by this challenge since it merges both the Solid Mechanics area, which is of his personal interest, and an area that until now was entirely unexplored by him, Optics, which is not approached in any of the mechanical engineering subjects in Faculty of Engineering of the University of Porto.

1.2 Dissertation Synopsis

This work is divided into six chapters: an introductory chapter, two chapters describing the theoretical framework, a chapter focused on the experimental development, a chapter of analysis and discussion and finally, the conclusion.

The present chapter, introduction, is divided into two parts: the first part, which explains the origin and definition of the problem, lists the objectives defined *a priori* and explains what motivated the author, and the second part, which presents the structure of this document, explaining each one of its chapters.

Chapter 2 is dedicated to a brief theoretical framework about the main theme of this project, residual stress, presenting its definition, origin and different types. In addition, it lists and describes several methods of measuring residual stress, classifying them as non-destructive, semi-destructive and destructive, since the Contour method fits in the last mentioned class.

Chapter 3 makes a short introduction to camera calibration from the algebraic point of view, describing the model considered in the present work, the Pinhole model, and presenting the necessary steps to determine the intrinsic and extrinsic parameters of the camera, considering or not lens distortions.

Chapter 4 describes the experimental steps developed throughout the project. First, it describes the procedure to determine the depth resolution for the measurement system to be developed, using CMM topographies made available by LOME. Then, it focuses on the components needed to build the system, justifying the choice of each one. Having the components chosen, it presents three possible alternatives for the system's configuration, from which only one is chosen and explained. Finally, it explains the system calibration, which includes camera, laser and speed calibration, and culminates by the acquisition of the 3D point cloud.

Chapter 5 refers to the results acquired and their discussion, addressing first the results obtained with a CT specimen and then with a specimen provided by Quantal S.A. Both sections begin by introducing the studied specimen, followed by the explanation of the acquired point clouds and improvements made in each attempt, then a comparison is made between the residual stress distribution determined by CMM and by the measuring system developed, ending with a brief critical analysis of the obtained results.

Finally, chapter 6 makes a general overview of the developed work, considering difficulties overcome and difficulties yet to, summarizing the obtained results and making brief conclusions upon them. Moreover, suggestions for improvements are presented in future works.

Chapter 2

Introduction to Residual Stress

The majority of the manufacturing processes involving material deformation, heat treatment, machining or processing operations originate residual stresses, which are those that remain in a material or body as a consequence of these processes, in the absence of external forces or thermal gradients, (Mechanical Engineering, 2016; Withers and Bahdeshia, 2001).

"Residual stress is that which remains in a body that is stationary and at equilibrium with its surroundings." (Withers and Bahdeshia, 2001)

Measuring the in-service stress and comparing it to the yield stress using failure criteria is a key factor to keep the component safety, however, sometimes the combination with residual stresses shortens component life leading to its failure before reaching the yield strength, which makes this procedure insufficient. As will be shown in the following section, there are different methods that detect different types of residual stress and that are classified as destructive or non-destructive methods, (Withers and Bahdeshia, 2001).

The presence of residual stresses can be irrelevant or harmful, but can also be beneficial, as the creation of areas of compressive residual stress is a common way to counter the tensile stresses necessary for crack nucleation and growth. In aeronautics, for instance, laser shock peening (LSP) is used on structural parts of the air crafts, by applying a layer of compressive residual stress, (Veitch and Laprade, 2). Therefore, residual stresses can accelerate or delay the start of plastic deformation or decrease the fatigue crack growth rate, (Panontin and Hill, 1996; Withers and Bahdeshia, 2001).

2.1 Origin and Types of Residual Stresses

Residual stresses origins may be classified as:

- Differential plastic flow;
- Differential cooling rates;

- Phase transformation with volume changes.

Negative consequences generally come from the presence of tensile residual stresses, that are often the major cause of fatigue failure and stress-corrosion cracking. On the other hand, compressive residual stresses are linked to beneficial effects, because they prevent fatigue crack initiation, decrease their propagation rates, and increase wear and corrosion resistance. There are several ways to induce compressive residual stresses, such as shot peening, hammer peening, laser shock peening, ultrasonic peening and cavitation peening. Figure 2.1 shows the distribution in-depth of the compressive residual stresses produced in the surface of a 2014-T6 aluminum alloy, comparing the conventional and the ultrasonic peening, (Mechanical Engineering, 2016).

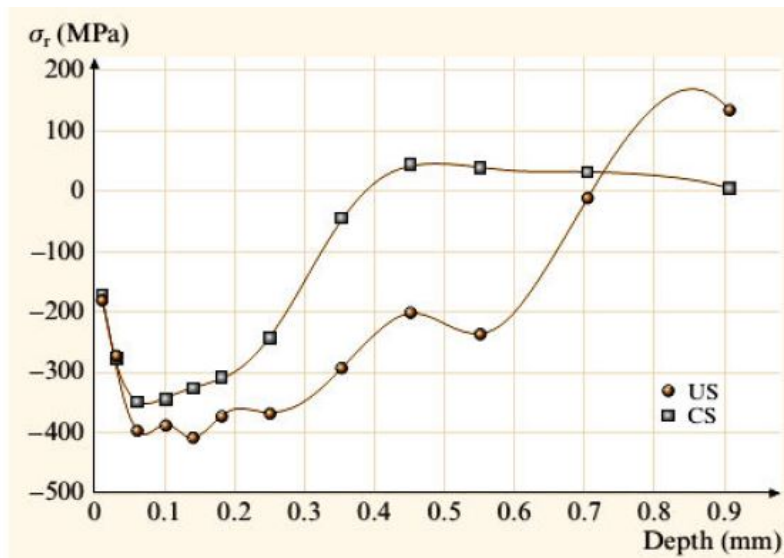


Figure 2.1: In-depth of residual stress in 2014-T6 aluminum alloy produced by conventional (CS) and ultrasonic (US) shoot peening. (Mechanical Engineering, 2016)

Residual stresses can be grouped according to the scale over which they self-equilibrate and thereby are classified as: **Type I**, which represents macro residual stresses that self-equilibrate over a considerable fraction of component, on a scale larger than the grain size of the material, and so they vary continuously over large distances; **Type II** represents micro residual stresses that self-equilibrate over the grain scale, where the misfitting regions span microscopic dimensions. Type II stresses are very common in polycrystalline materials, because grains with different orientations have different elastic and thermal properties, the microstructure is composed by several phases or phase transformations occur; **Type III** represents micro residual stresses that self-equilibrate over the atomic scale, usually resulting from the presence of dislocations and other crystalline defects, (Mechanical Engineering, 2016; Withers and Bahdeshia, 2001).

2.2 Measurement Methods of Residual Stresses

There are several techniques to measure residual stress and most of them are applied for type I since, due to their macro dimension, are the most studied by engineers, (Mechanical Engineering, 2016). The methods that will be described next are divided into destructive, semi-destructive and non-destructive methods and are summed up in table 2.1. Destructive and semi-destructive techniques are based on material removing, which will relieve the residual stresses, measurable through the strain field. These methods should not be used in structural components, since these ones should be inspected periodically and they can not have material removed, which is why the non-destructive methods are preferable in such case, (Rossini et al., 2012).

The next subsections present and describe the different measuring methods approached in table 2.2, including applicability, advantages and disadvantages. The contour method, as the main topic of this work, is the last one explained.

Table 2.1: Summary of the methods to measure residual stress.

	Method	Penetration	Accuracy	Comments
Non-destructive	X-ray diffraction	< 50 μm (Al) < 5 μm (Ti) < 1 mm (w/ layer removal)	± 20 MPa	surface technique; applicable to crystalline and small grain materials;
	Neutron diffraction	200 mm (Al) 25 mm (Fe) 4 mm (Ti)	$\pm 50 \times 10^{-6}$ strain	can evaluate internal residual stresses; the beams can be produced by a reactor or by spallation;
Semi-destructive	Hole-drilling	< hole diameter	± 50 MPa	measures in-plane type I stresses;
	Deep-hole drilling	>7 mm and <1000 mm	± 30 MPa for $E = 200$ MPa	Combines deep-hole and ring-core methods;
Destructive	Curvature	0.1 to 0.5 of thickness	Limited by minimum measurable curvature	Stress variation is shown as a function of the deposit thickness;
	Sectioning	the cutting line	± 1.38 MPa	a complete sectioning enables the determination of the residual stress distribution over a cross section;
	Contour	the cutting area	Depends on the measurement technique	only measures the stress normal to the cutting plane; does not provide reliable results near the perimeter of the cutting area;

2.2.1 Curvature

This method is usually employed to determine stresses on thin plates by measuring the curvature caused by adding or removing layers of material containing residual stresses, (Rossini et al., 2012). Figure 2.2a) shows the curvature originated due to the stresses induced by the layer deposition. While depositing, the curvature changes and that change can be registered, showing the stress variation as a function of the added plate thickness. On the other hand, figure 2.2b) shows the curvature originated due to layer removal, which has been done for metallic and polymeric composites and for thin coatings, (Withers and Bahdeshia, 2001).

Curvature is measured on narrow strips to avoid multi-axial results and mechanical instability, and it can be performed either using contact methods, as shown in figure 2.2 with a strain gauge, or using methods that do not require contact, (Withers and Bahdeshia, 2001).

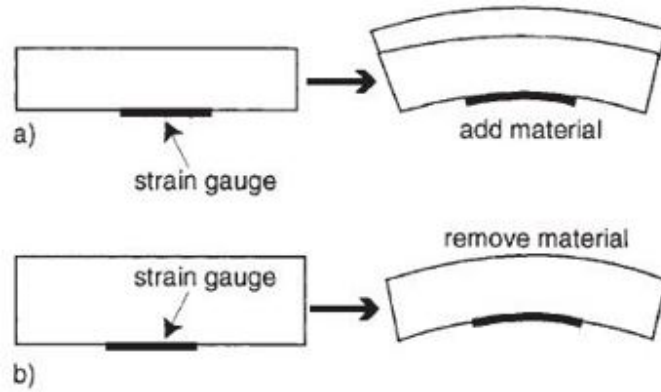


Figure 2.2: Curvature methods of measuring residual stress: a) layer deposition; b) layer removal. (Withers and Bahdeshia, 2001)

The expression 2.1 shows the Stoney equation, often used to describe the deflection g of a beam to its stress σ .

$$\sigma = -\frac{4}{3}E \frac{h^2}{l^2} \frac{dg}{dh} \quad (2.1)$$

where l is the length of the beam, h is its thickness and E is the Young's modulus of the beam's material.

2.2.2 Sectioning method

Sectioning is a destructive method and consists in cutting off a section of the test piece and measuring the consequent deformation on the cutting line, (Rossini et al., 2012).

The simpler way to explain this method is through the example of a bar. Cutting off a stripe of a bar, longitudinal stresses will be relieved on it and this will originate a length change, which can be used to determine the residual stress through electrical or mechanical strain gauges. The stress distribution over a cross section is usually considered negligible, but if necessary, figure 2.3 shows a scheme of a complete sectioning on an I-beam. In the case of the I-beam, both axial deformation

and curvature are exhibited, corresponding to membrane residual stresses, linked to hot rolled and fabricated sections, and bending residual stresses, dominant in cold formed sections, (Rossini et al., 2012; Tebedge et al., 1972).

Figure 2.3 illustrates the three main steps of the present method. First, a basis of measurement must be created by marking parallel points all around the steel plate. These points produce distances that if measured before cutting create the initial values. Then, the specimen is sawed from the initial plate, ending the first step. Second step consists of cutting the specimen in strips to perform its complete sectioning. At this point, the basis of points is measured again to obtain the change of length of each strip. This way, the distribution of longitudinal strains over the cross section is determined and by applying Hooke's Law the stress distribution may also be. Moreover, the strips thickness must be adjusted depending on the present stress gradient, as the strips should be thinner in locations where there is a residual stress gradient. The third and last step consists of slicing each strip, in order to obtain the residual stress distribution through the thickness by measuring the strain change of each slice. A base measurement must be implemented on each side of each strip. (Thiébaud and Lebet, 2012)

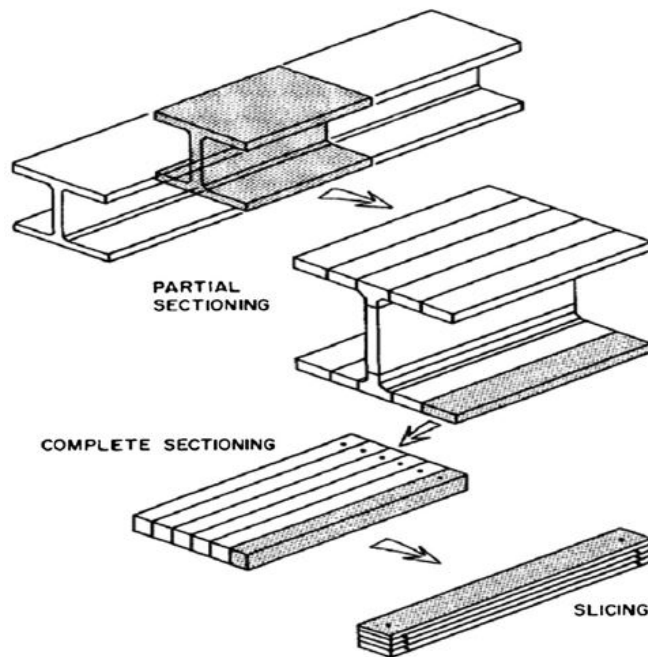


Figure 2.3: Sectioning method. (Rossini et al., 2012)

The cut performed on the specimen must be executed carefully, in order to prevent the introduction of plasticity or heat, so there will be no influence on the measuring process, thereby decreasing transverse stresses and improving the accuracy of the measurement, (Rossini et al., 2012; Tebedge et al., 1972).

Overall, the sectioning method is mainly suitable for elements in which the longitudinal stress is dominant. Moreover, as the Contour Method, Sectioning method also provides a 2D RS map of

a cross section, however, the specimen is completely destroyed in the latter one, allowing a single measurement only.

2.2.3 Hole-drilling method

This method consists in drilling a small hole into the analyzed structure or part, which will cause stress relaxation around the hole, determined through the lateral strain measurement. The strains are measured nearby the hole and upon the surface where it was drilled. Figure 2.4a) shows the configuration of the triple strain gauge rosette needed to measure the in-plane stresses existing in the material and equation 2.2 enables the calculation of the maximum and minimum stresses, introducing the principal strains $\varepsilon_1, \varepsilon_2, \varepsilon_3$ and considering the stress field uniform with depth, which may not be the case, (Johnson, 2008). E and ν represent the Young modulus and the Poisson coefficient, respectively, while $\bar{a}$ and $\bar{b}$ are tabulated constants that are geometrically determined, (Withers and Bahdeshia, 2001).

$$\sigma_{max}, \sigma_{min} = -\frac{E}{2} \left(\frac{\varepsilon_3 + \varepsilon_1}{(1+\nu)\bar{a}} \mp \frac{\sqrt{(\varepsilon_3 - \varepsilon_1)^2 + (\varepsilon_3 + \varepsilon_1 - 2\varepsilon_2)^2}}{\bar{b}} \right) \quad (2.2)$$

In spite of consisting in material removal, the drilling-method is considered a semi-destructive technique, because the drilled hole dimensions are small, around 1.8 mm diameter, and so the damage caused is often tolerable or repairable, (Rossini et al., 2012). When compared to other techniques, hole-drilling determines the type I stresses and is applicable to isotropic and machinable materials, since their elastic parameters are known, (Rossini et al., 2012). Moreover, the hole-drilling method measures stresses along both in-plane directions, however, each measurement provides those stresses on a single point only.

The experience of the operator is very important because there are many potential systematic errors that can be introduced to the measuring process in the drilling action, such as the introduction of machining stresses, non-cylindrical hole shape and eccentricity, (Rossini et al., 2012). Adding to this, the technique has other weaknesses: the drilled hole causes a stress concentration that limits measurement to 50% of the yield stress before plastic deformation occurs; the drilling's depth is limited to the dimension of the hole diameter, due to the decreased sensitivity of the strain gauges at increasing depths, (Johnson, 2008). However, this technique has the advantage of being easily used in the field with standard equipment and the requirements for drilling accuracy are lessened by the usage of laser speckle interferometry and moiré fringe techniques, instead of the strain gauge rosettes to measure the strains around the hole, which makes this a widely used technique, even though the measurements stability in the field require the usage of strain gauges, (Johnson, 2008).

The ring-core method is very similar to the hole-drilling one, but has one big difference, it consists in measuring the resulting deformation in a central area caused by the cutting of an annular slot in the surrounding material, as shown in figure 2.4b), (Rossini et al., 2012). The method carries out removing material in depth and as happens in the hole-drilling technique, ring-core has a basic formulation to evaluate in-plane stresses and an incremental one to determine the stress profile,

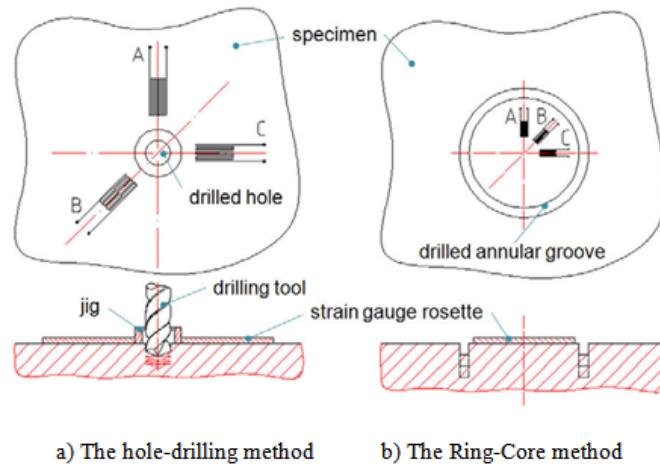


Figure 2.4: Strain gauge rosette for the hole-drilling method a) and the ring-core method b). (Sargal and Mendal, 2013)

where the strains are measured for each chosen depth. The ring-core method provides a larger depth limit and a complete stress relief in the pedestal, however, it creates a greater specimen damage and adds an extra degree of experimental difficulty, as the strain gauges must be detached and reattached for every drilling increment, (Rossini et al., 2012; Johnson, 2008).

There are other variants of the hole-drilling technique, such as high-speed (HS) hole-drilling, which introduces lower additional induced stress. HS hole-drilling provides results with a better accuracy. HS hole-drilling is also better suited for specimens with high hardness and toughness. However, the drill will be subjected to severe wear, causing the induced stress to increase and the tool to crack, in extreme cases. The electrical discharge machining (EDM) process, on the other hand, has no constraint on mechanical properties of ferrous materials, and has proven its capability to drill highly precise holes on various metals, (Rossini et al., 2012). Ghanem et al. (2003) have shown that EDM creates a tensile residual stress within its transformation layer, and Tekkaya et al. (2006) developed a modified empirical equation to scale the residual stress in the machined surface, which increases from the surface until reaching a maximum value around the tensile strength, and then falling to zero or to a small compressive residual stress. Part of the released strain measured by the gauge comes from the residual stress induced by the hole-drilling process, which introduces a measurement error. Lee et al. (2008) have shown that the EDM method has the same degree of measurement reliability and stability of the HS method, and they also found that the induced error by EDM depends on the working parameters applied. For that reason, the authors suggested the accuracy of the residual stresses measurements could be improved by calibration using the hole-drilling induced stress, σ_{IS} .

Using a high-speed pneumatic drill ensures that no further machining stresses are induced and so, the strain data acquisition precision is higher. The greater the number of drilling increments for the same laminate thickness, the more accurate the stress profile will be, as the precision of the method depends on the number of increments and respective depths, (Rossini et al., 2012).

2.2.4 Deep-hole method

This method is a combination of the hole-drilling and the ring-core methods, since it carries out the drilling of a reference hole and the removal of a ring, afterwards.

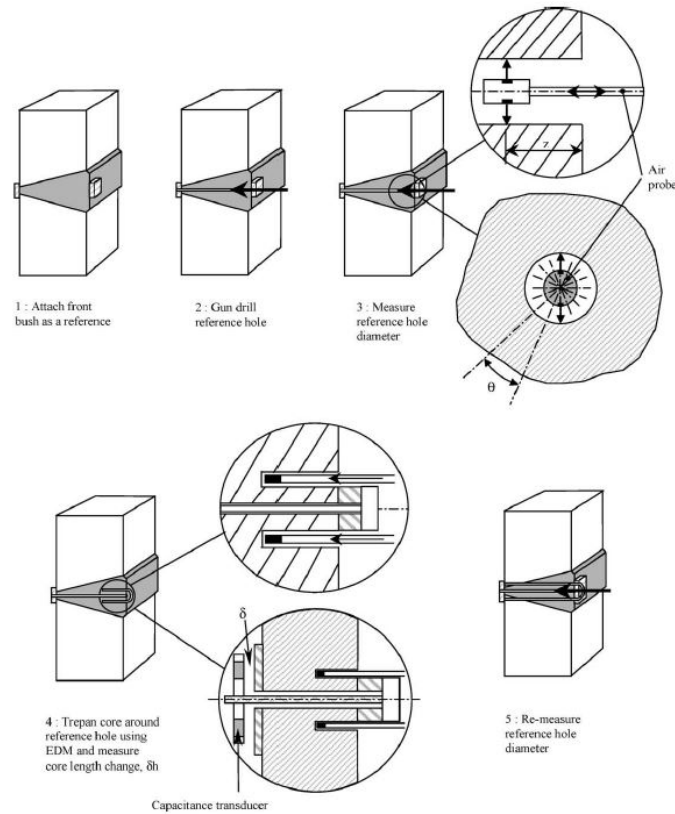


Figure 2.5: The 5 steps of the Deep-hole method. (George and Smith, 2005)

The deep-hole method consists of five steps, which are illustrated in figure 2.5. First, reference blocks are attached to the component, which ensure that there will be alignment and initial material for the drilling. Afterwards, a gun-drill is used to open a 3.175 mm diameter hole completely through the specimen, because it provides a straight hole and a good surface finish. The third step consists in measuring the diameter of the reference hole, introducing an air probe in it. Measurements are made every 0.2 mm in depth and at every 10° angle around the hole. Then, trepanning out a 20 mm diameter core containing the reference hole as its axis follows this. In the final step, the deformed hole is re-measured by the air probe and measured distortions are registered, (George and Smith, 2005). Using an elastic analysis that assumes that there is no plastic deformation when residual stresses relax during the fourth step, one can easily calculate the residual stresses, (George and Smith, 2005).

As a combination of two techniques, the deep-hole method is also considered a semi-destructive technique, since the damage dealt is not relevant to the structure. This method can be used to measure deep interior stresses, since the trepanning can be carried out incrementally. This way, it is a suitable method for thick and heavy specimens, (Rossini et al., 2012).

2.2.5 X-ray and neutron diffraction

Generally, the destructive and semi-destructive methods use strain gauges to measure the deformation caused by the material removal, but the diffraction based methods are non-destructive methods. Diffraction based methods use the diffraction from the crystal lattice planes as a measuring principle. When an array of scattering objects is hit by a wave, diffraction occurs, providing that the distance between the objects is of the same order of magnitude as the length of the incoming wave, (Johnson, 2008). The collision of the wave results in many scattered waves, which originate either destructive or constructive interference, depending if they are in phase or out of phase with the incident one, respectively. The incoming and diffracted phase waves create constructive interference, and these are described by Bragg's Law, given by equation 2.3, (Johnson, 2008; Stresstech, 2020).

$$n\lambda_d = 2d\sin(\theta) \quad (2.3)$$

Bragg's Law relates the distance between atomic planes d and diffraction angles θ to the incoming wave's wavelength, λ_d , and its multiples $n\lambda_d$. In a microscopic point of view, there is no deformation due to RS, however, when a metal is under applied or residual stress, the resulting elastic strains cause the atomic planes in the metallic crystal structure to change their spacing. Then, the induced strain, which corresponds to changes in lattice spacing, can be determined either by varying the wavelength of the incoming radiation or by observing the diffracted wave from different angles. Figure 2.6a) shows an illustrating scheme where it is noticeable that a wave goes through the lattice or not, depending on the controlled parameters, the wavelength multiple $n\lambda_d$ and the observation angle θ , (Stresstech, 2020).

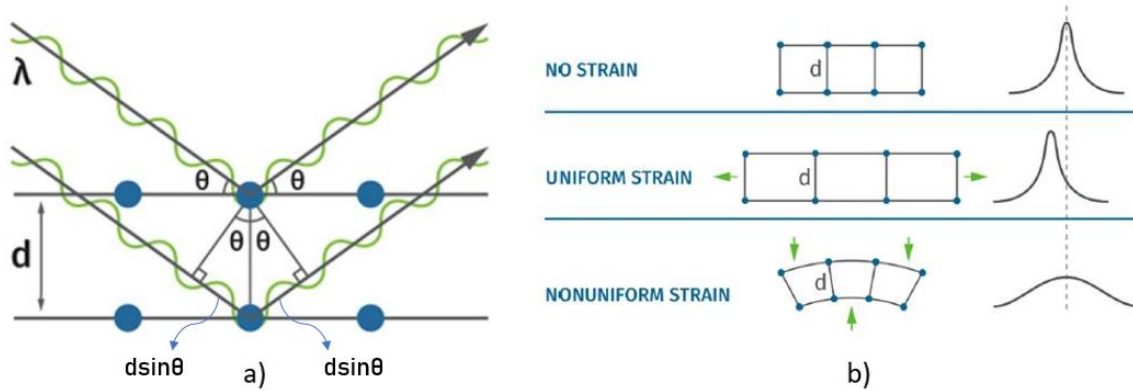


Figure 2.6: a) Diffraction from crystal lattice planes b) Diffraction peak. (Stresstech, 2020)

In a poly-crystalline structure with disordered crystals at the grain boundaries, precipitations and lattice defects, the diffraction line forms a Gaussian-like peak. Figure 2.6b) shows the curve and what happens to it while the lattice is strained. As shown, while the lattice is subjected to uniform strain, the curve is shifted and so is the diffraction peak, and while subjected to non-uniform strain, both peak shape and position change, (Stresstech, 2020). Having an accurate measure of the

stress free spacing d_0 and of the change in theta $\Delta\theta$, we can use equation 2.4 to obtain the strain ϵ and consequently the stress, using Hooke's Law together with elastic modulus E and Poisson's ratio ν , (Johnson, 2008). Stress calculation is affected by parameters such as differences in lattice parameters, precipitations, interstitial occupation, and micro stresses, (Stresstech, 2020).

$$\epsilon = \frac{\Delta d}{d_0} = -\cot(\theta)\Delta\theta \quad (2.4)$$

According to Withers and Bahdeshia (2001), in the diffraction method the shift in samples for both type I and type II stresses for a particular grain set and type III stresses, only give peak broadening. In multi-phase materials, this can provide information about individual phases in separate, but in single-phase materials presenting type II stresses, the elastic strain recorded for a given reflection may not be representative of the bulk elastic strain. Type I stresses are those of interest in this work, but with diffraction they can only be determined after applying corrections to the inter-granular stresses, (Withers and Bahdeshia, 2001).

There are many ways to estimate the value of d_0 , such as measuring it in a stress free section, measuring it in a powder from the sample material, measuring it in a comb-like structure cut out of the material or with measurements through the depth of the sample, (Johnson, 2008).

The X-ray method is applicable to materials with a crystalline structure, with small grains, known elastic constants and that produce diffraction for any orientation of the sample surface, and can directly measure the inter-planar atomic spacing, (Stresstech, 2020)(Rossini et al., 2012). The sample may be metallic or ceramic as soon as the available radiations can produce a diffraction peak of suitable intensity and free of interference from neighboring peaks, in the back reflection region, (Rossini et al., 2012).

When using this technique to determine the residual stresses in large welds, the limited space available on most beam lines or X-ray diffractometers makes it necessary to cutdown the samples to proceed with the measuring process. The geometry has to allow the X-rays to hit the measurement area and still be diffracted to the detector without hitting any obstructions. The stress profile can be generated combining with a layer-removal technique, but that would turn the method into a destructive one and would imply a careful cutting process, in order to avoid interference in the stress state. Diffraction techniques are not easy to apply to a nanostructured material because it is difficult to pinpoint its peak location or to determine its peak shift, which helps studying microstresses. Therefore, mechanical methods are the most reliable to study residual stresses in nanostructured materials. Measurements can be quicker or slower, depending on the X-ray source, the type of examined material and the degree of required accuracy, and the gauge volume is a trade-off between the need for spatial resolution and the time available for the operation, (Rossini et al., 2012).

The fact that synchrotron sources provide access to high X-ray energy makes the attenuation length increase which, combined with the very high X-ray intensities, leads to path lengths of centimetres. These kind of sources present strengths such as high intensity and the high collimation of the beam, which allow data acquisition rates of the order of seconds. The intense beams

available offer unparalleled spatial resolution lateral to the beam and fast data acquisition times, which make the method well suited to the collection of detailed maps of the strain field in two or three dimensions, or to check phase transformations, where neutron diffraction would be obsolete. On the other hand, this method also presents weaknesses such as the low scattering angles, which means that the sampling gauge is usually very elongated and so the spatial variation is very poor along the beam. Moreover, it does not have a good performance in materials with large grain size or with high degrees of texture, because the small gauge volume gives a poor statistical measure of type I stresses, (Rossini et al., 2012; Johnson, 2008).

Overall, X-ray diffraction measures the residual stresses on the surface of materials and is applicable to materials that are crystalline, relatively fine grained, and produce diffraction for any orientation of the sample surface. (Rossini et al., 2012)

Neutron diffraction is very similar to the X-ray method because they are both dependant on changes in the spacing of the lattice planes, that come from their stress-free condition. However, the penetration of neutrons into engineering materials is greater and so it can be applied to the evaluation of internal residual stress of materials, contrarily to the X-rays, (Rossini et al., 2012) (Johnson, 2008). The wavelength of thermal neutrons and the spacing of the lattice planes of most materials are similar, which results in a diffraction angle of approximately 90° , while synchrotron X-rays offer approximately 6° . Regarding the second term of equation 2.4, better strain resolutions are produced, (Johnson, 2008).

There are also two neutron diffraction techniques, the conventional $\theta/2\theta$ scanning where the beams are produced from a nuclear reactor and time of flight approaches where neutron beams are produced by spallation, which is a process where a proton collides with a heavy metal target. In the latter one, the diffraction profile is not collected as a function of the Bragg angle θ , instead Bragg angle is held constant ($2\theta = 90^\circ$). The most energetic neutrons arrive at the specimen first and the least energetic last, thus, the wavelength of each detected neutron can be deduced from the flight time, t_f , and the strain is given by $\varepsilon = \Delta t_f / t_f$, (Johnson, 2008) (Withers and Bahdeshia, 2001).

Choosing between the two neutron diffraction involves deciding between a measure of all diffracting neutrons using a single wavelength with $\theta/2\theta$ scan, and a measure of the diffracting neutrons for all wavelengths for a fraction of the time with the time of flight approach, (Withers and Bahdeshia, 2001).

2.2.6 Contour method

The contour method is a destructive method, originally proposed in 2000, that enables the determination of a 2D stress map over a plane from its relaxation, (Rossini et al., 2012). It involves three main steps explained in this section: the specimen cutting, that needs to be accomplished with the least additional stresses possible, the cut surface topography measurement and the stress determination through a FE-model, (Withers and Bahdeshia, 2001; Rossini et al., 2012).

The method is based on a superposition principle, developed by Bueckner. Figure 2.7 shows three different steps of the specimen during the method. Step A shows the specimen before being

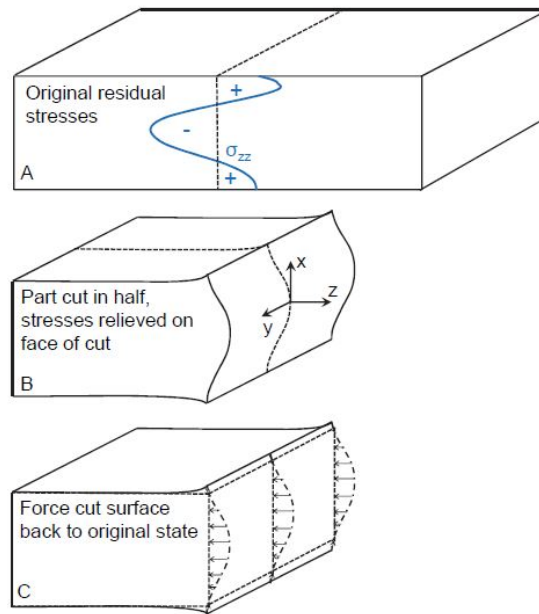


Figure 2.7: Bueckner's superposition principle. (Weber et al., 2014)

cut, with a representative distribution of the residual stresses on the plane of interest. Then, the specimen cutting takes place, relieving the residual stresses, consequently, its surface deforms according to the relief. Moreover, step C consists in forcing the deformed surface in B back to original step A, (Weber et al., 2014; Johnson, 2008). In other words, the principle says that forces applied to put surface back in place after cutting need to be equivalent to the residual stresses present before cutting, (Johnson, 2008). Equation 2.5 shows the equivalence between the stresses of the three steps. As step B represents the stress relief, $\sigma_{zz}^{(B)} = 0$ and so 2.5 results in 2.6, (Weber et al., 2014).

$$\sigma^{(A)} = \sigma^{(B)} + \sigma^{(C)} \quad (2.5)$$

$$\sigma_{zz}^{(A)} = \sigma_{zz}^{(C)} \quad (2.6)$$

The cutting process of the specimen has very specific conditions, because if done wrongly, the obtained measurements are faulty. In order to avoid introducing additional significant stresses to the specimen, it is necessary to apply a single flat cut that causes a minimum of plastic deformation. Moreover, it is intended to have the least material removal possible, so that a measurement of each new surface (opposing sides) may be made to average out shear stresses and transverse displacements. The only technique available that fulfills these requirements is the wire Electrical Discharge Machining (wEDM) technique, (Weber et al., 2014).

"Wire EDM cutting is a erosion process induced by electrical breakdown between a continuously travelling wire electrode and the work piece." (Johnson, 2008)

This technique uses electrical discharges instead of cutting tools to remove material and is preferably performed with the specimen clamped on both sides, to reduce the bulge error. The cut must be done normal to the stress component of interest, since this technique provides a 2D stress map but of only one stress component, σ_{zz} , using the axis of the figure 2.7. The cut is preferably performed through symmetrical planes of the specimen and the usage of small cutting speeds reduces the off-axis motion, (Johnson, 2008; Weber et al., 2014; Rossini et al., 2012).

Upon cutting, the residual stresses on the cut plane are released and a contoured surface is formed in each half of the specimen. The determination of the contour may be done through a co-ordinate measuring machine (CMM) that consists on a touch trigger probe that registers mechanical contact, usually a ruby-tipped stylus as a sensor for detecting a specimen surface. Alternatively, the measuring can be performed using a laser probe, which presents many advantages over the CMM process, as explained further, (Weber et al., 2014; Rossini et al., 2012).

The relaxation of the residual stresses is considered to be elastic in both directions of the cutting plane, however, the contour method is limited to measure only the stresses normal to this plane, disregarding shear stresses. Therefore, the approximation explained in step C of figure 2.7 consists in putting the surface back to its original configuration only in the normal z-direction, leaving displacements in other directions unconstrained. This means that if the residual shear stresses are originally zero along the plane of cut, then Poisson contractions will return the surface to its original position and the approximation will be exact, otherwise both halves need to be averaged, (Prime, 2001).

The surface contour data cannot, however, extend to the perimeter of the cutting surface and since all that area needs to be defined in order to apply the FE model, those points need to be extrapolated from the nearest ones. In the EDM process, a phenomenon called "barreling" causes more material to be removed near the edges of the cut than in the center, which justifies the limitation mentioned before. Moreover, the CMM's spherical tip goes past the edge of the cut, which does not always provides reliable data, (Prime, 2001). Nevertheless, the stresses out of the perimeter are not significantly affected by the extrapolation and stresses on the interpolation region are not reported, (Prime et al., 2004).

Generally, the residual shear stresses are not zero and so the results in each half must be averaged, to correctly determine the normal stress distribution over the cutting plane. Regarding both halves, it is known that normal traction stresses are symmetric with respect to the cut plane and the transverse traction stresses are anti-symmetric, consequently and regarding the elastic assumption, averaging will cancel the shear stresses and present the normal ones, (Prime, 2001). Therefore, it is possible to determine the normal stresses at the cut plane only by the normal strain measuring and averaging, disregarding shear stresses, (Johnson, 2008). Additionally, it is required to smooth the data, in order to minimize errors and to remove any noise present in the dataset, (Johnson, 2008; Rossini et al., 2012).

As previously said, the contours may be measured using a laser probe instead of CMM and some of the advantages will be shown next. Measuring the surface contour with a touch probe offers a low resolution and accuracy of the measured stresses, reaching a maximum of 1 – 3 μm ,

according to Prime et al. (2004). As the stress magnitude or the part size decrease, the peak-to-valley amplitude measured is expected to decrease and at a certain point the touch probe will have trouble acquiring the data. Moreover, touch probes present a low rate of acquiring data and locally deform parts during the measurement, due to their small contact area, (Prime et al., 2004). For the reasons mentioned before, a laser probe arises as better choice. A laser probe presents more precise and accurate measurements, specially in parts with lower stress levels. Additionally, the data acquisition rate is higher with laser contouring, making the process faster and enabling the possibility of increasing point density. Since the best touch probe CMMs have the capability of measuring 3-D fields, it makes them cost around the double of a laser contour system, which is of further advantage, (Prime et al., 2004).

Prime et al. (2004) have compared a confocal laser probe to a Brown & Sharpe XCEL 765 touch probe CMM and the results are shown in figure 2.8. Even though the measurements on both halves present the same shape, they also show different amplitude, which may have occurred due to the non-centered cut that took place. Performing an asymmetrical cut to the clamps causes the material on the cutting plane to move as soon as stresses are released, but this cause anti-symmetric deviations, which can be corrected, as previously said, averaging the two contours. The high resolution of the laser measurements implies a higher noise level in the data, (Prime et al., 2004).

Further, Prime et al. (2004) presented a comparison between the cross-sectional maps of the measured residual stresses by the contour method using either laser contouring or touch probe CMM against neutron diffraction. The results achieved a good agreement between the two contour measuring techniques, and between them and the neutron diffraction. Weber et al. (2014) also successfully compared results between the contour method and depth-corrected X-ray measurements.

After studying the measuring methods summarized in table 2.1, one may conclude that no method is ideal for every application. In order to choose the most suitable measuring method for a specific application, it is necessary to consider some aspects *a priori*, such as the type of stress to measure, its magnitude and the location of the area/point to be measured within the specimen. The Contour Method provides a 2D stress map over a cross section but it only measures the stress component normal to the surface, whereas the hole-drilling method, for instance, measures stresses in both plane directions, but point by point. The sectioning method provides the same 2D map as the Contour Method and measures the same normal stresses, however, it demands the specimen to be completely destroyed, which is more time dispensing and disables a second measuring attempt. To measure the surface topography, regarding the Contour Method, there are three possible categories: tactile, optical and electromagnetic. The tactile category is that of the CMM, and the electromagnetic domain is typified by the atomic force microscope, which has a very limited working volume and a slow speed, and for that reason it is not suitable for engineering applications, (Braga, 2012). The optical category has been mentioned along the three previous paragraphs. The aforementioned laser probe provides punctual measurements, as well as the CMM, and the difference is there is no contact between the specimen and the measuring

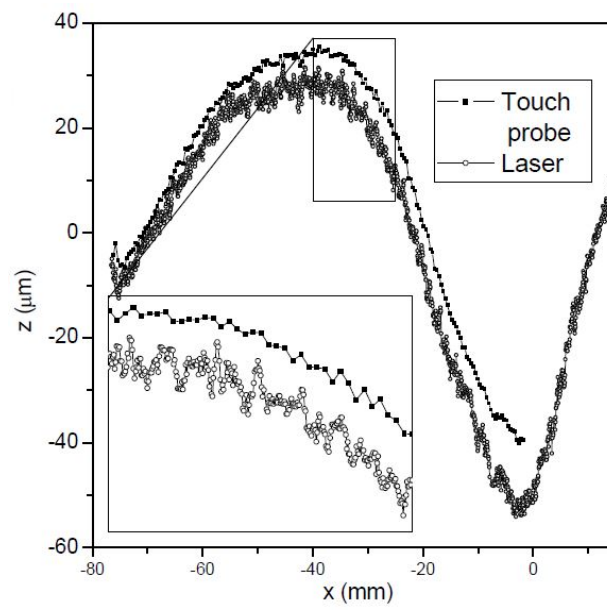


Figure 2.8: Comparison between the results of a touch probe and a laser scanning system. (Prime et al., 2004)

instrument. Having this in mind, the present work will focus on implementing a laser line scanning to measure the surface topography, which will still provide a series of discrete points, but the measurement will be performed continuously.

Chapter 3

Introduction to Camera Calibration

Camera calibration consists of determining its intrinsic and extrinsic parameters, and there are several techniques for this purpose. Regarding calibration objects, it is possible to classify those techniques into four categories: calibration with a 3D reference object, with a 2D reference object, 1D based calibration, or a calibration without a specific reference object, known as self calibration. The calibration with a 3D reference object usually offers a better accuracy when compared to a 2D based calibration, but it is not always affordable to make a 3D object, whose geometry should be known with high precision. During the present work, camera calibrations were performed using a 2D reference object, through a MATLAB algorithm, attached to Appendix A, that use *estimateCameraParameters*, a MATLAB function that assumes a pinhole camera model. Moreover, the mentioned function determines the intrinsic and extrinsic parameters in closed form, assuming that lens distortion is zero, and then estimates all the parameters simultaneously, distortion coefficients included, using the least-squares minimization, topics that will be explained further. (MathWorks, 2020c)

Note that this entire section referring to camera calibration is based on Zhang's calibration method, present in Zhang et al. (2003) and Zhang (2000).

3.1 Pinhole Camera Model

The pinhole model relates any 3D point $M = [X, Y, Z]^T$, or its augmented vector $\tilde{M} = [X, Y, Z, 1]^T$, to the augmented vector of its projected image $\tilde{m} = [u, v, 1]^T$. The camera is modeled as a pinhole, where the image of any 3D point M is formed by an optical ray coming from M to the optical center, C, and intersecting the image plane, as these three points are collinear. Figure 3.1 shows the image plane positioned between the optical center and the scene point, instead of being on the opposite side, but there is mathematical equivalence for these two physical setups, therefore, the relationship between the 3D point M and its image projection m is given by equation 3.1.

$$s\tilde{m} = A[R \ t]\tilde{M} \equiv P\tilde{M} \quad (3.1)$$

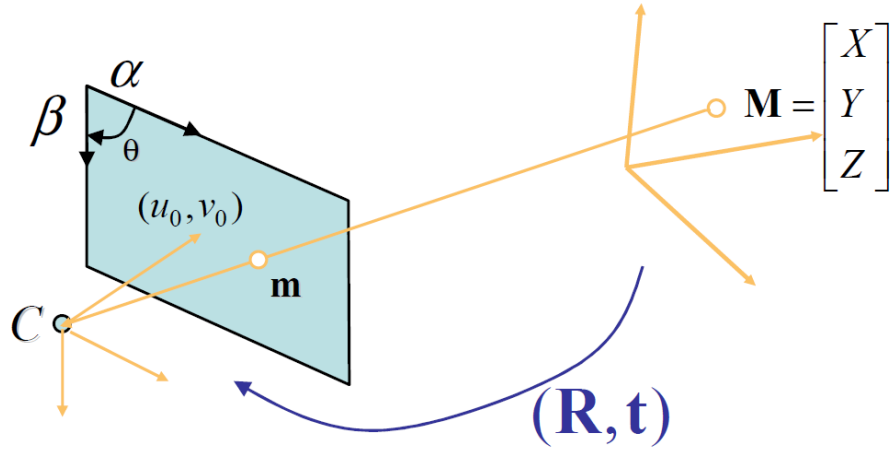


Figure 3.1: Pinhole camera model. (Zhang et al., 2003)

where s is an arbitrary scale factor, (R, t) are denominated extrinsic parameters, A is called the camera intrinsic matrix and P the camera projection matrix, that mixes both intrinsic and extrinsic parameters. The latter two matrices are shown in expressions 3.2 and 3.3, respectively.

$$A = \begin{bmatrix} \alpha & \gamma & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.2)$$

$$P = A[R \ t] \quad (3.3)$$

The intrinsic parameters, also known as internal parameters, depend exclusively on the camera characteristics, such as geometry. α and β are the scale factors of the projected image axes, u and v , and (u_0, v_0) are the coordinates of the principal point, as a result of the intersection between camera's optical axis, perpendicular to the image plane and that passes through the optical center, and the image plane itself. Moreover, the skew of the two image axes, γ , relates to α and its angle with β , θ , having $\gamma = \alpha \cot \theta$.

The extrinsic parameters, also known as external parameters, are strictly related to the orientation and position of the camera image relative to a certain world coordinate system, having R as the 3x3 rotations matrix and t as the translation vector.

3.2 Absolute Conic

This concept has a very important property, its invariance to any rigid transformation, as proved next

The absolute conic Ω is a set of points in a plane at infinity and has a very important property: its invariance to any rigid transformation. Therefore, for constant intrinsic parameters its image

is also constant and as such, it will be used later on to determine the system parameters. The Absolute Conic is defined by the following equation:

$$\begin{cases} x_1^2 + x_2^2 x_3^2 = 0 \\ x_4 = 0 \end{cases} \quad (3.4)$$

or $\mathbf{X}_\infty^T \mathbf{X}_\infty = 0$, as shown in figure 3.2.

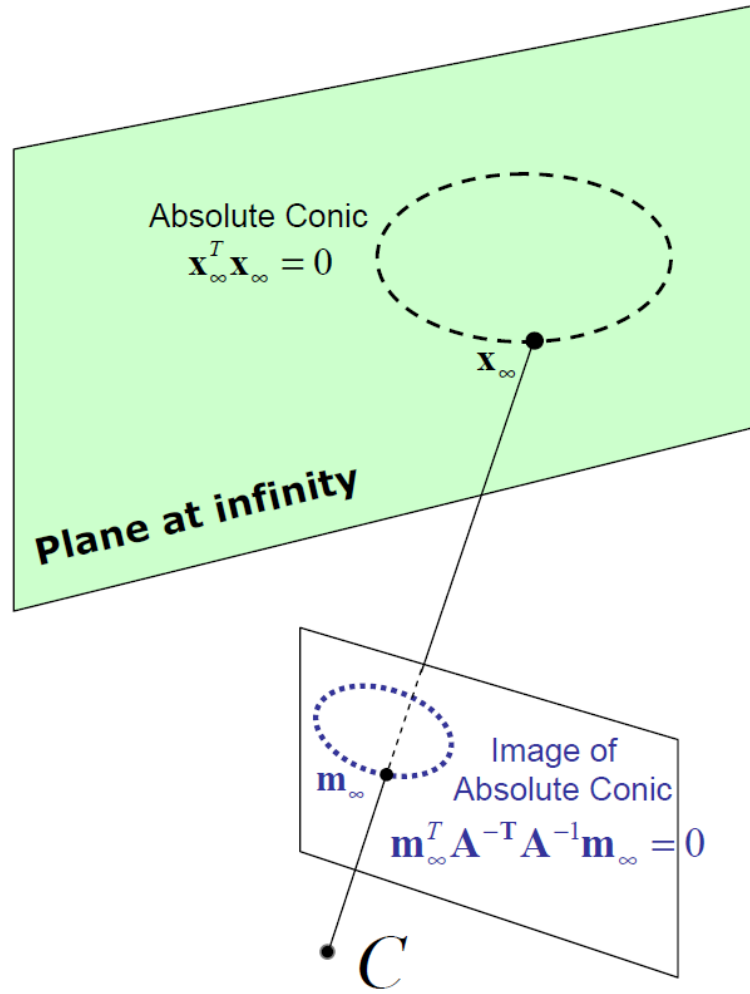


Figure 3.2: Absolute conic and its image. (Zhang et al., 2003)

If the rigid transformation is set by $H = \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix}$ and the projective coordinates of a point on Ω are defined by $\tilde{x}_\infty = \begin{bmatrix} x_\infty \\ 0 \end{bmatrix}$, the point after that rigid transformation is denoted by x'_∞ , and

$$\tilde{x}'_\infty = H\tilde{x}_\infty = \begin{bmatrix} Rx_\infty \\ 0 \end{bmatrix} \quad (3.5)$$

which means x'_∞ is also on the plane at infinity. Adding to this, equation 3.6 shows x'_∞ is on the same Ω .

$$x'^T_\infty x'_\infty = (Rx_\infty)^T (Rx_\infty) = x^T_\infty (R^T R) x_\infty = 0 \quad (3.6)$$

Moreover, the image of the absolute conic, ω , is determined exclusively by the intrinsic parameters of the camera, which means that once the image of the absolute conic is determined, the camera's intrinsic parameters are calculated and, consequently, its extrinsic parameters are also determined. Considering $\tilde{m}_\infty$ the projection of a point x_∞ on Ω and applying the reasoning performed in 3.6, comes equation 3.7:

$$\tilde{m}^T A^{-T} A^{-1} \tilde{m} = 0 \quad (3.7)$$

This way, the image of the absolute conic is an imaginary conic, defined by $A^{-T} A^{-1}$, which does not depend on the extrinsic parameters of the camera.

3.3 Lens Distortion

The pinhole camera model considers collinearity between the point in 3D space, its image projection and the optical center, as previously mentioned, but this consideration might be insufficient sometimes. As the system developed in the present work requires high accuracy on its results, lens distortion might have to be considered. Therefore, the present subsection shows how to determine the image points considering distortions.

There are two types of distortion, **radial distortion** and **decentering distortion**. The former one is due to imperfections that take place during the manufacturing process, which originates symmetric distortions along the radial directions from the distortion center. The latter one occurs due to the misalignment of the lens, that comes from a poor lens assembly, and originates ideal image points distorted in both radial and tangential directions.

Under the pinhole camera model, the effective focal length, f , is also the distance between the optical center and the image plane, thus, the projection from 3D camera coordinates to ideal image coordinates are given by the expressions 3.8.

$$x = f \frac{X}{Z}, \quad y = f \frac{Y}{Z} \quad (3.8)$$

Then, to consider the distortions applied to (x, y) , comes the expressions in 3.9

$$\check{x} = x + \delta_x, \quad \check{y} = y + \delta_y \quad (3.9)$$

where $\check{x}, \check{y}$ are the distorted image coordinates and δ_x, δ_y are the distortions applied. To transform real image coordinates $(\check{x}, \check{y})$ into pixel image coordinates $(\check{u}, \check{v})$, we use the following expressions:

$$\check{u} = \frac{1}{d_x}\check{x} + u_0, \quad \check{v} = \frac{1}{d_y}\check{y} + v_0 \quad (3.10)$$

where (u_0, v_0) are the principal point coordinates, and d_x and d_y are distances between the adjacent pixels, in the respective directions.

The distortions, δ_x and δ_y , can be expressed as power series in radial distance $r = \sqrt{x^2 + y^2}$, as follows in expressions 3.11 and 3.12:

$$\delta_x = x(k_1 r^2 + k_2 r^4 + k_3 r^6 + \dots) + [p_1(r^2 + 2x^2) + 2p_2 xy](1 + p_3 r^2 + \dots), \quad (3.11)$$

$$\delta_y = y(k_1 r^2 + k_2 r^4 + k_3 r^6 + \dots) + [2p_1 xy + p_2(r^2 + 2y^2)](1 + p_3 r^2 + \dots), \quad (3.12)$$

where k_i are radial distortion coefficients and p_j decentering distortion coefficients.

Considering only the first two radial distortion terms, combining 3.9 and 3.10 and remarking that ideal pixel image coordinates are given by $u = x/d_x$ and $v = y/d_y$, a simplified relationship between $(\check{u}, \check{v})$ and (u, v) arises in expressions 3.13 and 3.14.

$$\check{u} = u + (u - u_0)[k_1(x^2 + y^2) + k_2(x^2 + y^2)^2] \quad (3.13)$$

$$\check{v} = v + (v - v_0)[k_1(x^2 + y^2) + k_2(x^2 + y^2)^2] \quad (3.14)$$

3.4 Closed-form Solution

The closed-form solution resorts to the image of the Absolute Conic to determine an analytical solution for the camera parameters, which will be used as an initial estimation of the nonlinear minimization performed on the next subsection.

In camera calibration with a 2D object, assuming that the model plane is on $Z = 0$ of the world coordinate system, a model point $\tilde{M}$ and its image $\tilde{m}$ are related by a homography H , by the following expression:

$$s\tilde{m} = H\tilde{M} \quad (3.15)$$

where H is a 3x3 matrix, defined as $H = A[r_1 \ r_2 \ t]$. Having $H = [h_1 \ h_2 \ h_3]$, comes:

$$[h_1 \ h_2 \ h_3] = \lambda A[r_1 \ r_2 \ t] \quad (3.16)$$

where λ is an arbitrary scalar. Once that r_1 and r_2 are rotations is orthonormal axes, they also are, so the two constraints of the intrinsic parameters come as:

$$h_1^T A^{-T} A^{-1} h_2 = 0 \quad (3.17)$$

$$h_1^T A^{-T} A^{-1} h_1 = h_2^T A^{-T} A^{-1} h_2 \quad (3.18)$$

The analytical solution starts with the image of the absolute conic, $A^{-T} A^{-1}$, here assigned as matrix B :

$$B = A^{-T} A^{-1} = \begin{bmatrix} \frac{1}{\alpha^2} & -\frac{\gamma}{\alpha^2 \beta} & \frac{v_0 \gamma - u_0 \beta}{\alpha^2 \beta} \\ -\frac{\gamma}{\alpha^2 \beta} & \frac{\gamma^2}{\alpha^2 \beta^2} + \frac{1}{\beta^2} & -\frac{\gamma(v_0 \gamma - u_0 \beta)}{\alpha^2 \beta^2} - \frac{v_0}{\beta^2} \\ \frac{v_0 \gamma - u_0 \beta}{\alpha^2 \beta} & -\frac{\gamma(v_0 \gamma - u_0 \beta)}{\alpha^2 \beta^2} - \frac{v_0}{\beta^2} & \frac{(v_0 \gamma - u_0 \beta)^2}{\alpha^2 \beta^2} + \frac{v_0^2}{\beta^2} + 1 \end{bmatrix} \quad (3.19)$$

where b is a 6D vector which contains the six different elements of the symmetric matrix B , therefore, b can be written as $b = [B_{11}, B_{12}, B_{22}, B_{13}, B_{23}, B_{33}]^T$.

Denoting the i^{th} column vector of H as $h_i = [h_{i1}, h_{i2}, h_{i3}]^T$ and that $v_{ij} = [h_{i1}h_{j1}, h_{i1}h_{j2} + h_{i2}h_{j1}, h_{i2}h_{j1}, h_{i3}h_{j1} + h_{i1}h_{j3}, h_{i3}h_{j2} + h_{i2}h_{j3}, h_{i3}h_{j3}]^T$, comes:

$$h_i^T B h_j = v_{ij}^T b \quad (3.20)$$

This way, the two constraints of the intrinsic parameters, written in 3.17 and 3.18, can be rewritten as two homogeneous equations, as follows:

$$\begin{bmatrix} v_{12}^T \\ (v_{11} - v_{22})^T \end{bmatrix} b = 0 \quad (3.21)$$

At last, collecting n such equations by observing n images of the model plane, it comes:

$$Vb = 0 \quad (3.22)$$

where V is a $2n \times 6$ matrix and where a unique solution is provided, up to a scalar factor, if $n \geq 3$.

After estimating vector b , the intrinsic parameters can be easily determined, using the following expressions:

$$\begin{cases} v_0 = (B_{12}B_{13} - B_{11}B_{23}) / (B_{11}B_{22} - B_{12}^2) \\ \lambda = B_{33} - [B_{13}^2 + v_0(B_{12}B_{13} - B_{11}B_{23})] / B_{11} \\ \alpha = \sqrt{\lambda / B_{11}} \\ \beta = \sqrt{\lambda B_{11} / (B_{11}B_{22} - B_{12}^2)} \\ \gamma = -B_{12}\alpha^2\beta / \lambda \\ u_0 = \gamma v_0 / \alpha - B_{13}\alpha^2 / \lambda \end{cases} \quad (3.23)$$

As B is fully determined then matrix A , according to 3.19, may also be. Therefore, the extrinsic parameters for each image can be calculated according to 3.24:

$$\begin{cases} r_1 = \lambda A^{-1} h_1 \\ r_2 = \lambda A^{-1} h_2 \\ r_3 = r_1 \times r_2 \\ t = \lambda A^{-1} h_3 \end{cases} \quad (3.24)$$

where $\lambda = 1 / \|A^{-1} h_1\| = 1 / \|A^{-1} h_2\|$.

3.5 Least-squares Minimization

The closed-form solution is a minimization of an algebraic distance, which is not physically meaningful. On the other hand, assuming that the image points are corrupted by independent identically distributed noise, we can minimize the functional on 3.25:

$$\sum_n \sum_m^{j=1} \|m_{ij} - \hat{m}(A, R_i, t_i, M_j)\|^2 \quad (3.25)$$

where $\hat{m}(A, R_i, t_i, M_j)$ is the projection of point M_j in image i , according to 3.15, and m is the number points on the model plane.

In order to consider lens distortion in the calibration process, it is necessary to estimate k_1 and k_2 after all the other parameters are estimated. 3.13 and 3.14 give two equations for each point in each image. Having k_1 and k_2 , the estimation of the other parameters can be performed by solving 3.25, with 3.13 and 3.14 replacing $\hat{m}(A, R_i, t_i, M_j)$.

To improve convergence, 3.25 can be extended to the following functional:

$$\sum_n \sum_m^{j=1} \|m_{ij} - \hat{m}(A, k_1, k_2, R_i, t_i, M_j)\|^2 \quad (3.26)$$

where $\hat{m}(A, k_1, k_2, R_i, t_i, M_j)$ is the projection of point M_j in image i , according to 3.15. This is a nonlinear problem solved with the Levenberg-Marquardt Algorithm, where an initial estimation of A and $R_i, t_i | i = 1, \dots, n$ can be done through the closed-form solution or through 3.25. Moreover, an initial guess of k_1 and k_2 can be obtained through the replacement of 3.13 and 3.14 in 3.25, or by setting them to zero.

Chapter 4

Contour Measuring System

4.1 Resolution

In order to determine the components of a measuring system, it is important to know in advance the resolution necessary to obtain valid results. A lower resolution implies a less complex and, therefore, cheaper system, and thus a parametric study was performed, in which all parameters were kept constant throughout the simulations, except the resolution of the CMM data. First, the CMM measurements of welded plate specimens, used in Braga (2012) were used with 7 decimal digits, whose resulting stresses served as benchmark to compare to the other results, with that same specimen. The iterative process performed is presented in figure 4.1 .

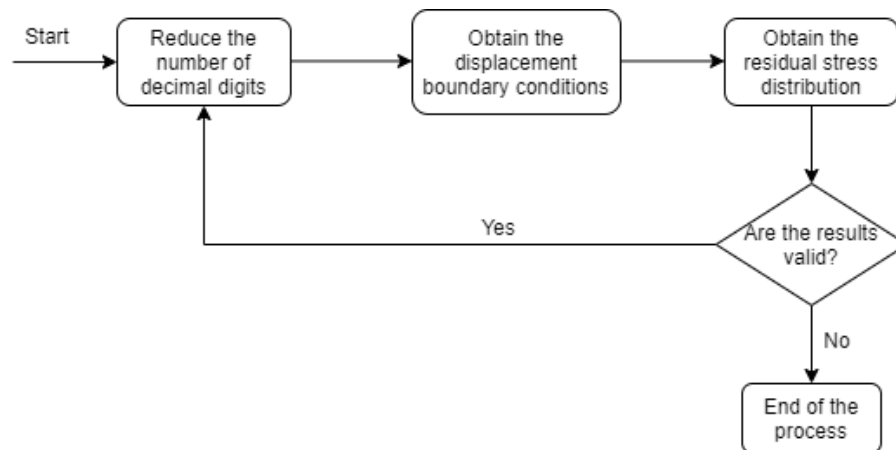


Figure 4.1: Flowchart describing the resolution *a priori* study.

After reducing one decimal digit to the CMM measurements, the corresponding MATLAB algorithm is run and generates the displacement boundary conditions for the cut surface. These are then included in the ABAQUS input file, where the specimen was previously modelled, and from which the stresses for the surface nodes are obtained. Finally, the obtained stresses are compared with those obtained with 7 decimal digits and their validity is checked, since the simulations continue until the results cease to be valid.

Initially, it was used the relative error as a comparison parameter between the results obtained through the 7 decimal digits and the others, however, it often led to the analysis of many nodes with low absolute stress values, comparing to yield strength of the material. Therefore, it was used the absolute error divided by the material's yield strength instead, which indicates validity until a maximum reference value of 10%, according to Braga (2012). This enabled the detection of only the nodes whose error could interfere with the evaluation of the specimen's elastic behaviour.

The Z direction and the XY plane were tested independently and the results are presented next. While testing the Z direction, the decimal digit reduction of the CMM data was only along this axis, keeping the other directions with 7 decimal digits. Testing the XY plane, the exact opposite occurred.

4.1.1 Welded Plate Specimen

The study of the system's resolution was first performed using the CMM data of a specimen resulting from two welded S355NL steel plates. This specimen has already been used in a previous work, Braga (2012), therefore it was available at the beginning of this project. It was cut to half of its length by wire EDM (wEDM), resulting in two 380 x 300 x 6 mm halves. Linear finite element analysis (FEA) was performed with Abaqus software package, using a mesh of C3D8R elements on the cut surface, with 2 x 0.5 x 1 mm spacing, as shown in figure 4.2, and taking into account 400 MPa of yield strength for the base material and 420 MPa for the filler material, (Braga, 2012). The algorithm used to calculate the displacement boundary conditions is attached to Appendix B.



Figure 4.2: Overall view of the Welded Plate ABAQUS model.

Tables 4.1 and 4.2 summarize, respectively, the results obtained along the Z axis and the XY plane, showing the number of critical nodes for each simulation performed and detecting the greatest error associated to a node.

As seen in table 4.1, simulations were only performed until the results ceased to be valid, which happened with the 2 decimal digits data. The latter case, being the only one to show error values above 10%, was discarded, so a measurement system providing displacements with only

Table 4.1: Error determination for the decimal digit reduction along the Z axis of the Welded Plate specimen.

No of decimal digits	Amount of nodes w/ an error >10%	Maximum error found [%]
6	none	0.0175
5	none	0.1020
4	none	0.1920
3	none	2.2240
2	57	15.8893

2 decimal digits along the Z axis is not sufficient to obtain valid stress values. Therefore, the hypothesis of having displacements with 1 decimal digit was also discarded.

With a maximum error of 2,224%, below 10%, measuring data with 3 decimal digits is a feasible solution for the problem, so if one uses a contour measuring system with a resolution of 0,001 mm along the Z axis, the results will not show significant errors. To corroborate the previous conclusion, let one look at figure 4.3, which compares the residual stress distribution obtained by measuring data with 7, 3 and 2 decimal digits. Due to the fact that contour method provides unreliable results near specimen's edges, for reasons explained in chapter 2, the presented stresses were measured at half thickness of the specimen, approximately. It is noticeable that the blue and red curves are practically coincident along the width of the specimen, contrarily to the yellow curve, that shows several regions of deviation from the other two.

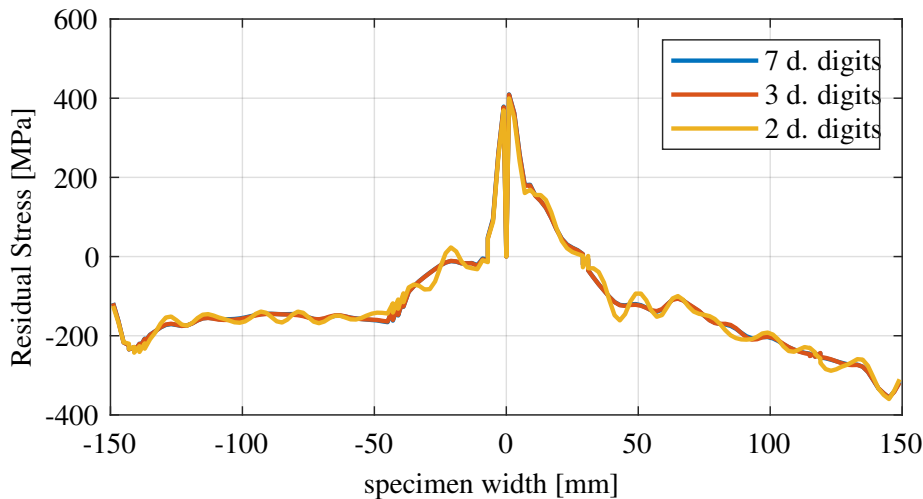


Figure 4.3: Residual stress distribution at mid thickness for the Z axis study of the Welded Plate specimen.

Performing the same evaluation to the XY plane and looking at table 4.2, it is noticeable that the obtained results are also very similar, since they cease to be valid with 2 decimal digits and the maximum error found with 3 decimal digits, 3,516%, is just slightly greater than 2,224%.

Therefore, the contour measuring system must provide a minimum resolution of 0,001 mm along the Z axis but also along the X and Y axes.

Table 4.2: Error determination for the decimal digit reduction along the XY plane of the Welded Plate specimen.

No of decimal digits	Amount of nodes w/ an error >10%	Maximum error found [%]
6	none	0.00
5	none	0.00125
4	none	1.21
3	none	3.516
2	56	21.324

Similarly to figure 4.3, figure 4.4 presents the stress values at half thickness, along the width of the specimen, corroborating the analysis presented in table, in this case, table 4.2. The courses of these three curves confirm the deviation between results relative to 2 decimal digits and results obtained with 3 and 7. It would be expected the data within $-7 \text{ mm} < X < 7 \text{ mm}$, considering the center of the cut surface the origin of the referential, to present greater deviations, since it is the zone of the weld bead, but both studies, in the Z axis and in the XY plane, showed the opposite.

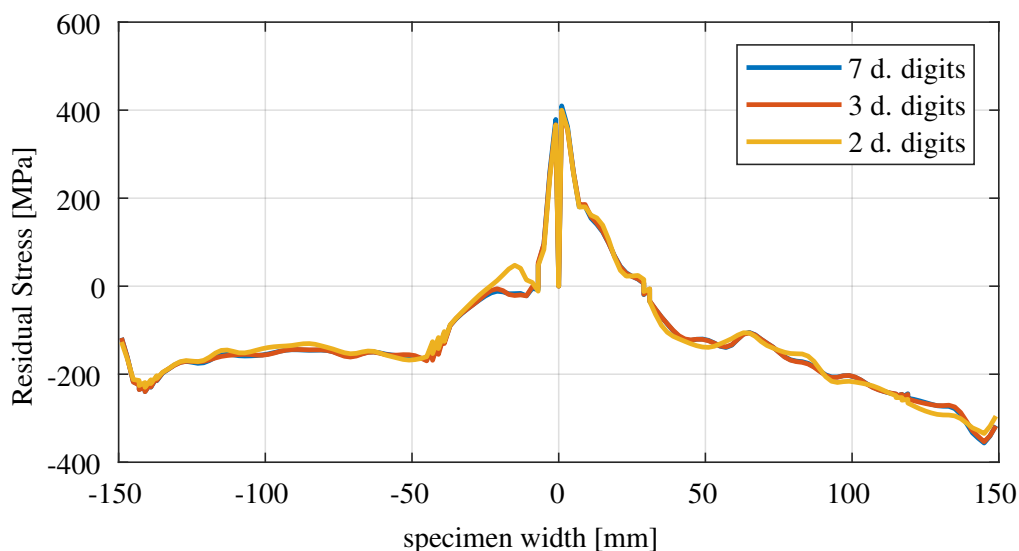


Figure 4.4: Residual stress distribution at mid thickness for the XY plane study of the Welded Plate specimen.

It should be noted that the CMM measurements, in this specimen, were performed in such a way that each acquired point coincided with a mesh node of the model, which eased this process. The same did not happened with the aluminum specimen, as will be seen further.

4.1.2 Aluminum specimen

For validation purposes, the previous study was also developed for an aluminum specimen. The present specimen was also cut to half of its length by wEDM, resulting in two plates, A and B, with equal dimensions, 300 x 143 x 6 mm. The ABAQUS model used a more refined mesh of C3D8R elements this time, with a spacing of 0.5 x 0.5 x 0.5 mm.

As seen in figure 4.5, this aluminum specimen shows an asymmetry relative to the Y axis, due to the smaller protuberance present in the central zone of the specimen and also due to the inclination of the right plate. In addition, the CMM measurements of the present specimen, shown in figure 4.7, were obtained without using a specific pattern and disregarding the mesh of the model, reinforcing the need of adapting the MATLAB algorithm presented in Appendix B.

Regarding the reasons mentioned above, it was necessary not to use all the CMM points obtained, which made the analysis of the resolution in the XY plane not feasible with this specimen, a conclusion that will be explained further.

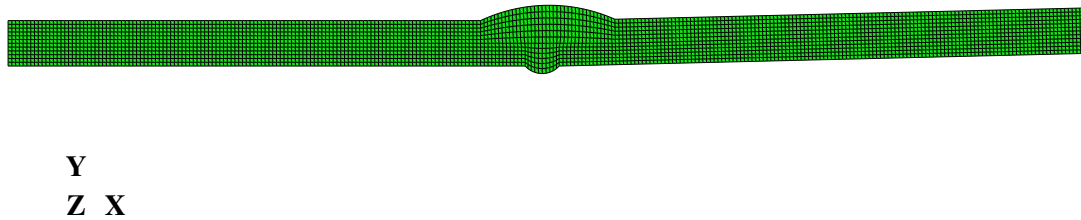


Figure 4.5: Cross-section view of the Aluminum specimen ABAQUS model.

As this is a validation study only, no simulations were performed with 6, 5 or 4 decimal digits, skipping the process directly to simulations with 3 and 2. Thus, table 4.3 presents the corresponding results, which corroborate the conclusions drawn with the previous specimen, for the Z axis. It should be noted that the author of this work does not know the composition of the aluminum alloy that constitutes this specimen, and thus its yield stress is unknown. Regarding safety and considering that the less resistant aluminium alloys have a yield strength between 90 and 110 MPa, the errors were calculated considering $\sigma_y = 90$ MPa.

Table 4.3: Error determination for the decimal digit reduction along the Z axis of the Aluminum specimen.

No of decimal digits	Amount of nodes w/ an error >10%	Maximum error found [%]
3	none	9.2089
2	924	91.3670

The greatest error for the case regarding the 3 decimal digits, in table 4.3, is very close to 10%, however, it must be taken into account that the yield strength used belongs to the aluminum alloys used in soda cans and other applications of the same kind, so the errors calculated with the actual yield strength would certainly be lower. Therefore, it can be corroborated that the resolution of the measurement system should present at least 0.001 mm along Z direction. Figure 4.6 shows the previously mentioned deviations between the curve corresponding to the 2 decimal digits and the other two curves.

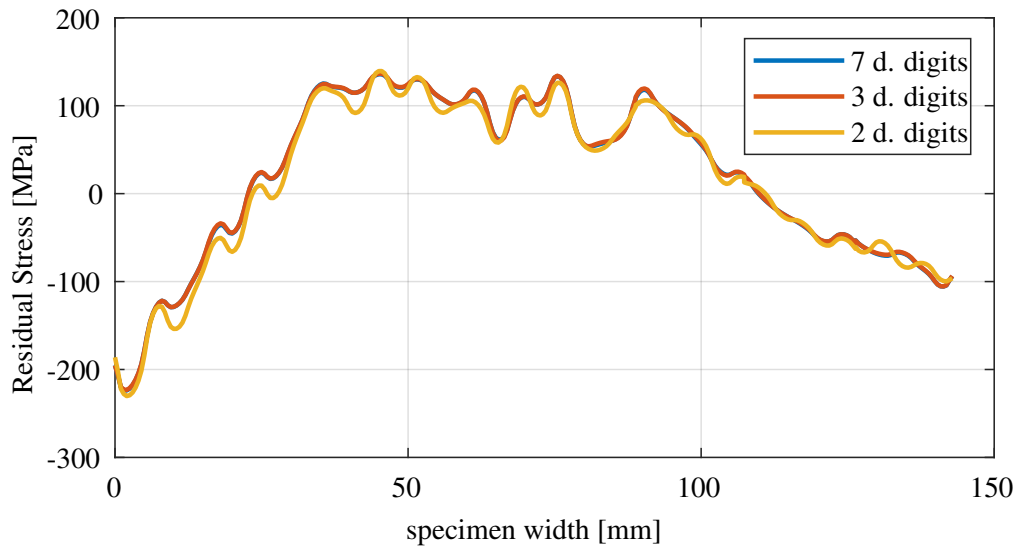


Figure 4.6: Residual stress distribution at mid thickness for the Z axis study of the Aluminum specimen.

As previously mentioned, the MATLAB algorithm organizes the points obtained by CMM in matrices. Taking as an example the y-coordinate of each point, we have the matrix '*measured_A_matrix_y*', where each column corresponds to the y-coordinates of the CMM points found in a given x-range, and such range is set according to the x-spacing of the CMM measurements. As in this case there is no fixed spacing in the measurements, as shown in figure 4.7, it was necessary to discard some of the points. Furthermore, it should be noted that in an attempt to perform the resolution study in the XY plane for this specimen, there were CMM points that easily switched their positions in the aforementioned matrix columns, due to the decimal digit reduction, leading to debugging errors. Thus, for this study to be possible, using this MATLAB algorithm, it would be necessary to repeat the CMM measurements, maintaining a constant spacing along the X axis.

As a concluding remark, the system's resolution must be at least 0.001 mm along both Z axis and XY plane.

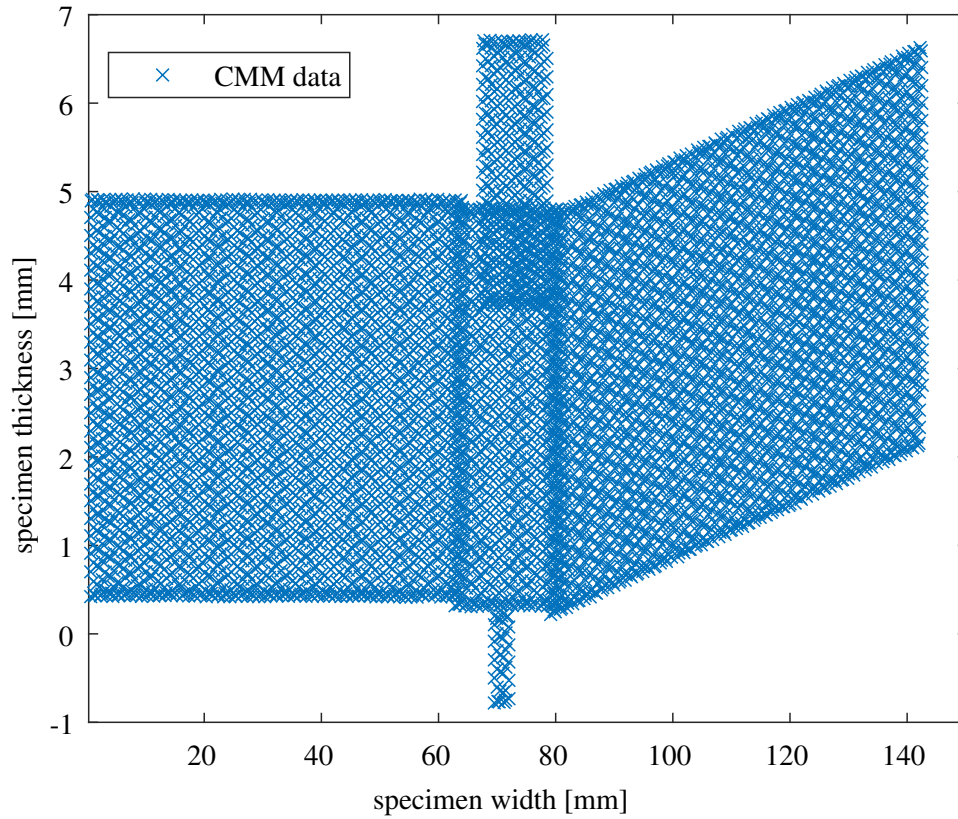


Figure 4.7: CMM data distribution over the Aluminum specimen cut surface.

4.2 System Parts

The present section describes and justifies the choice of each system component used in this work, starting with a brief explanation about laser triangulation, followed by a detailed view of each component.

A 2D scan requires the use of a camera, however, for the present work it would not be enough, since a 3D acquisition is required, which will be conferred by the intersection between each world point and the laser light plane. Thus, a laser line projector is required to determine the z coordinate of the image points that will take part on the profile construction, i.e., the in-depth positions, parallel to the optical axis. In order to construct the whole surface profile, the specimen must be moved relative to the camera+laser subsystem, therefore, it is used a linear positioning device for that purpose, (MVTec).

Note that it was used a LED line projector instead of a laser, the reasons will be explained further.

4.2.1 Camera

For the image acquisition, it was used an UI337 uEye camera, figure 4.8. Its USB 3.0 interface guarantees a high data rate of 420 MByte/s and low CPU utilization, as well as its compact format (29 x 29 x 29 mm) improves the space optimization on the assembling table. Moreover it offers a 4.19 megapixel (MP) resolution and 5.5 μm large pixels. (iDS, 2020)

Table 4.4 summarizes the main specifications of the uEye camera, some of which will be used as input to determine the assembly lens+focussing unit needed.



Figure 4.8: UI337 uEye iDS Camera

Table 4.4: uEye Camera main specifications

Camera Specifications	
Interface	USB 3.0
Frame Rate	30 fps
Resolution	2048 x 2048
Sensor Size	1"
Pixel Size	5.5 μm

The determination of the suitable lens configuration was performed resorting to a machine vision software, MachVis, developed by Excelitas company. MachVis needs five input parameters to give out the output: the free working distance (FWD), that consists of the distance between the lens and the analyzed surface, as shown in figure 4.9, the analyzed object height, the sensor size, the camera adapter type and the pixel size, (Qioptiq, 2020). All the input data required is summed up in table 4.5. The FWD was set to 250 mm because shorter distances would decrease the depth of field (DOF) from 0.58 mm to 0.40 mm, which would complicate the calibration process, as explained in the calibration section. On the other hand, FWD enlargement could increase the DOF up to 0.65 mm, but would also demand a greater gap between camera and LED projector when implementing large triangulation angles.

Note that 32 mm, set as the object height, is referring to the thickest specimen analyzed, the CT specimen, which has that precise thickness. Moreover, the pixel size inserted was $5.0\text{ }\mu\text{m}$ instead of $5.5\text{ }\mu\text{m}$, because MachVis allows integer values only, for this parameter.

Table 4.5: MachVis input parameters

MachVis Input	
FWD	250 mm
Object Height	32 mm
Sensor Size	1"
Camera Adapter	C-Mount
Pixel Size	$5.0\text{ }\mu\text{m}$

As output, MachVis generates a list of suitable lenses from LINOS lens catalogue, where are also displayed all the calculated parameters for each choice, (Qioptiq, 2020). Then, the operator is meant to select the most suitable lens according to its price/availability in the laboratory and, for this specific purpose, the offered DOF. After the selection, the software prints out a scheme with the chosen lens as well as all the calculated parameters, figure 4.9, and prints out another scheme showing the assembly sequence of the lens, focussing unit and all extensions and/or adapters required, figure 4.10.

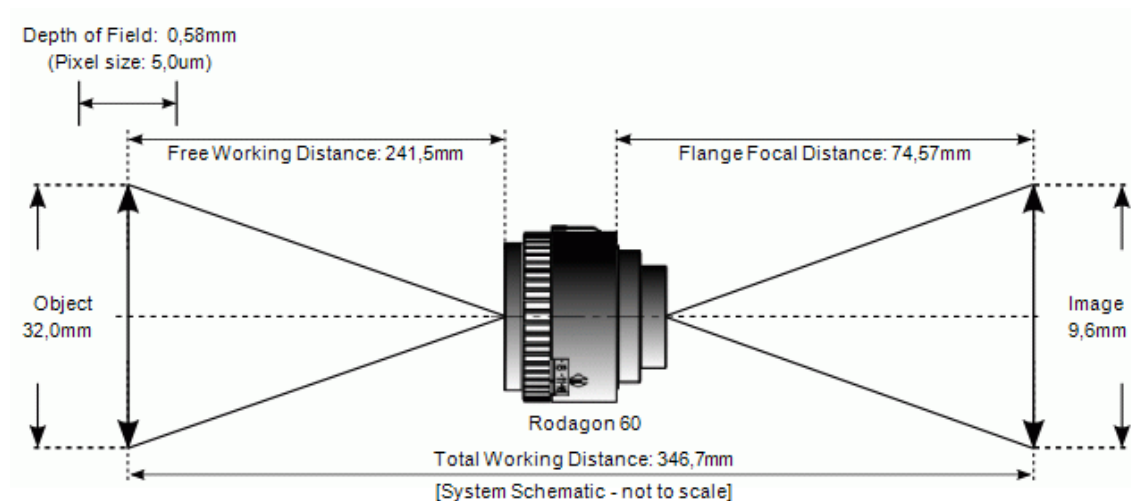


Figure 4.9: Scheme of the lens and the calculated parameters

Rodagon 60 was the chosen lens and so, DOF and FWD were calculated accordingly, as this setup offers a **DOF of $0.58\text{ }\mu\text{m}$** - distance between the nearest and the farthest points of an image that are in sharp focus conditions - and an actual **FWD of 241.5 mm** , figure 4.9. As final step of this section, the set of components shown in figure 4.10 is assembled to the uEye camera.

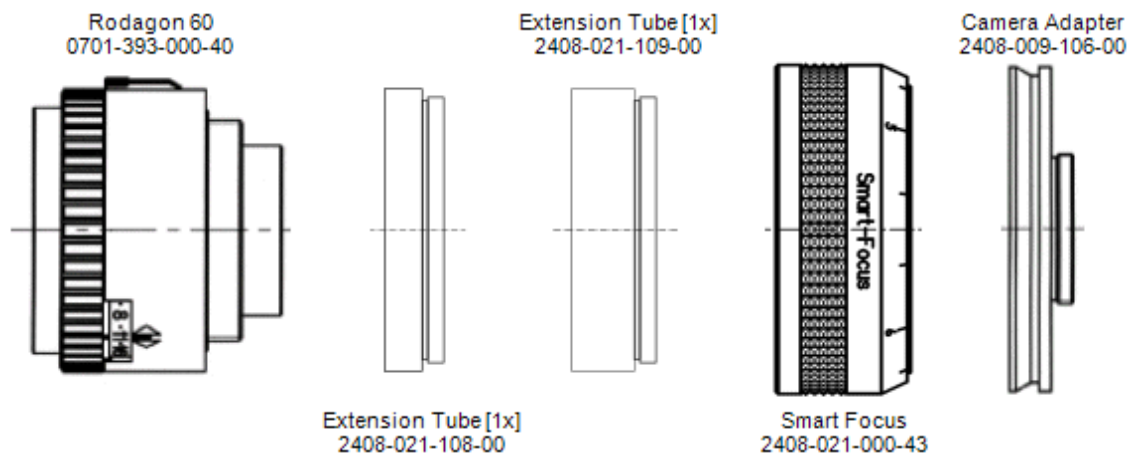


Figure 4.10: Scheme of the lens assembly

4.2.2 LED Line Projector

Initially, this measuring system was built using a default laser line beam to perform the first calibration attempts, however, it was later replaced by a LED line projector which provides a sharper and thinner line, the **EFFI-Lase V2**, in figure 4.11. It includes an equipped **V7531-MPZ**, a compact and light weight lens that offers 75 mm of focal length and a maximum aperture ratio of 1:3.1.



Figure 4.11: EFFI-Lase V2 LED Projector

Effi-Lase V2 has many available options within the optical field, as the user may choose the pattern to be projected by selecting the corresponding LED card. One may also choose the wavelength of the beam, and the 3D profilometry, by selecting the corresponding mask, Efflux (2020). For the application of the present work, it was selected the LX1 green LED card, which corresponds to a line projection of 525 nm wavelength. Having the sub pixel resolution in mind, it was chosen the L01 mask, that enables a 13 mm LED line, with a thickness of 50 μm .

Table 4.6: Optical specifications of Effi-Lase V2

LED card	LX1
Wavelength	525 nm
Mask	L01

Figure 4.12 shows both laser line, on the left, and LED line on the right, projected on the same CT specimen surface. The sharpness they show is clearly different from each other, as the laser line shows a creased noise around it, and mostly near its right edge, where one can notice a secondary tiny line, which is detrimental to calibration and to the 3D point cloud acquisition itself. Besides being sharper, the LED line, facilitates the use of sub-pixel resolution by forming the 13 mm line, also highlighted in figure 4.12. To obtain this thinner line, it is necessary to regulate the MPZ lens aperture and focal unit until it is visible on the specimen's surface. Note that the first specimen observed in chapter 5 will not be subjected to this thinner line, since its thickness is 32 mm, which exceeds the maximum length of this line, 13 mm.

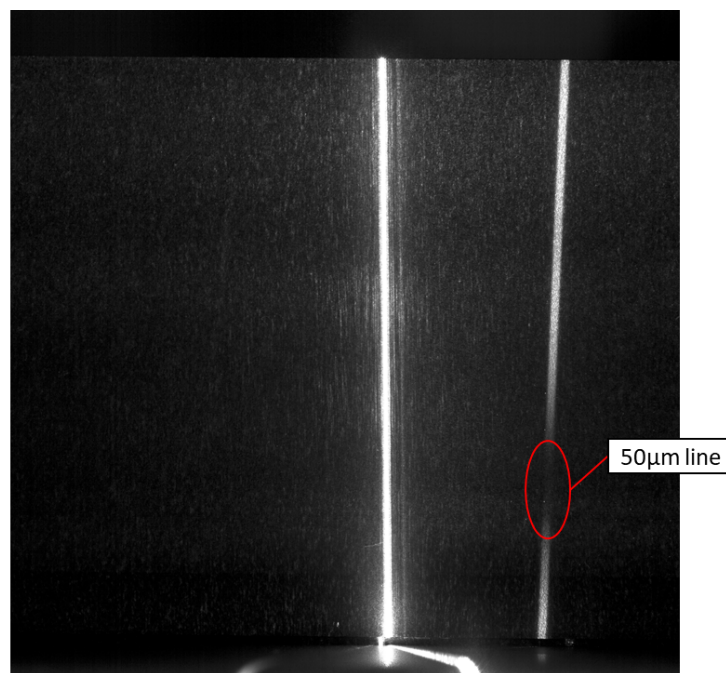


Figure 4.12: Laser line (on the left) vs LED line (on the right)

4.2.3 Linear Positioning System

The movement of the studied specimens was performed using Micro-Controle UT100 translation stages, which are motorized by stepper motors. Figure 4.13 shows one of the used translation stages, with a 0-20 mm displacement range, which has a resolution of 0.0001 mm, as shown on the control unit display, figure 4.14.

Figure 4.13 highlights two specific parts of the linear table: the cable connector, used to connect it to the control unit, and the moving plate, which is the part that actually transmits motion to the specimens. Note that the moving plate is positioned slightly above the rest of the upper surface of the linear table - approximately 0.2 mm, in order to avoid friction between the specimen and the static surface. If the displacement range offered by one single linear table is not enough to scan the whole specimen's length, then it is necessary to stack the ranges of two of them.

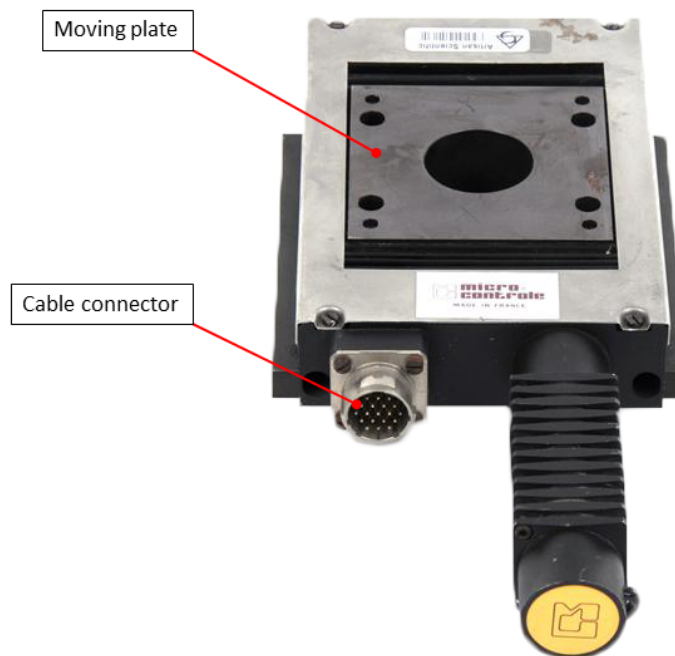


Figure 4.13: UT100 Linear Table

The linear tables are controlled by a Micro-Control TL78 control unit supplied by 220V AC power source, figure 4.14. A single unit can control up to three linear tables simultaneously, as each display and set of buttons correspond to each one of the controlled devices. This device enables the operator to set the movement's way by pressing either left or right displacement button, and also the movement speed, by pressing the high-speed button, low-speed button or none of them. All the experiments of the present work were performed using the high-speed option, whose speed value is going to be determined further, in the speed calibration section.

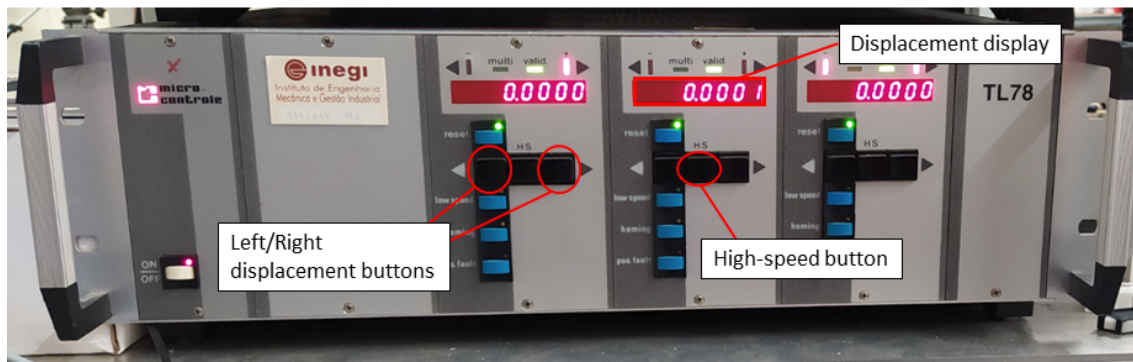


Figure 4.14: TL78 Control Unit

4.3 System Configuration

It is known that the analyzed object has to move relative to the camera+LED projector subsystem, but the configuration that both should adopt relative to the object remains to be determined, therefore, one of the three possible configurations listed below is going to be chosen. Each one of the a), b) and c) cases from the list below is shown in figure 4.15, with the corresponding letter:

- (a) Camera orthogonal to the surface and tilted LED plane;
- (b) Tilted camera and orthogonal LED plane;
- (c) Both camera and LED plane tilted;

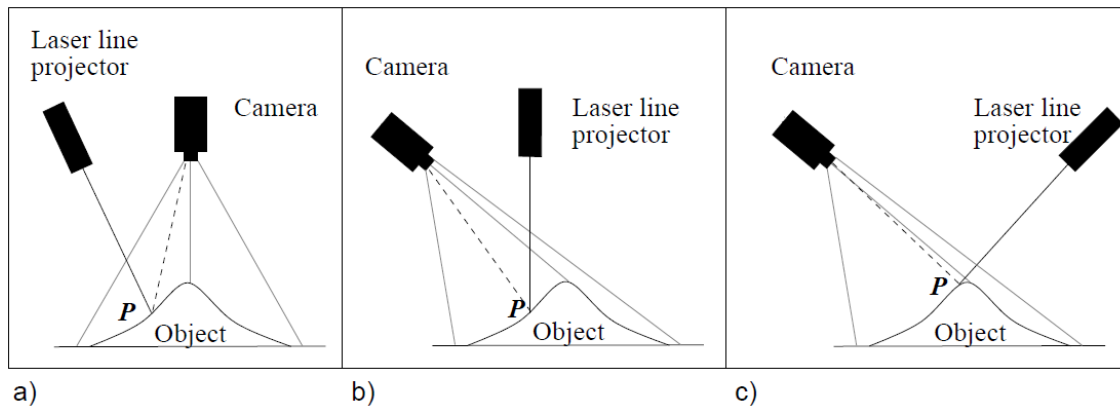


Figure 4.15: Possible system's configurations. (MVTec)

The first calibration experiments were still performed with the laser projector and were also performed using configuration b), which turned out to be detrimental to the quality of the captured images. Recalling to the previous section, the DOF achieved with the chosen camera+lens setup is 0.58 mm, which is easily exceeded if the camera is tilted relative to the surface/calibration target, causing the acquired images to go blurry. Therefore, configuration c) would cause the same problem and for that reason **configuration a)** was adopted.

In the chosen configuration, a), the position the LED projector relative to the camera must be such that avoids shadowing effects and occlusions. Shadowing effects occur when one part of the surface lies between the LED projection and other surface area, causing it to be acquired by the camera but not being illuminated by the LED line. Occlusions occur if an object point is illuminated by the laser line but is not visible in the image. As the scanned surfaces show displacement variations in the order of hundredths of a millimeter, these phenomena are very unlikely to occur. Moreover, the triangulation angle should be between 30° and 60° and the only restriction to its value would be the presence of shadowing effects and/or occlusions, (MVTec).

After the aforementioned considerations, each system component was arranged as shown in figure 4.16, where the camera is connected to a laptop through a USB 3.0 cable. It should be noted that the presented configuration is the one used for the CT specimen analysis, which requires the use of two linear tables coupled to each other, and so forces the lifting of both camera and LED projector, through the metal parts also shown in figure 4.16.

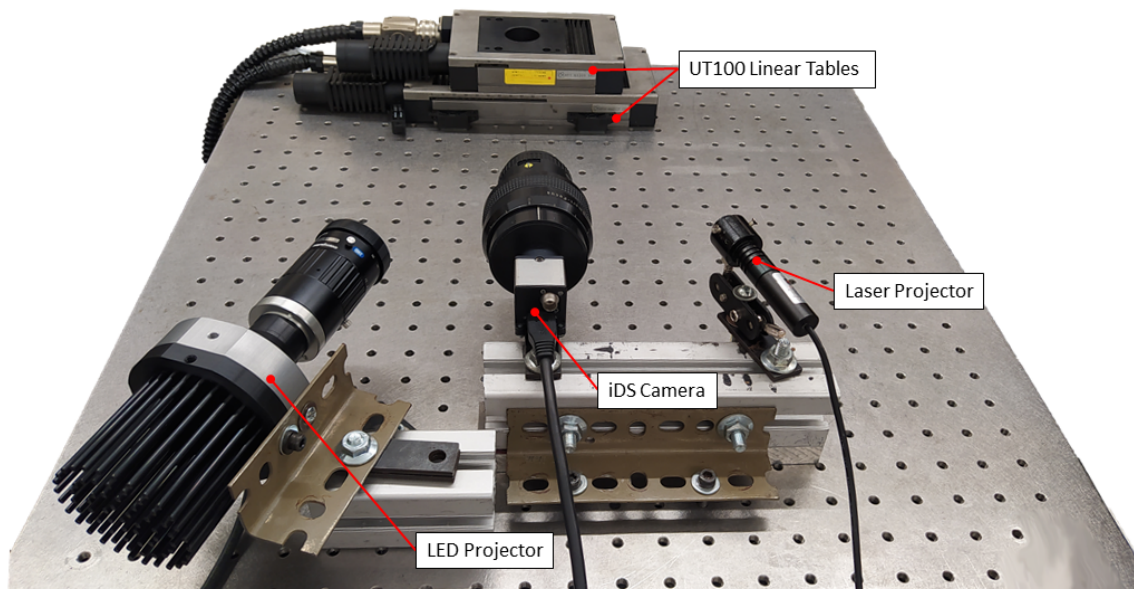


Figure 4.16: System configuration

4.4 System Calibration

Before acquiring the 3D point clouds of the measured surfaces, it is necessary to calibrate the measuring system in order to check if the extrinsic parameters of the system are correct and also to determine many other parameters, important to build the point cloud. The system calibration involves the following three types of calibration:

- Camera calibration;
- LED Projector calibration;

- Speed calibration;

The system calibration was performed resorting to the MATLAB algorithm attached to Appendix A, developed at the Optics and Experimental Mechanics Laboratory (LOME). Camera calibration must be the first calibration performed as its main goal is to obtain camera's intrinsic and extrinsic parameters. Then, the calibration order is arbitrary: the LED projector calibration aims to determine the equation of its luminous plane, and the system speed calibration aims to calculate the specimen's speed along the 3 different spatial directions. The three calibration stages are going to be studied in the next subsections, however, as the calibration process is strictly related to the 3D point clouds acquired, its numerical results are going to be shown in chapter 5 only.

Note that after calibrating the system, none of the components may be moved nor the lens focus be changed, otherwise a re-calibration has to be performed. On the other hand, actions such as changing lens aperture or varying the LED line thickness do not require re-calibration.

4.4.1 Camera Calibration

The camera calibration was performed using the Zhang's method (Zhang, 2000), which enables the use of any planar pattern, as long as the metric on the plane is known. Thus, knowing that MATLAB has an appropriate function for it, a checkered pattern was used to calibrate the system. There are several generators of this type of pattern, such as Calib.io (2020), to cite one, where one just have to enter the number of columns, number of rows and the size of each square, being returned a .pdf file with the real size pattern. Then, the pattern needs to be printed and fixed to a rigid and flat surface. Thereby, an **8x9 pattern** was printed, with **2.5 mm square size** and attached with double-faced duct tape to an aluminum 150x150x2 mm plate. While performing the first fixing attempts it was used glue instead of duct tape, but it ended up leaving stains and some reliefs on the checkerboard, which are detrimental artifacts to the calibration process. It is important to remark that one side of the checkerboard must have an even number of squares and the other side must have an odd number of squares, so that the board contains two black corners along one side and two white corners along the opposite side, which enables the detection of the pattern orientation (MathWorks, 2020a).

To start the camera calibration it is necessary to position the checkerboard on the linear table, approximately 241.5 mm (FWD) far from the lens, and to turn on the LED projector, as the acquired image set will also be used in its calibration. Before starting acquiring images, the operator has to ensure good focal conditions by slightly rotating the smart focus and by observing the corresponding change on the image display, and for this purpose a ruler was used as help, figure 4.17.

When the ruler numbers and marks are displayed with maximum sharpness, one can proceed to the image acquisition. Afterwards, a set of images, whose amount may vary, is acquired using the image acquisition tool of MATLAB or similar. A high number of images may result in error accumulation, but a set with few images may not contain 3 images in good focal conditions.

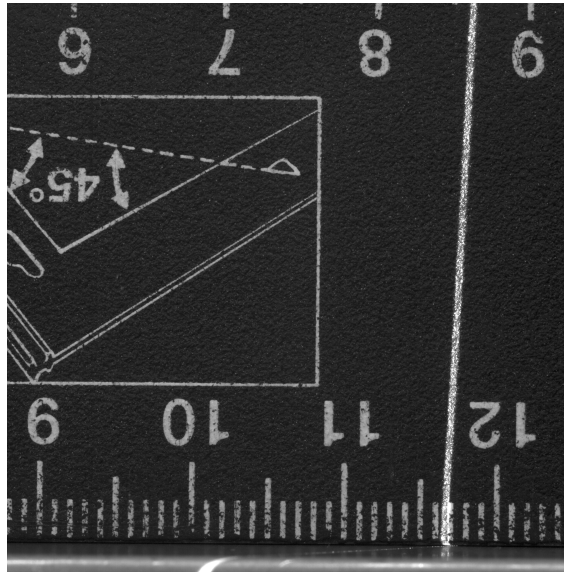


Figure 4.17: Image of ruler numbers under sharp focal conditions.

Therefore, it is recommended the use of **20 to 26 images** of the kind shown in figure 4.18, (Yu et al., 2018).

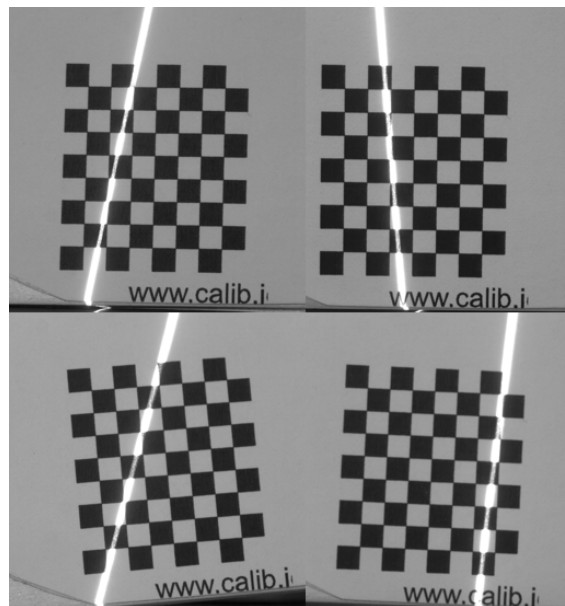


Figure 4.18: Sample of 4 images used to perform camera calibration.

The images acquired for camera calibration require, not only sharp LED lines, but also sharp pattern shapes, since the quality of calibration will depend on the accuracy with which the intersection points between white and black squares of the checkerboard are located, so it is important that the acquisition parameters such as brightness, gain and exposure are adjusted according to the intended purpose. It is recommended to acquire images in which the checkerboard shows in different planes, either by rotating it around a vertical or horizontal axis, or by translating it over

the optical axis, or both.

Theoretically, the 0.58 mm DOF will make the calibration process more difficult since the image points that exceed this value will blur, so the image acquisition process will depend in part on the operator's experience. Note that different sets of images, even using the same calibration pattern and maintaining all other parameters, might lead to significantly different results, so it is recommended to make several calibration attempts before the final acquisition.

The calibration algorithm has as input the number of lines and columns, the square size and the captured images. Then, it uses the *detectCheckerboardPoints* function, which stores the image coordinates of the intersection points detected in the captured images, and then the function *generateCheckerboardPoints* will generate an N-by-2 matrix considering the introduced board size and square size, having N points of intersection. Finally, the *estimateCameraParameters* function, taking as input the two previous functions outputs, calculates the intrinsic and extrinsic camera parameters. Note that the three aforementioned functions are existing functions from the MATLAB library.

Figure 4.19 contains an image taken from a calibration set, in which the detected intersection points are green marked, and in red, the re-projection of each one of these points, resorting to the extrinsic parameters.

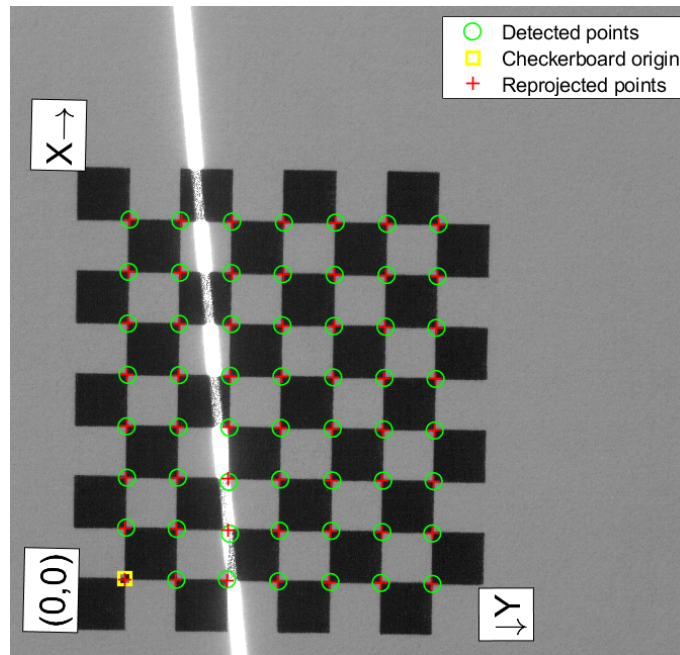


Figure 4.19: Re-projection of the checkerboard intersection points

4.4.2 LED Plane Calibration

Calibrating LED projector consists of determining the equation of its luminous plane, and this does not require new image acquisition because the algorithm sets the variables that are necessary

throughout the whole system calibration as global variables. Therefore, this step of system calibration only requires selecting the option "Calibrate Laser", available on the displayed window, after running the algorithm.

The 3D coordinates of the checkerboard points, illuminated by the LED line, can be determined through their image coordinates, obtained by applying a threshold value, and the extrinsic and intrinsic parameters, $[R_i t_i]$ and A , previously calculated. Equation 4.1 defines the equation of the plane containing the checkerboard in each image:

$$n^T [x \ y \ z]^T - n^T t_i = 0 \quad (4.1)$$

where $n = R_i[0 \ 0 \ 1]$ is the plane normal vector. The homogeneous coordinates of the LED points are given by:

$$x_h = \frac{1}{z} [x \ y \ z]^T = A^{-1} x_{laser} \quad (4.2)$$

Substituting 4.2 in 4.1, we obtain:

$$n^T x_h z - n^T t_i = 0 \quad (4.3)$$

Thus, the depth coordinate, z , is determined through the following expression:

$$z = \frac{n^T t_i}{n^T x_h} \quad (4.4)$$

which leads to:

$$[x \ y \ z]^T = \frac{n^T t_i}{n^T x_h} x_h \quad (4.5)$$

Having the 3D coordinates calculated, the A to D coefficients are determined and replaced into equation 4.6, resulting in the equation of the LED projector luminous plane.

$$Ax + By + Cz + D = 0 \quad (4.6)$$

Once that camera and LED projector calibration is complete, a diagram is shown on the display, in which the system extrinsic parameters are visible, figure 4.20. This visualization may be used by the operator to evaluate the calibration accuracy, either by the distance between the camera and the targets or by the baseline distance, distance between the camera and the LED projector. Note that the observed distance between the camera and the targets should be greater than the FWD, since the optical center is located within the optical lens (Amir and Thörnberg, 2017).

4.4.3 Speed Calibration

The purpose of speed calibration is to obtain the three speed values along each axis, X, Y and Z, wherein the ideal result is to obtain the real value of travel speed along the X direction and

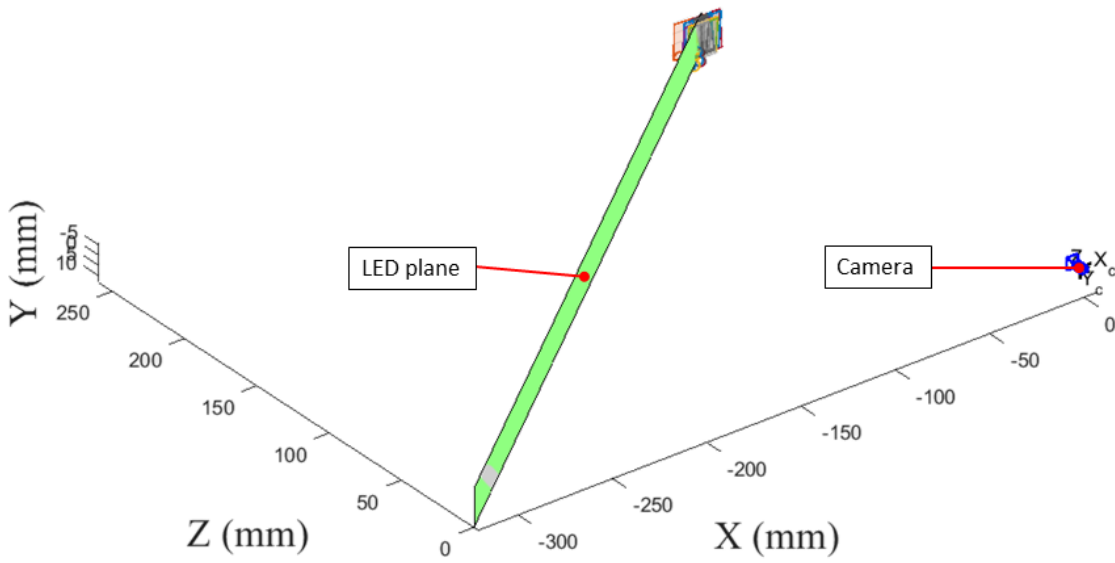


Figure 4.20: System extrinsic parameters visualization.

to obtain null values along the other two directions, since these are not desired. This step has as input the number of columns and rows of the checker pattern and its square size, similarly to the previous calibration steps, but also needs the acquisition frame rate, which was set to 7 frames per second (fps). Although the uEye camera is able to acquire images at a maximum of 30 fps, it was connected to a CPU without a USB3.0 interface, which decreased this capacity to about 7/8 fps, so it was used the iDS software, uEye Cockpit, for the acquisition, which enables the set of a constant frame rate.

The practical procedure consists of attaching the checkerboard to the moving plate of the linear table, so that its face is set perpendicular to the optical axis. Then, the operator must turn off the LED projector and start the movement of the platform, using the high speed option, and start the image acquisition. Note that all captured images must have all their board points visible, as shown in figure 4.21, where one can see a sample of images taken from a captured sequence at this calibration stage.

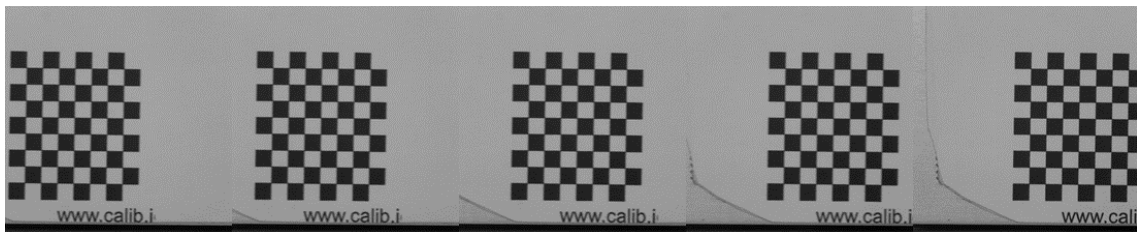


Figure 4.21: Sample of images captured during the speed calibration stage.

Ideally, the image coordinates of the board points would be determined using the intrinsic matrix calculated in the camera calibration stage, but this would imply knowing already the coordinates of these same points along the depth direction, Z . Instead, the camera extrinsic parameters are estimated again, using an array with all the images collected at this stage plus some images

used in the camera calibration stage. The algorithm was prepared to use 3 camera calibration images only, but it was adapted to use all of them, in order to increase speed calibration accuracy. Thus, the determined rotation and translation matrices are used to correct the board points of the images acquired in this calibration stage. The problem with this methodology is that the images acquired in this stage have no variation in the checkerboard plane, which is detrimental for the calibration accuracy, and consequently for the calculated parameters. The reason why the algorithm was set to add all the camera calibration images at the present stage's is that the more images added, the smaller the detrimental effect.

Once the speed calibration is complete, the display shows the speed values along each direction and three graphs like those in figure 4.22, where the displacements are represented over time.

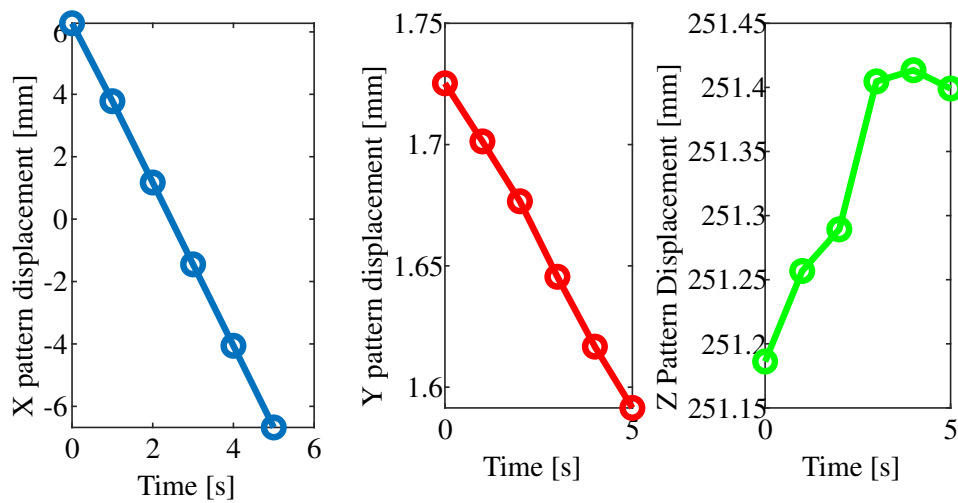


Figure 4.22: System displacements along X,Y and Z directions.

4.4.4 3D Point Clouds

This section describes how to obtain the 3D point clouds, starting by describing the necessary images. Contrarily to the camera calibration images, images giving rise to the 3D point clouds should be dark and should show the LED line well defined, as only the illuminated points matter, so both the image acquisition parameters and the lens aperture should be adjusted accordingly. Therefore, it is intended that the images acquired at this stage look like the one on figure 4.23, in terms of sharpness and noise absence.

The MATLAB program has the option of viewing all the pixels intensity (0-255) of an image picked from the acquired image set, which is advantageous to check whether there is saturation of the LED line or not, and to set the threshold value to be used as input. Figure 4.24A shows the spectrum of a saturated LED line and the noise around it, so the aim is to adjust the image acquisition parameters and to apply a threshold value so that the LED line is not saturated and the noise is not included, thus, giving rise to images similar to the one in figure 4.24B.

There are two different ways to obtain the illuminated points: by previously selecting the option "Consider only the right edge" or the option "Consider only the maximum", where the

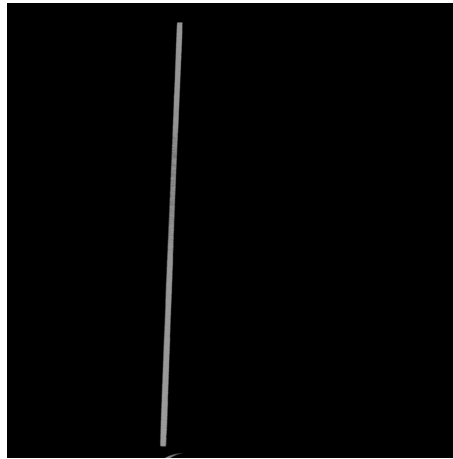


Figure 4.23: Sharp LED line contrasting with dark background.

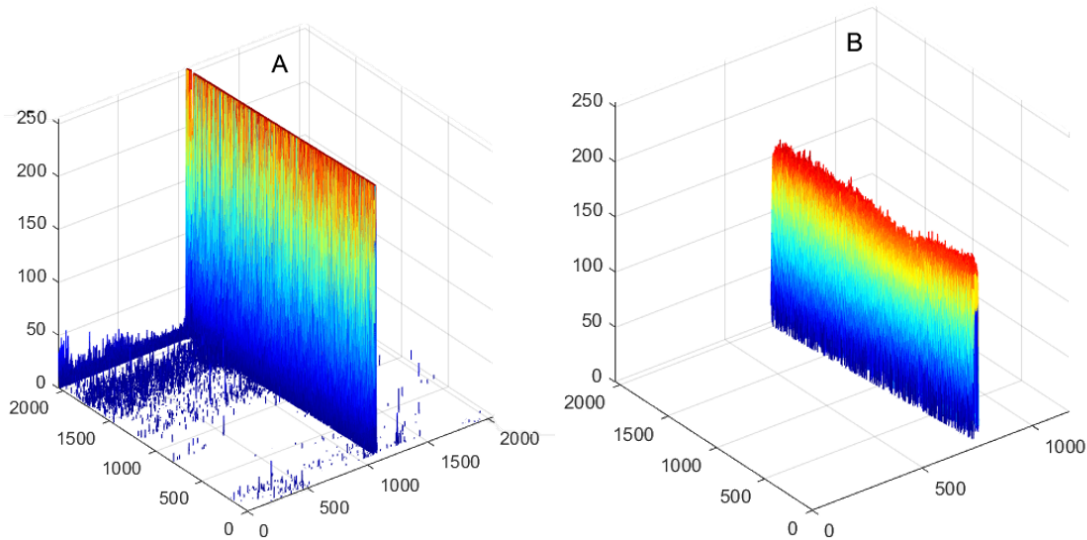


Figure 4.24: Pixels intensity of a saturated LED line (A) and a not saturated LED line (B).

first will make the algorithm save the coordinates of the pixels positioned on the right edge of the LED line, and the last will make the algorithm save, in each horizontal line of the illuminated pixels, the pixel of maximum intensity only, and if there are several with maximum intensity, it is chosen the midpoint in-between, something that happens when the LED line is saturated. Considering the right edge would require a very precise choice in the threshold value, on the other hand, considering only the maximum makes it crucial to obtain unsaturated images, which is more accessible, so this was the option chosen throughout the measurements.

Having acquired the illuminated points, the camera intrinsic matrix, the speed values along the three directions and the equation of the luminous plane, one can determine the 3D coordinates of each point of the cloud. First, the depth coordinates are determined, by substituting the image coordinates of each point, $\tilde{u}$ and $\tilde{v}$, into the luminous plane equation 4.6, originating equation 4.7:

$$Z = \frac{-D}{\frac{A\ddot{u}}{\alpha} + \frac{B\ddot{v}}{\beta} + C} \quad (4.7)$$

Then, the remaining camera coordinates are given by:

$$X = Z \frac{\ddot{u}}{\alpha} + \frac{vel_x}{f_{rate}}(i-1) \quad (4.8)$$

$$Y = Z \frac{\ddot{v}}{\beta} + \frac{vel_y}{f_{rate}}(i-1) \quad (4.9)$$

where vel_x and vel_y are the speed values along the corresponding axes, f_{rate} is the image acquisition frame rate and i is the image number ($i=1,2,...,n$). The right side of equations 4.9 and 4.10 include a speed related term, that compensates an eventual misalignment between the specimen and the optical axis, whereas in equation 4.8 this term guarantees the length of the 3D point cloud. Chapter 5 shows the difficulties that imprecise speed calibration arise, as the resulting 3D point clouds are tilted.

$$Z = Z + \frac{vel_z}{f_{rate}}(i-1) \quad (4.10)$$

Ended the process, the algorithm exports a .txt file with the coordinates of each point of the 3D point cloud, which can be transferred to a post-processing software, such as MeshLab.

Chapter 5

Results

5.1 Actual Depth Resolution

Once the system is assembled, it is possible to calculate the resulting depth resolution, res . Munaro et al. (2015) show a possible and viable way of calculation, through expression 5.1, where the angular resolution, θ_{res} , is given by $\theta_{res} = \tan^{-1} \frac{sz_{px}}{F}$. Having the FWD and the pixel size, sz_{px} , already determined in chapter 4, it is only missing the focal length, F . Note that considering only the lens focal length is not enough in this case, since the system focal length is affected by the use of 2 extensions and one focusing unit (smart focus).

$$res = FWD \frac{1}{\frac{1}{\tan(\phi_c) \tan(\theta_{res})} - 1} \quad (5.1)$$

According to MathWorks (2020b), $\alpha, \beta = F \times s_{x,y}$, where $s_{x,y}$ assigns the number of pixels along the respective direction, per world unit, which is millimeter in this case. It is necessary to calibrate the camera to reach the intrinsic matrix, something that will be performed in the next section.

Regarding that $\alpha \approx \beta = 1.4 \times 10^4$ and that there are 2048 pixels along 11.264 mm of sensor width, regardless of the direction considered, we have that $s_{x,y} = 2048/11.264 = 181.8$ pixels/mm, thus, the obtained focal length is **F=77**.

About the ϕ_c angle, it is known that it is complementary to the triangulation angle, represented in figure 5.1. Being that $\phi_c = 90^\circ - \phi$, it is possible to determine res values depending on ϕ . Thus, table 5.1 shows the calculated res value for $\phi = 30^\circ, 45^\circ$ and 60° , a value that increases the wider ϕ is.

Table 5.1: res value for different ϕ

$\phi(^{\circ})$	res (mm)
30	0.0299
45	0.0172
60	0.01

Note that 60° is the ϕ value that provides the best *res* to the system, 0.01 mm, however, as studied at the beginning of chapter 4, to successfully perform contour measurements, the built system should have a minimum depth resolution of 0.001 mm, i.e., a 10x smaller *res*.

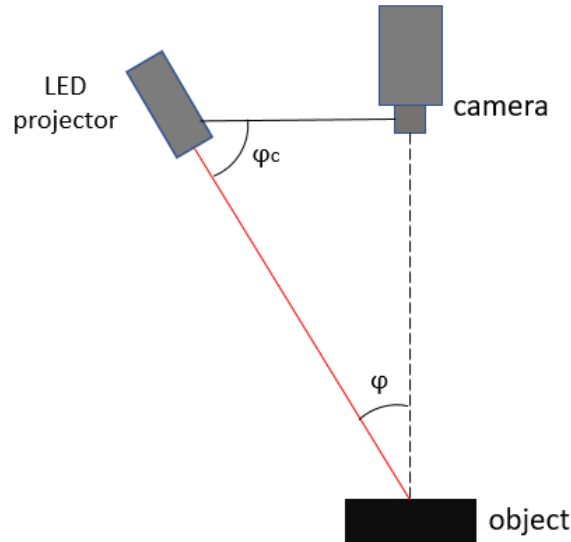


Figure 5.1: Triangulation scheme showing triangulation angle (ϕ) and its complementary (ϕ_c)

5.2 CT specimen

The first tests were performed resorting to a CT specimen, aiming at the superposition between the acquired point clouds and their respective CMM point clouds. The studied specimen has been used in two previous works, Braga (2012) and Rich-Trummer and Castro (2011), and so it was already available at the beginning of this project, as well as its contour CMM measurements. The specimen was extracted from a S355NL steel plate welded under similar conditions to those used for a railway bridge. Moreover, Rich-Trummer and Castro (2011) name this specimen as CT1 and analyze the RS distribution through its thickness. Figure 5.2a shows the specimen geometry as well as its dimensions, and figure 5.2b shows the referential used in the CMM measurements. Table 5.2 summarizes the material properties that are useful for the linear elastic FEM calculations and for the error determination.

From the linear tables available, the one with the greatest displacement range (0-100 mm) is not enough to perform the scan of the entire specimen surface, so it was necessary to couple another 0-20 mm linear table through 4 M4 studs. Next, it will be explained successive steps to improve the acquired point clouds, in order to validly determine the residual stress distribution.

5.2.1 3D Point Clouds

The first acquisition was performed at 2 fps so the low point cloud density allows a better visualization of its points. Moreover, the first point cloud presented in this section was acquired using

Table 5.2: CT specimen properties

Properties	
σ_y [MPa]	400
E [GPa]	210
ν	0.3

a triangulation angle of about 45° . Figure 5.3 shows the point cloud of the CT specimen A half, just after its acquisition, without any transformation and with about 80000 points, some of which do not belong to the studied surface and consequently must be removed. This first acquired cloud, assigned as c1A, will then be subjected to successive transformations so that it can be compared to its respective CMM cloud. The system parameters resulting of the pre-measurement calibration are not presented, since it will be necessary to resort to several re-calibrations later.

Figure 5.4 shows the c1A point cloud in stage 2, as it is the second round of subjected transformations, that results from two applied transformations: removal of external points and the re-positioning along the positive X and Y semi axes.

At stage 2, one may start noticing anomalous rotations of the point cloud along X and Y axes, which arise due to the poor accuracy of the speed values obtained in the speed calibration, explained in chapter 4, and which is detrimental to the point cloud shape visualization. Several calibration attempts were performed in order to obtain valid speed values, but the resulting point clouds would always show anomalous rotations. Since these rotations are supposed to be corrected by the second terms on the right side of equations 4.9 and 4.10, which are terms linearly depending on the determined speed values, they do not perform it accurately.

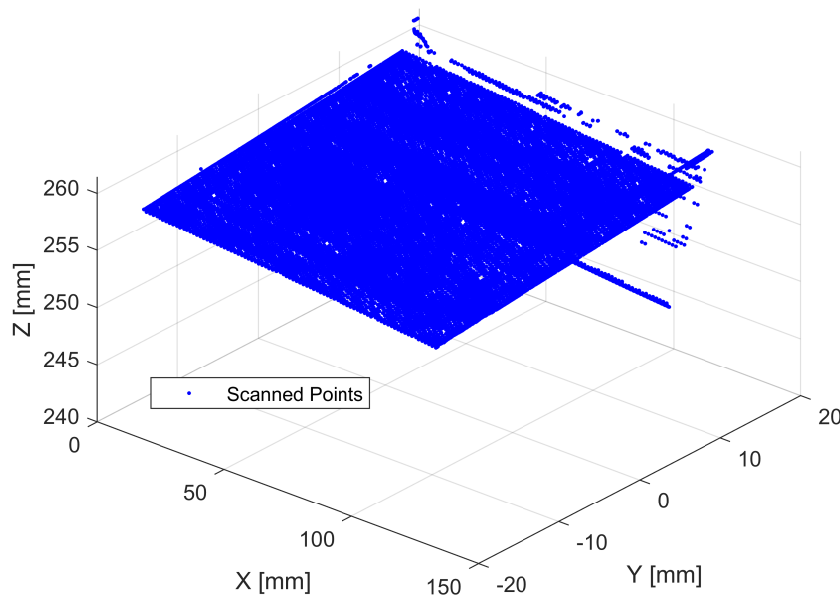


Figure 5.3: c1A point cloud.

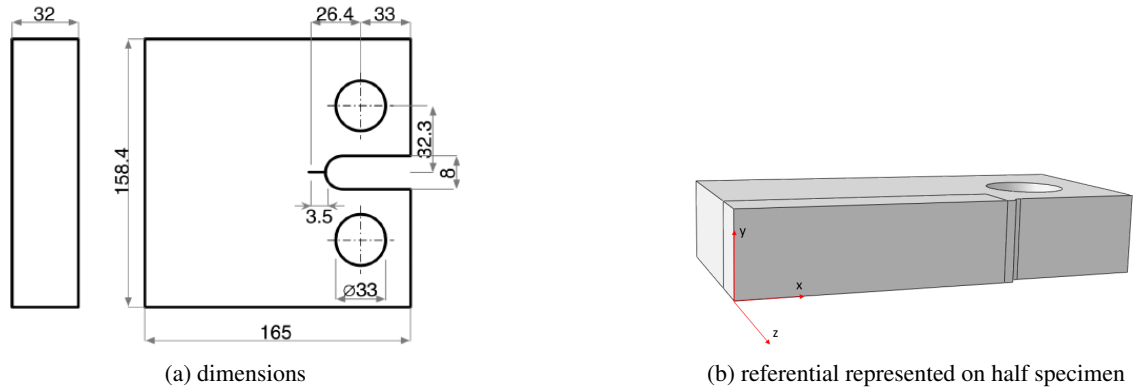


Figure 5.2: CT specimen details.

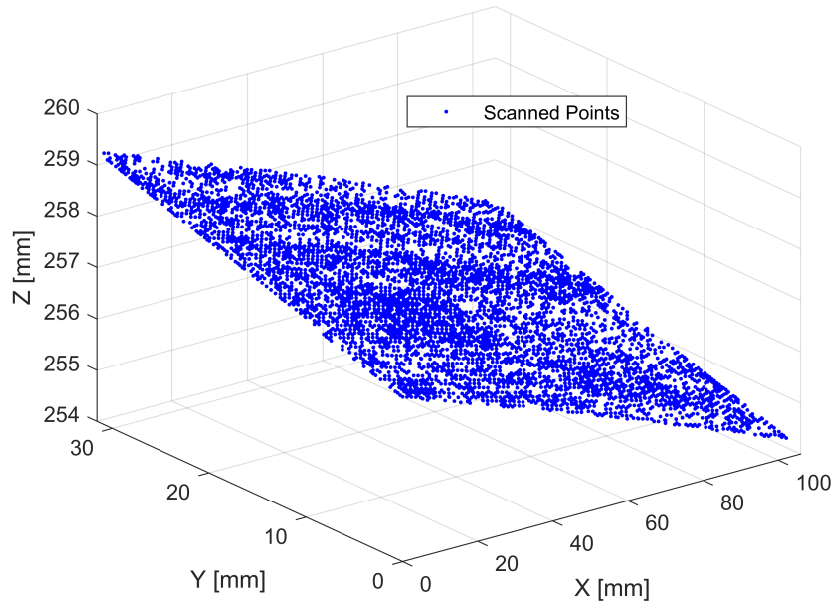


Figure 5.4: Stage 2 of c1A point cloud.

To obtain the scanned point cloud overlapped to its corresponding CMM point cloud, it is necessary to superimpose their referential axes. Moreover, one should notice that the captured images come symmetrical relative to the Oxy plane, since the depth axis orientation of the developed system is opposite to the depth axis orientation considered in the CMM measurements. Thus, it is necessary to convert the Z coordinate of each cloud point j , to its symmetric, applying the transformation: $Z_j = -Z_j$. It should also be noted that the Z coordinates of the cloud points are supposed to assign local displacements relative to the specimen cut plane, thus, it is necessary to translate the clouds along the Z axis, in a value equivalent to the distance between the cut plane and the optical center. As this translating distance is not known, in a first analysis this value will be arbitrated so that the scanned and CMM clouds touch each other, figure 5.5. After applying the

aforementioned transformations to the cloud c1A in stage 2, it is obtained in stage 3.

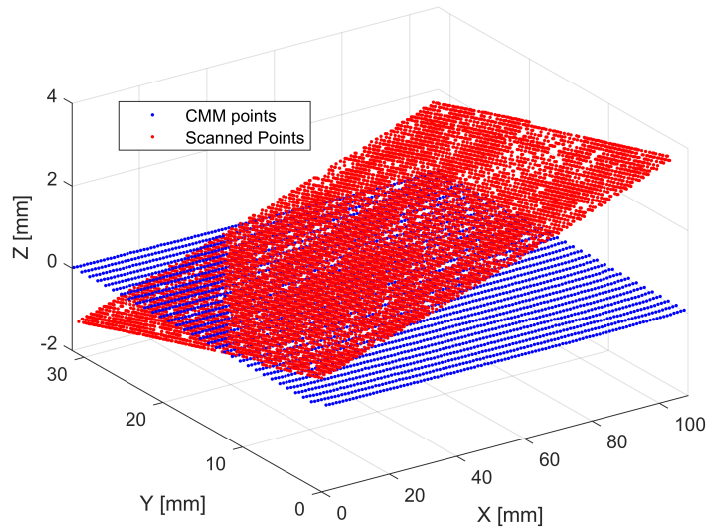


Figure 5.5: c1A point cloud in stage 3 versus its respective CMM point cloud.

Ignoring the speed values along Y and Z directions, calculated by the calibration algorithm, by setting them to zero, it is obtained the c1A cloud in stage 4, shown in figure 5.6. At this stage, the cloud shape is noticeable for the first time and one may also notice that the rotation along the Y axis has decreased, which proves the inefficiency of the speed calibration. The rotation along the X axis has remained almost unchanged, which indicates a bad positioning between specimen and camera.

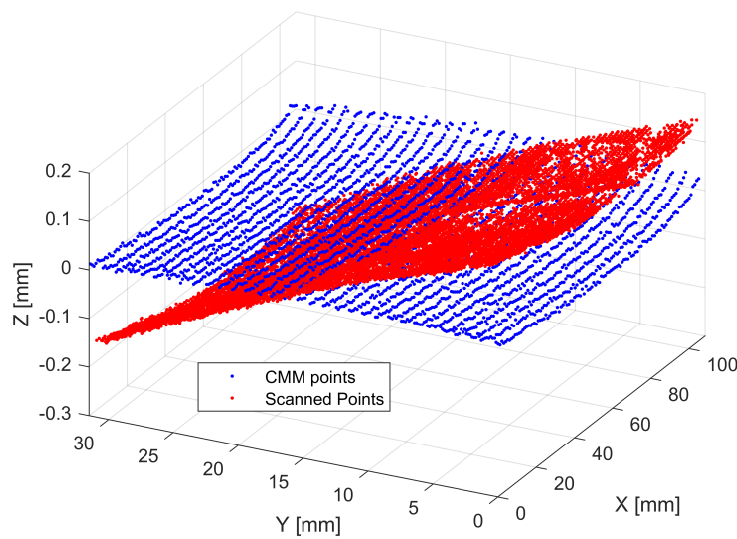


Figure 5.6: c1A point cloud in stage 4 versus its respective CMM point cloud.

The former system configuration, presented in chapter 4, shows that the fastening applied to the camera does not provide alignment along any of the axes, mainly along the Y axis, and so the camera fastening had to be improved. A fixable structure was assembled to the assembling table, using 4 M6 screws. This structure has a platform movable along the X direction, which facilitates camera re-positioning relatively to the LED projector in case of changing the triangulation angle, ϕ . The need to assemble two linear tables located the specimen 60 mm above the assembling table, so it was necessary to elevate the camera, which was achieved by adding two standard Thorlabs bases, a BA1S, figure 5.7a, and a BA1, 5.7b. The used camera has only one hole for fixing purposes, which deteriorates the alignment along the Y axis. After assembling the camera's fixing structure, figure 5.8a, the parallelism between the upper face of the camera and the assembling table was confirmed, through a level, figure 5.8b. Then, a system re-calibration was performed and a new surface contour measurement of the A half took place, resulting in the c2A cloud.



(a) BA1S



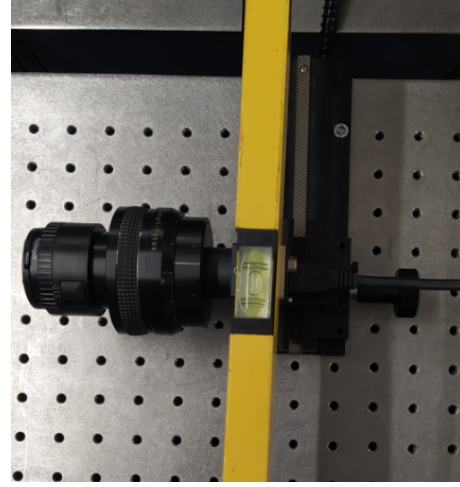
(b) BA1

Figure 5.7: Thorlabs standard bases.

The graph of figure 5.9 is rotated -90° along Z direction relatively to the graph of figure 5.6 in order to facilitate the point cloud visualization. It shows that the change in the camera fastening resulted in the attenuation of the cloud rotation along the X axis, on the other hand it increased the rotation along the Y axis. By choosing the two extreme points of the scanned point cloud, regarding the X direction, the correction angle, α_A , needed to rotate along the Y axis was roughly determined through $\alpha_A = \tan^{-1} \frac{\Delta z}{\Delta x}$. Note that the c2A point cloud, shown in figure 5.9, was subjected to the same transformations as the c1A cloud in stage 4, but in this case the speed corrections were ignored.



(a) Fixable structure



(b) Level measurement

Figure 5.8: Camera fastening.

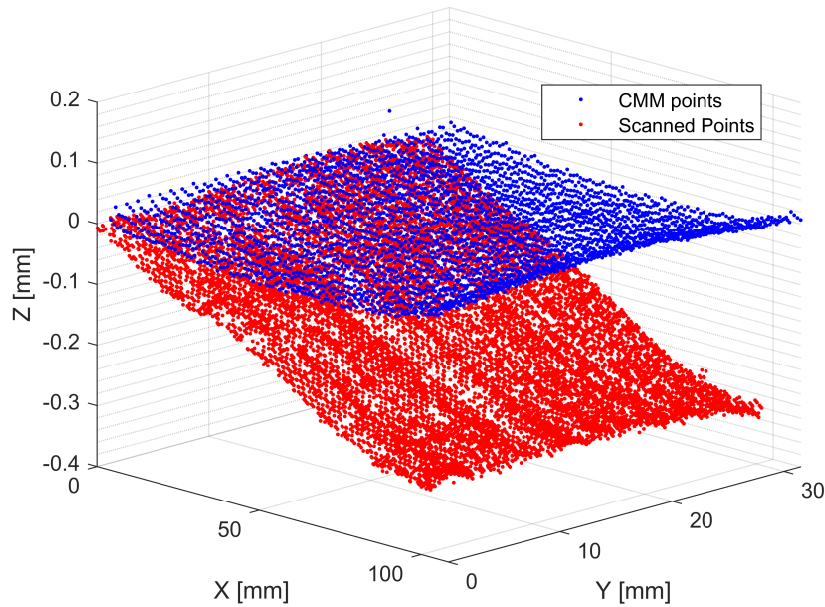


Figure 5.9: c2A point cloud in stage 4 versus its respective CMM point cloud.

Remember that all transformations applied, up to this stage, are rigid transformations, i.e., they do not change a point cloud shape. However, the applied transformations were determined based on trial-and-error, using the CMM point cloud as reference, which has no practical validity, since it is supposed not to need alternative systems' measurements to correct the determined results. Still, these transformations are useful to draw conclusions about the shape of the scanned clouds.

Applying rotation α_A along the Y axis to the cloud of figure 5.9, it is obtained in stage 5, which is represented in figure 5.10a. One may confirm that the cloud shapes are similar, however, an anomalous "bump" within $20 < x < 60$ mm appears, which was initially thought to be due to table vibration, during the acquisition process. The linear tables controller, TL78, was also placed on the assembling table and its operation could have caused unwanted vibration on it, so the controller was moved to other table. Once the absence of vibration was guaranteed, it was measured the contour of the B half, 5.10b, to corroborate both the matching cloud shapes and the cause of the "bump". The c2B cloud was corrected with different translation and rotations from those of the c2A cloud. The c2B cloud shape is in fact close to its respective CMM cloud's, however, the "bump" remained present within the same positions, thus, it was concluded that vibration was not the cause.

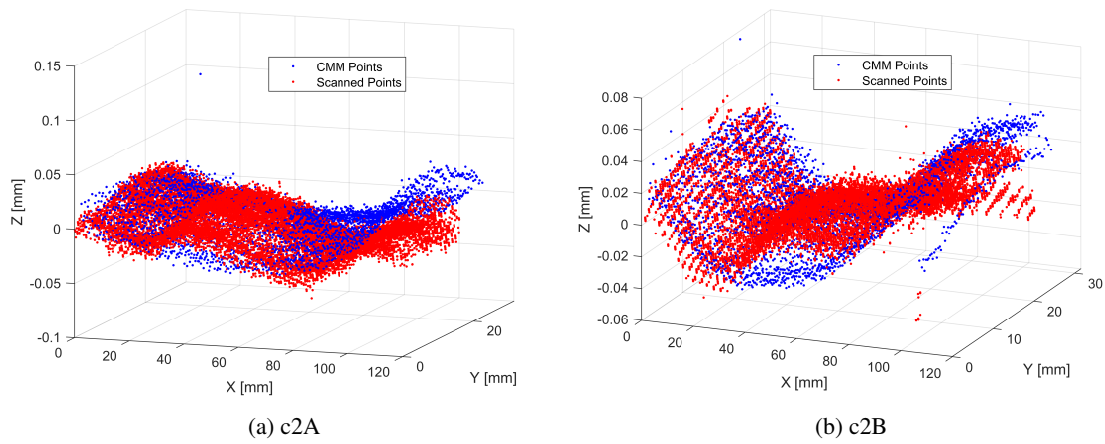


Figure 5.10: c2A and c2B point clouds in stage 5 versus their respective CMM point clouds.

Other possible cause for the presence of the "bump" is the rigidity of the cables that connect the linear tables to the controller, as they are long (about 3m each) and somewhat rigid. When the lower table's moving plate moves, the whole upper linear table also does, and as such, its cable moves as well, which may stick to the other cable or to the assembling table. To avoid this effect, the linear tables have been fixed to the assembly table, as shown in the figure 5.11. Note that, ideally, it should be used a single linear table with a larger displacement range. Moreover, two cylindrical head screws, with the same diameter, have been screwed to the upper moving plate. Putting each specimen half in touch with these screws will guarantee that both halves are measured considering the same reference plane.

Fixing the linear tables to the assembling table implied a re-adjustment of the camera position, so it was necessary to re-calibrate the system. Taking advantage of the fact that a new calibration was necessary, the LED projector was re-positioned to form a triangulation angle of approximately 60° , in order to obtain the maximum possible resolution, with this setup (MVTec).

The intrinsic matrix, A , in 5.2, results from the camera re-calibration, which registered a distance of about $\Delta z = 262$ mm between the calibration target and the optical center. Figure 5.12

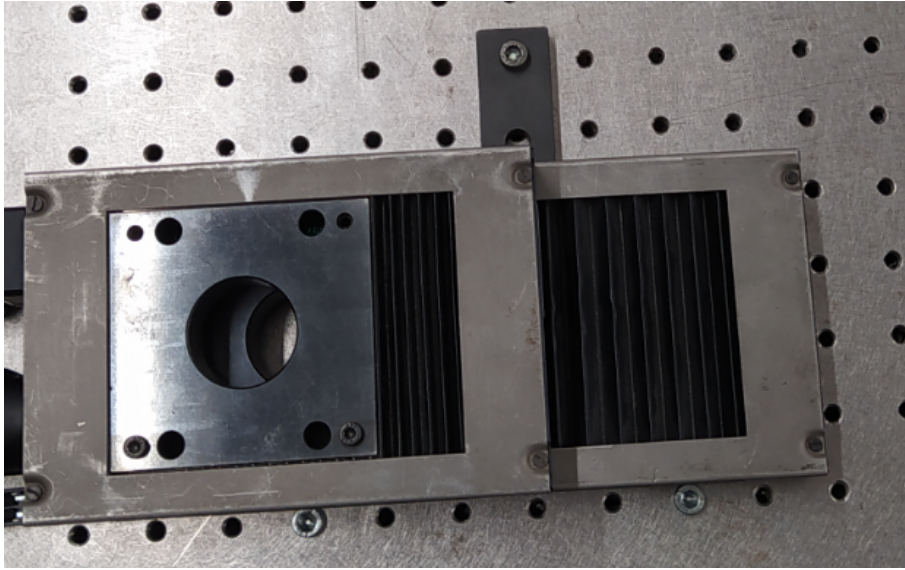


Figure 5.11: Fastening of the linear tables.

shows the displacements along the three directions, resulting from the speed calibration, confirming the linear displacement along X. The ideal result regarding the Y and Z displacements would be a horizontal lines, meaning constant position, which would give rise to null speed values along these axes. However, the calculated speeds were $vel_y = 0.015m/s$ and $vel_z = -0.028m/s$ which being considerably small, indicate a good specimen positioning. The scanning speed, calculated along the X direction, was $vel_x = 2.65 \text{ mm/s}$.

$$A = \begin{bmatrix} 1.39e4 & 0 & 0 \\ -52.84 & 1.40e4 & 0 \\ -17.60 & 1.23e3 & 1 \end{bmatrix} \quad (5.2)$$

Finally, the LED projector calibration indicated the actual distance between this and the camera, $\Delta x = 446 \text{ mm}$. The triangulation angle is given by $\phi = \tan^{-1} \frac{\Delta x}{\Delta z} \approx 59.5$, which is very close to the actual angle, and therefore it is considered a valid calibration.

After the re-calibration, the point clouds were re-acquired, this time at 7fps, ensuring that:

1. The assembling table does not vibrate;
2. The controller cables do not interfere in the measurement;
3. A smooth motion switch between linear tables;

Then, c3 point clouds, A and B, were obtained, figure 5.13. They were subjected to the same transformations as the clouds of the figure 5.10, but with different angles and offsets, using their respective CMM point clouds as reference.

The next step to improve the resulting point clouds is to assign them the right corrections, without taking the CMM point clouds as reference. A method similar to the one applied in Sousa

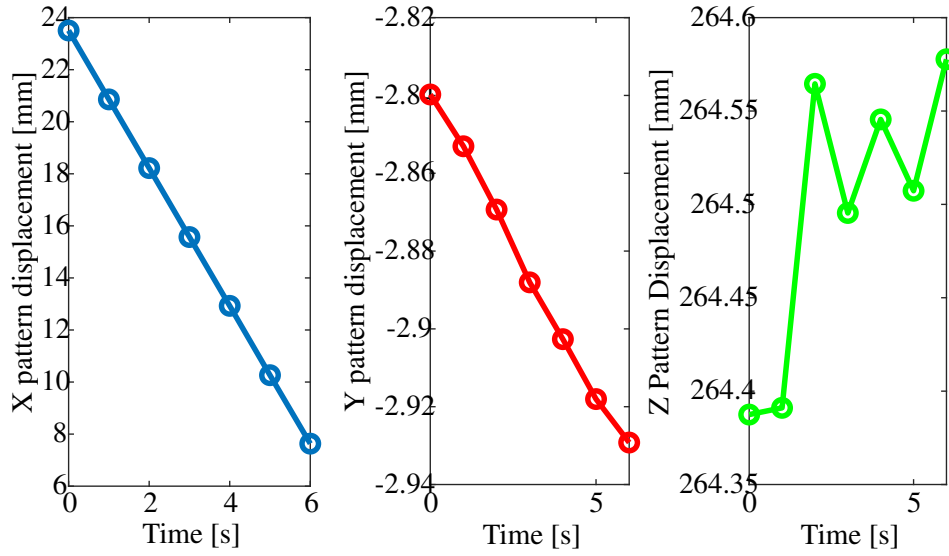


Figure 5.12: Displacements registered along the 3 directions.

et al. (2018) suggests the creation of a reference surface, according to which are calculated the rotations that best approximate the point cloud's best-fit plane to that same reference. This method resorts to a least squares regression, aligning the cloud referential with the referential of the created reference. It should be noted that this method corrects both cloud rigid rotations and their offset along Z, and so it is no longer necessary to know the position of the optical center within the lens. The algorithm used to correct the point clouds is attached the Appendix C.

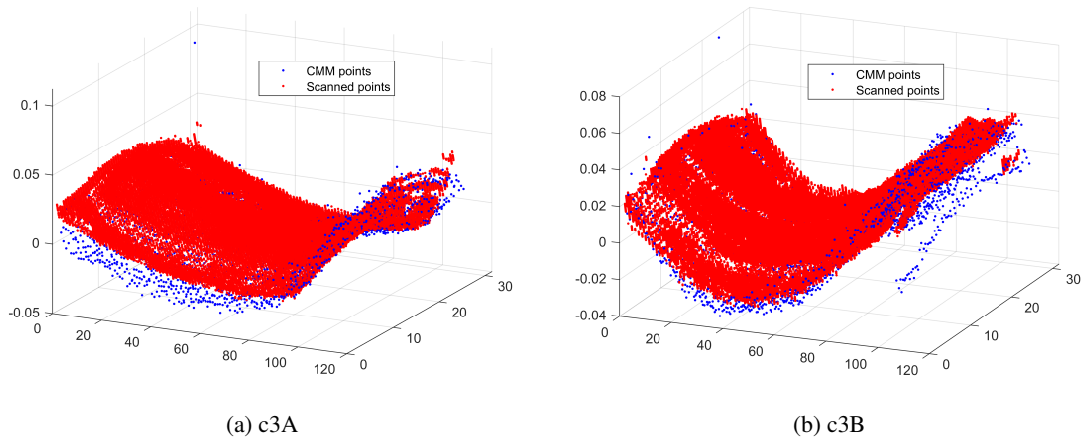


Figure 5.13: c3 point clouds in stage 5 versus their respective CMM point clouds.

It was assumed that the CMM clouds had the Oxy plane as reference, thus, the correction method was applied to the acquired A and B point clouds taking as reference that same plane, using the CMM clouds only as benchmark. It is observed on figure 5.14a that the A half point clouds are overlapped, therefore, the correction worked well. On the other hand, figure 5.14b

shows considerable disparities between the B half point clouds, which indicates that the reference plane of the CMM point cloud is not the Oxy plane, but a slightly tilted plane. However, it is known that, before calculating the displacement boundary conditions, both halves must be aligned to the same plane (Oxy), before being averaged, (Braga, 2012), and applying this transformation to the CMM clouds, the overlap between CMM and acquired clouds is finally obtained, figure 5.15.

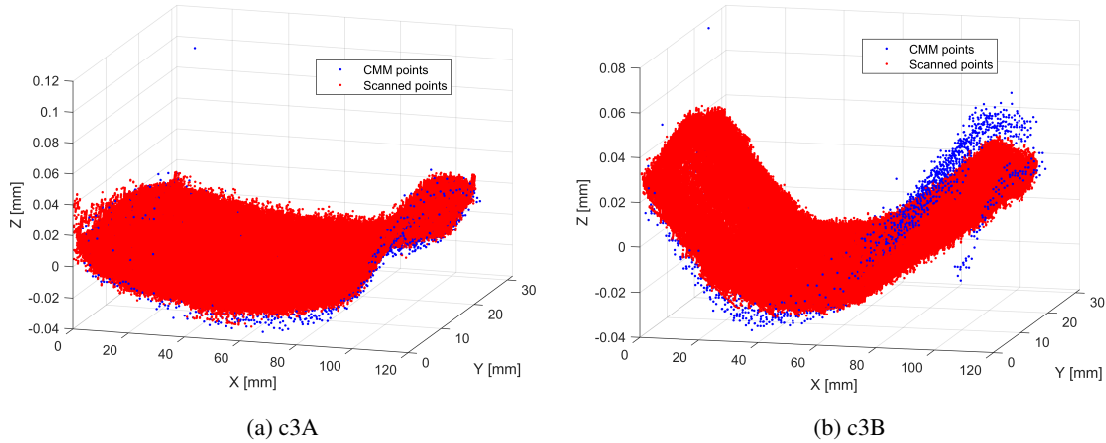


Figure 5.14: c3 point clouds in stage 6 versus their respective CMM point clouds.

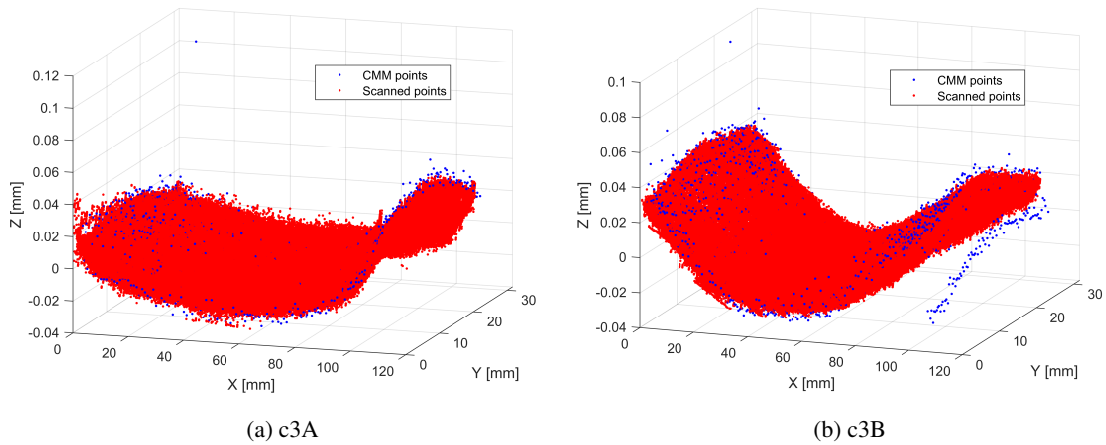


Figure 5.15: c3 point clouds in stage 6 versus their respective corrected CMM point clouds.

Having the post-processed clouds, i.e.: after correcting their symmetry related to the Oxy plane, after correcting their rigid rotations and offset along Z by applying the method shown in Sousa et al. (2018), having removed external and noise causing points and after re-positioning them into the positive X and Y semi-axes, the 2D map of normal stresses can be determined.

Since the disparities between CMM and acquired clouds are not visible to the naked eye, the relative error between node displacement values was calculated, having the CMM displacements

as the actual values. Thus, we obtained the color map of figure 5.16, which shows the distribution of the calculated relative error along the studied surface.

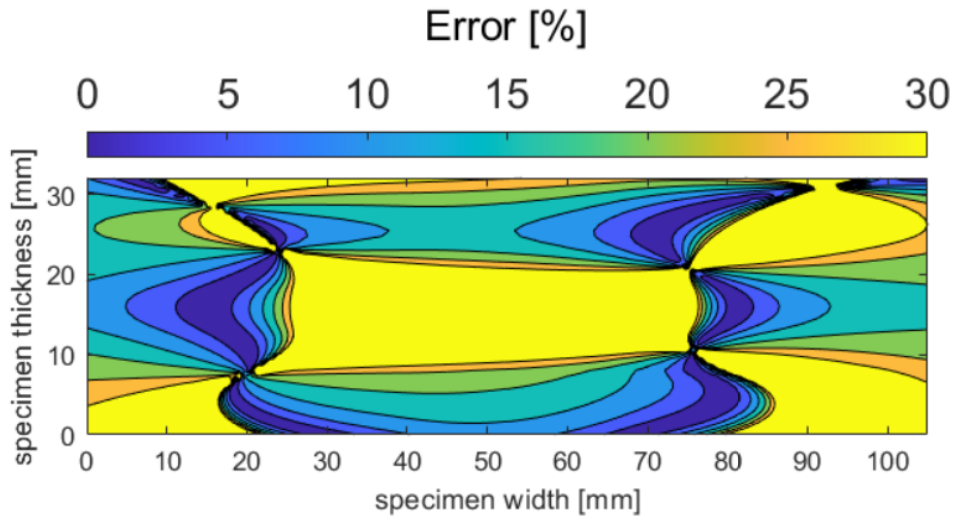


Figure 5.16: Displacement comparison along the cut plane.

It is noted that the most discordant areas, in terms of displacement, are located, in greater density, in the central zone of the surface and where $x > 70$ mm. This concentration of disparities may be due to:

1. Motion disruption during acquisition;
2. Inefficient point cloud cleaning;
3. Wrong re-positioning along the positive X and Y semi-axes;

Points 2. and 3. are interlinked, since an ineffective cloud cleaning may make it impossible to re-position the cloud correctly. Cloud cleaning consists in removing acquired points that do not belong to the surface and/or points that clearly represent artifacts of the measurement process, i.e. points that have a Z coordinate significantly different from neighbouring points. The point removal, being applied more or less than necessary, may influence point 3., since the re-positioning is done having the points with the lowest X-Y coordinates as reference. Point 2. can, however, negatively influence the results on its own, since different cloud clearances give rise to different projections of these on the Oxy plane.

Point 3. has already been mentioned throughout the document, i.e. the motion disruption due to the need of 2 linear tables instead of just one. Since the motion switch is up to the system operator's timing, it is a source of measurement errors. If the transition is made at the wrong instant, it is perceived by equation 4.8 that the cloud formation will not take place correctly, and consequently, the cloud correction will be affected.

Overall, points 2. and 3. depend only on the operator's experience, but point 1. can be completely corrected by using a single linear table. For this purpose, a specimen whose studied surface is only 20 mm long will be measured further in this document.

5.2.2 Residual Stress Field Comparison

After acquiring and post-processing the c3 clouds of both specimen halves, shown in the previous section, Linear finite element analysis (FEA) was performed with Abaqus software package, using a mesh of $0.4 \times 0.73 \times 0.4$ mm C3D8R elements. The algorithm used to calculate the displacement boundary conditions is attached to the Appendix D.

Recalling to the beginning of chapter 4, when it comes to residual stress, it was used the absolute error divided by the material yield stress, 400 MPa, having 10% as maximum value, to check validity. Thus, figure 5.17 shows the color map with the error distribution along the cut plane.

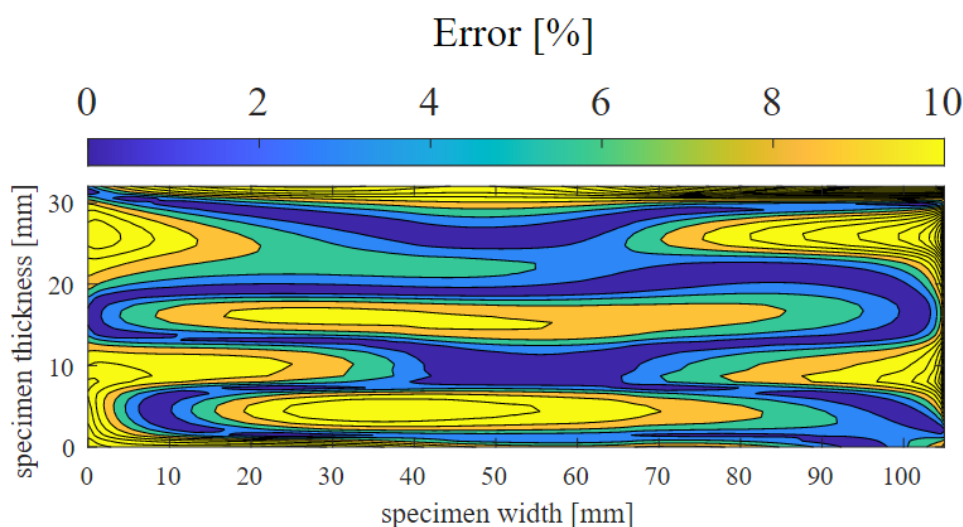


Figure 5.17: Residual Stress comparison along the cut plane.

Observing the stress distribution, one can notice that even though the areas of greatest stress error are concordant with the areas of greatest displacement error, they have smaller size, which indicates that many of the observed displacement disparities occur in nodes whose absolute stress values are not significant when compared to the material yield stress.

Adding to the shown distribution, it should be known that about 4000 out of 11925 mesh nodes present an error higher than the admissible 10%. The maximum error found was 59% and its average was 8%.

5.3 AISI D2 Plate

To corroborate the results obtained with the CT specimen, the same analysis was performed with a specimen supplied by Quantal S.A., which also supplied the CMM measurements used as benchmark. This plate results of a rolling technique, therefore, the stress magnitude denoted over the cut surface is expected to be relatively low. It presents two notches, which have been performed through wEDM, as well as the half length cut. The studied specimen consists of a plate of an AISI D2 tool steel, whose properties are presented on table 5.3. Figure 5.18a shows the dimensions of

the studied specimen, to which was applied a half-length cut, resulting in two 20x3x50 mm halves, considering the referential of figure 5.18b.

Table 5.3: AISI D2 properties

Properties	
σ_y [MPa]	470
E [GPa]	190
ν	0.28

Since the analyzed surface of this specimen is 20x3 mm, it is no longer necessary to use more than one linear table to perform the scan, therefore, the error affecting the controller operation will be totally eliminated.

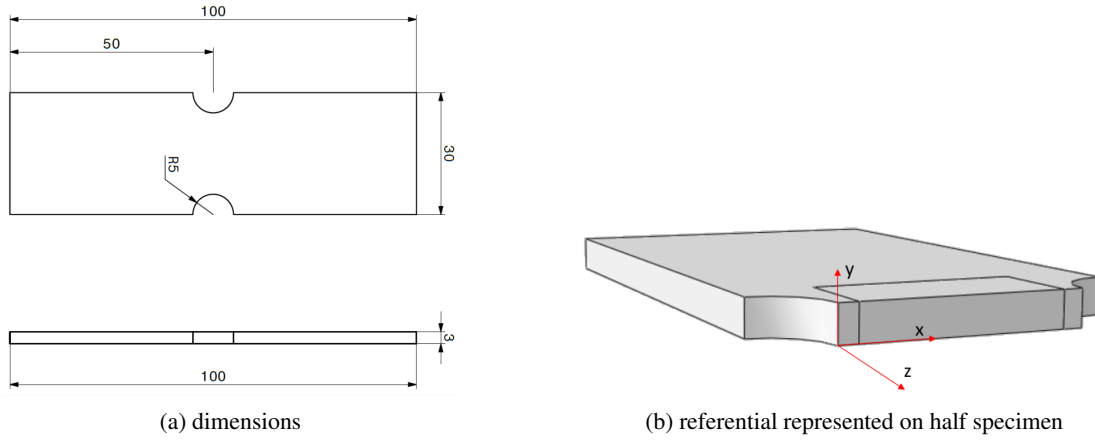


Figure 5.18: AISI D2 plate specimen details.

5.3.1 3D Point Clouds

The 0-20 mm linear table was removed from the system and, as such, it was necessary to lower the camera height. Thus, it was necessary to re-calibrate the system once again. The resulting parameters were not significantly different from the previous calibration, so they will be omitted.

Having a surface with a significantly smaller area, when compared to the CT specimen, demands higher point density for its reconstruction, so that the cloud shape can be perceptible. Having $vel_x = 2.65$ mm/s and keeping the acquisition frame rate at 7fps, one would obtain 2.64 frames/mm, that is, 2.64 points per millimeter, along the X direction, contrasting with the 10 points per millimeter of the CMM measurements along the same direction. For this reason, a laptop computer with a USB3.0 port was used, in order to take advantage of the 30 fps of the camera, making 11.3 points per millimeter along X.

After acquiring both halves point clouds, the same set of transformations applied to the CT specimen clouds are applied to the present specimen's, resulting in the point clouds shown in figure 5.19.

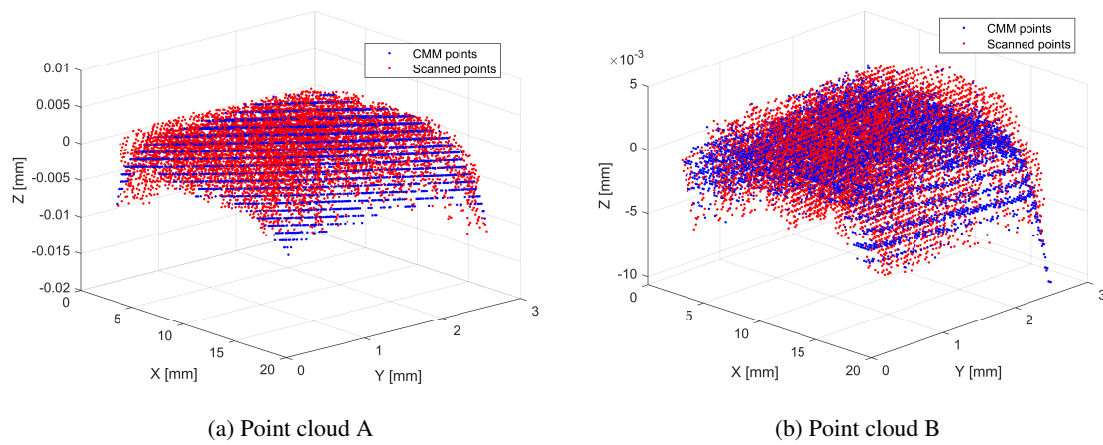


Figure 5.19: Scanned point clouds versus their respective corrected CMM point clouds.

5.3.2 Residual Stress Field Comparison

Having the point clouds rectified, a linear finite element analysis was performed to determine the residual stress distribution. Similarly to what was done with the CT specimen, a coarse mesh was defined for the generality of the specimen, in this case, with 2 mm spacing, and then a partition was created to bear a finer mesh of C3D8R elements with 0.2x0.2x0.2 mm spacing. The algorithm used to calculate the displacement boundary conditions was based on the algorithm in the Appendix D, changing only parameters related to the specimen geometry or mesh.

The RS distribution was calculated resorting to the CMM data, in order to verify its magnitude. Figure 5.20 shows that, in general, the specimen presents a low variation of RS along its cross section, as expected. In contrast, it is observed increasing RS variations nearby the specimen edges.

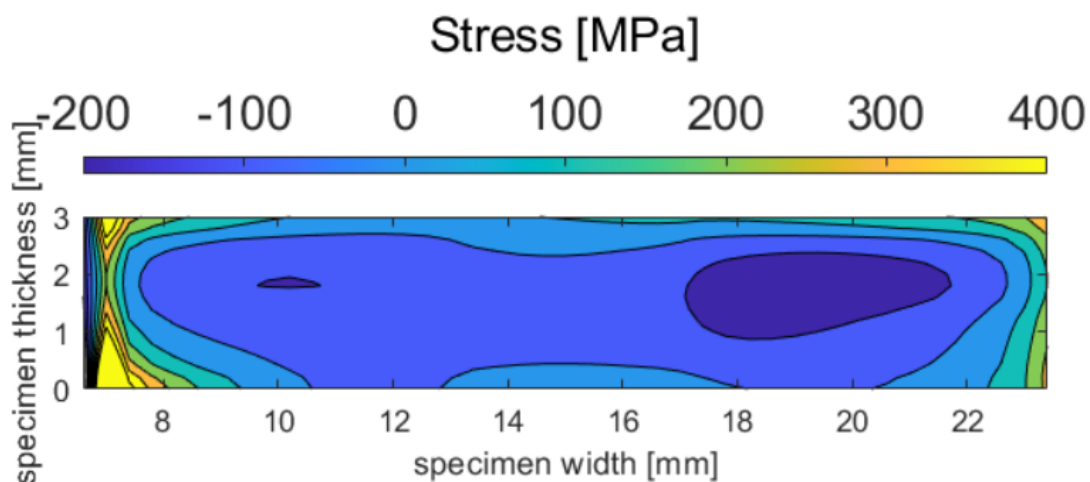


Figure 5.20: Residual Stress distribution along the cut plane.

Observing the RS error distribution along the surface, figure 5.21, and comparing it to the RS error distribution of the previous specimen, figure 5.17, it is noticeable that the critical area, i.e., with an error $>10\%$, is proportionally smaller, and is located mostly near the vertical edges. This error distribution is justified by the cloud shapes and the Z values range. Note that the points near the surface edges have the most significant depth variations, that are in the order of 0.005 mm, which is a value that the actual system depth resolution, 0.01 mm, may not reach. Moreover, the disparities denoted next to the top and bottom of the surface might exist due to wire entry/exit artifacts (Bouchard et al., 2012), which could be mitigated if the wEDM cut was performed using sacrificial layers (Toparli and Fitzpatrick, 2016).

In 1360 mesh nodes, only 271 showed errors $>10\%$, and the average value of error dropped to 6.8%, however, it is not possible to affirm that the results have improved relatively to the CT specimen, because the maximum error detected was 150%.

Overall it can be stated that the developed system does not present valid results for the studied specimens. Having eliminated three possible error sources (cables rigidity, multiple linear tables and table vibration), it remains uncertain the performance this system would have in case of matching the minimum depth resolution required. According to the study performed at the beginning of chapter 4, a resolution worse than 0.001 mm would not allow valid results to be obtained, which was confirmed.

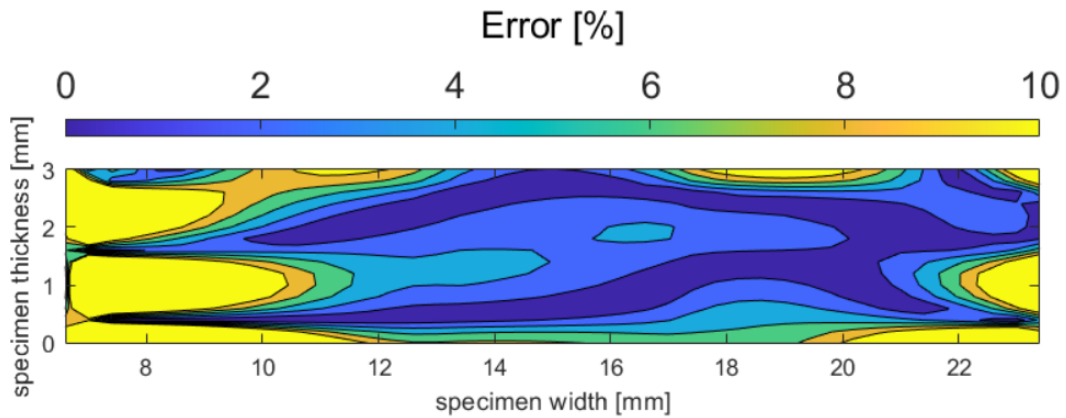


Figure 5.21: Residual Stress comparison along the cut plane.

Therefore, the system's depth resolution remains to be improved, something that implies changing one or many of the following parameters: triangulation angle (ϕ), pixel size (sz_{px}) or focal length (F). In the absence of time to proceed, this study will be explained in the future works section, in chapter 6, describing its main steps.

Chapter 6

Conclusion

6.1 Concluding Remarks

The present work focused on the development of a surface topography measuring system, based on laser triangulation. The developed system was built resorting to 3 main units: the movement unit (linear tables), the acquisition unit (camera+lens), and the luminous unit (LED projector). After assembling the system and calibrating the camera, the resulting intrinsic matrix was used to calculate the maximum depth resolution available, which turned out to be 10x worse than the minimum 0.001 mm required, calculated a priori in chapter 4. Having this in mind, it was expected to get an improved resolution by applying a Subpixel Method to the analysis of the AISI D2 specimen, which was ultimately not possible due to lack of time. As such, the author of the present work chose to explain such improvement in the future works section.

The first specimen measured was the CT specimen, from which general conclusions were drawn about the system performance and the point clouds transformation. After applying several trial-and-error based transformations to the clouds, in order to evaluate the clouds shapes, it was concluded that all point clouds have to be subjected to the following set of transformations:

1. Removal of external points and artifacts;
2. Revert the cloud symmetry along the Oxy plane;
3. Re-positioning along the positive X and Y semi-axes;
4. Correction of the rigid rotations;

In the Contour method, the point clouds represent surface normal displacements, thus, taking as reference the zero of that direction. For this reason and because the system depth is along the Z axis, the Oxy plane is used as the reference plane for the correction of the rotations.

The first cloud acquired in stage 2, figure 5.4, suggests insufficiency on system's speed calibration accuracy, which was confirmed with stage 4, figure 5.6, where both vel_y and vel_z were set to zero. Although the errors introduced by vel_y and vel_z are corrected by the rotations correction method, the error introduced by the vel_x value, resulting from the calibration process, is unknown.

The second couple of acquired clouds, figure 5.10, confirmed the existence of a systematic error affecting the acquisition, caused by the rigidity of the connecting cables. This error was solved by fixing the linear tables to the assembly table. Although it has not been proved that the vibration coming from the controller operation would jeopardize the measurements, it is advisable to operate it from an independent table/platform.

The c3 clouds, figure 5.15, was used to apply the Contour method, resulting in the distribution of displacement and residual stress errors. The comparison between these distributions allowed one to verify that, although the central area of the specimen has the most nodes with displacement disparities, many of them have absolute values of stress that are not significant when compared to the material's yield strength. In terms of residual stress error, there were 4000 out of 11925 nodes with error $>10\%$. Moreover, the average error was 8% and the maximum error detected was 59%.

The AISI D2 specimen dimensions contributed to eliminating the eventual error caused by the use of two linear tables. The analysis of the RS error distribution allowed one to verify that the anomalous areas are concentrated at the ends of the specimen, which can be explained by the shape of the point clouds. The clouds remain practically horizontal, except in the specimen ends, where they suddenly evolve to displacements in the order of 0.005 mm, that is, values out of the resolution's range. Although the number of nodes with error $>10\%$ has proportionally decreased to 217 out of 1360 and the error average dropped to 6.8%, it can not be concluded that there was improvement in results, since the maximum error registered was 150%.

Overall, although it is concluded that the developed system is not yet suited for residual stress inspection, the acquired clouds shapes have proved to be concordant with the CMM clouds. Moreover, it is concluded that the stress distributions obtained corroborate the disparity between the system's available resolution and the minimum resolution required, remarking that most of the analyzed nodes showed error values $<10\%$. It should be remarked that cleaning and re-positioning the clouds are operations that highly depend on the operator experience, but they are not worth of optimization until the resolution obstacle is overcome.

6.2 Future Works

Insufficient depth resolution was the main cause for the residual stresses determined to have exceeded the 10% admissible error, and as such, it must be improved. The easiest way to improve resolution is to decrease the pixel size, which is achieved through a camera with better resolution or by applying a sub-pixel method, which consists of subdividing the sensor pixels into several sub-pixels and does not imply extra costs. Shortis et al. (1994) mentions 3 possible sub-pixel methods. Applying a sub-division of at least 1/10th of pixel, the obtained resolution would allow obtaining valid 2D RS maps.

Moreover, it is recommended the use of a professional calibration software, aiming at a more accurate scanning speed determination. For that purpose, one may use the MVTec machine vision software, HALCON, to mention one.

Bibliography

- Y.M. Amir and B. Thörnberg. High precision laser scanning of metallic surfaces. *International Journal of Optics*, 2017, 2017.
- P. Bouchard, P. Ledgard, S. Hiller, and F. Hosseinzadh Torknezhad. Making the cut for the contour method. *International Conference on Experimental Mechanics*, 10, 2012.
- D. Braga. *Contour Method: Procedure Optimization*. PhD thesis, Faculty of Engineering of the University of Porto, 2012.
- Calib.io. *calib.io*. <https://calib.io/pages/camera-calibration-pattern-generator>, July 2020.
- Effilux. *EFFI-Lase V2*. <https://www.effilux.com/en/products/3d/effi-lase-v2#optical>, July 2020.
- D. George and D. J. Smith. Through thickness measurement of residual stresses in a stainless steel cylinder containing shallow and deep weld repairs. *International Journal of Pressure Vessels and Piping*, 2005.
- F. Ghanem, C. Braham, and H. Sidhom. Influence of steel type on electrical discharge machined surface integrity. *Journal of Materials Processing Technology*, 142, 2003.
- iDS. *UI-3370CP*. <https://en.ids-imaging.com/store/ui-3370cp-rev-2.html>, July 2020.
- G. Johnson. *Residual stress measurements using the contour method*. PhD thesis, Faculty of Engineering and Physical Sciences of the University of Manchester, 2008.
- H. T. Lee, C. Liu, F. C. Hsu, and J. M. Hsu. An enhanced calibration scheme for the edm hole-drilling strain gage method for the measurement of residual stress in ferrous materials. *Materials Transactions*, 49, 2008.
- MathWorks. *Single Camera Calibrator App*. <https://www.mathworks.com/help/vision/ug/single-camera-calibrator-app.html>, August 2020a.
- MathWorks. *cameraParameters*. https://www.mathworks.com/help/vision/ref/cameraparameters.html#mw_70aaf3c3-a920-4047-8310-ce5fddf5b225, August 2020b.
- MathWorks. *estimateCameraParameters*. <https://www.mathworks.com/help/vision/ref/estimatecameraparameters.html#bui17ss-1>, July 2020c.
- Department Mechanical Engineering. Residual stress. *Influence*, 2, 2016.

- M. Munaro, E.W. Yan So, S. Tonello, and E. Menegatti. Efficient completeness inspection using real-time 3d color reconstruction with a dual-laser triangulation system. *Advances in Computer Vision and Pattern Recognition*, 48, 2015.
- MVTec. *Solution Guide III-C*. MVTec.
- T.L. Panontin and M. R. Hill. The effect of residual stresses on brittle and ductile fracture initiation predicted by micromechanical models. *International Journal of Fracture*, 82, 1996.
- B. Prime. Cross-sectional mapping of residual stresses by measuring the surface contour after a cut. *Journal of Engineering Materials and Technology*, 123, 2001.
- B. Prime, J. Sebring, J. Edwards, J. Hughes, and J. Webster. Laser surface contouring and spline data-smoothing for residual stress measurement. *Experimental Mechanics*, 44, 2004.
- Qioptiq. *Lens configuration software MachVis*. https://www.qioptiq-shop.com/en/Optics-Software/MachVis-Lens-Configuration/MachVis/Lens-configuration-software-MachVis.html?force_sid=kjffjmesc4pp1fd1gqpnqfsjuildpvaed, July 2020.
- V. Rich-Trummer and P.M.S.T. Castro. The through-the-thickness measurement of residual stress in a thick welded steel compact tension specimen by the contour method. *The Journal of Strain Analysis for Engineering Design*, 46, 2011.
- N.S. Rossini, M. Dassisti, K.Y. Benyounis, and A.G. Olabi. Methods of measuring residual stresses in components. *International Journal of Materials in Engineering Applications*, 35, 2012.
- P. Sargal and F. Mendal. Comparison of ring-core method and hole-drilling method used for determining residual stresses. *American Journal of Mechanical Engineering*, 1, 2013.
- M.R. Shortis, T.A. Clarke, and T. Short. A comparison of some techniques for the subpixel location of discrete target images. *Proceedings*, 2350, 1994.
- P. J. Sousa, P.J.S. Tavares, P.M.G.P. Moreira, and J.M.R.S Tavares. Effect of checkerboard on the accuracy of camera calibration. *Correction of rigid body motion in deformation measurement of rotating objects*, 1, 2018.
- Stresstech. *X-Ray Diffraction*. <https://www.stresstech.com/en/knowledge/non-destructive-testing-methods/x-ray-diffraction/>, April 2020.
- N. Tebedge, G. A. Alpsten, and L. Tall. Residual stress measurement by the sectioning method, presented at sesa spring meeting. *Fritz Laboratory Reports*, 1972.
- E. Tekkaya, A. Erden, and B. Ekmekci. A semi-empirical approach for residual stresses in electric discharge machining (edm). *International Journal of Machine Tools and Manufacture*, 2006.
- R. Thiébaud and J. Lebet. Experimental study of residual stresses in thick steel plates. *Proceedings of the Annual Stability Conference*, 46, 2012.
- M.B. Toparli and M.E. Fitzpatrick. Development and application of the contour method to determine the residual stresses in thin laser-peened aluminium alloy plates. *Experimental Mechanics*, 56, 2016.

- L. Veitch and E. Laprade. Recent developments in the joint strike fighter durability testing. 2, 2, 2.
- R. Weber, J. Rötting, Scholtes.B., and M. Bacher-Höchst. Load stresses and residual stresses in the tooth interior. *International Gear Conference 2014*, 2014.
- P.J. Withers and H.K.D.H. Bahdeshia. Residual stress part 1: Measurement techniques. *Materials Science and Technology*, 17, 2001.
- S Yu, R. Zhu, L. Yu, and W. Ai. Effect of checkerboard on the accuracy of camera calibration. *Advances in Multimedia Information Processing*, 1, 2018.
- Z. Zhang. A flexible new technique for camera calibration. *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, 22, 2000.
- Z. Zhang, G. Medioni, and S. Bing Kang. Camera calibration. *Emerging Topics in Computer Vision*, 2, 2003.

Appendix A

Calibration Algorithm

```
function varargout = LaserScan(varargin)
% LASERSCAN MATLAB code for LaserScan.fig
%     LASERSCAN, by itself, creates a new LASERSCAN or raises the existing
%     singleton*.
%
%     H = LASERSCAN returns the handle to a new LASERSCAN or the handle to
%     the existing singleton*.
%
%     LASERSCAN('CALLBACK',hObject,eventData,handles,...) calls the local
%     function named CALLBACK in LASERSCAN.M with the given input arguments.
%
%     LASERSCAN('Property','Value',...) creates a new LASERSCAN or raises the
%     existing singleton*. Starting from the left, property value pairs are
%     applied to the GUI before LaserScan_OpeningFcn gets called. An
%     unrecognized property name or invalid value makes property application
%     stop. All inputs are passed to LaserScan_OpeningFcn via varargin.
%
%     *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
%     instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help LaserScan

% Last Modified by GUIDE v2.5 15-Mar-2019 18:11:07

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn', @LaserScan_OpeningFcn, ...
                  'gui_OutputFcn',  @LaserScan_OutputFcn, ...
                  'gui_LayoutFcn',  [] , ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
```

```

    gui_State.gui_Callback = str2func(varargin{1});
end

if nargin
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before LaserScan is made visible.
function LaserScan_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
% varargin   command line arguments to LaserScan (see VARARGIN)
global imageslaser useMax
imageslaser=ones(1,3);
% Choose default command line output for LaserScan
handles.output = hObject;
useMax = 0;
set(handles.checkbox2,'Value',0);
% Update handles structure
guidata(hObject, handles);
set(handles.rightEdge,'Value',1);

% UIWAIT makes LaserScan wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = LaserScan_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on button press in Calib_camera_button.
function Calib_camera_button_Callback(hObject, eventdata, handles)
% Load Calibration images and calibrate camera
global bs imagemcalib ImagePoints ip WorldBoardCoord cameraParams calibFiles numfiles
cd1=cd;
```

```

if (~isempty(calibPath))
    try
        cd(calibPath);
    catch
    end
end
[calibFiles, pathname, ~] = uigetfile({'*.png'; '*.tiff'; '*.jpg'}, 'Pick a file', '
numfiles=size(calibFiles,2);
%imagemcalib=cell(numfiles,1);
calibPath = pathname;
if (numfiles == 1 && ~ischar(calibFiles(1)))
    return;
end

prompt = {'Enter Square size in mm:', 'Number of Rows calibration pattern:', 'Numb
dlg_title = 'Calibration Specifications';
num_lines = 1;
def = {'2.5', '8', '9'};
answerStr = inputdlg(prompt,dlg_title,num_lines,def);
if (isempty(answerStr))
    cd(cd1);
    return;
end
answer = str2double(answerStr);
squaresize=answer(1);
boardsize=[answer(2),answer(3)];

%cd(pathname)
%h=waitbar(0,'A importar imagens!');
%for i=1:numfiles
%    imagemcalib{i}=imread(filename{i});
%    waitbar(i/numfiles)
%end
%delete(h)
cd(cd1)

h=waitbar(0,'Calculating!');

dist = zeros(6,1);
for i=1:numfiles
    imagem = imread([calibPath calibFiles{i}]);
    [ImagePoints,bsi]=detectCheckerboardPoints(imagem);
    if bsi(1)==boardsize(1) && bsi(2)==boardsize(2)
        bs = bsi;
        ip(:, :, i)=ImagePoints;
        temp = im2double(imagem);
        thresh=max(temp(:))*0.8;
        temp(temp<thresh) = 0;
    end
end

```

```

    for kk1=1:3 % 1 is right, 2 is middle, 3 is left
        for kk2=1:6
            x = ImagePoints(kk1*2*6-kk2+1,1);
            y = ImagePoints(kk1*2*6-kk2+1,2);
            roundY = round(y);
            [~, refX] = find(temp(roundY,:) > 0,1,'first');
            if (~isempty(refX))
                dist(kk2) = sqrt((refX - x)^2);
            else
                [~, refX] = find(temp(roundY-5:roundY+5,:) > 0,1,'first');
                if (~isempty(refX))
                    dist(kk2) = sqrt((refX - x)^2);
                else
                    dist(kk2) = size(temp,2);
                end
            end
        end

        %figure;imshow(temp);hold on;plot(x,y,'r+');hold off;
    end
    pointDistanceToLaser(kk1,i) = sum(dist);
end
else
    pointDistanceToLaser(:,i) = ones(3,1)*size(imagem,2) + 1;
end
waitbar(i/(numfiles*2+4));
end

[wp] = generateCheckerboardPoints(boardsize,squaresize);
waitbar((numfiles+2)/(numfiles*2+4));
[cameraParams, imagesUsed] = estimateCameraParameters(ip,wp,...
    'EstimateSkew', true, 'EstimateTangentialDistortion', false, ...
    'NumRadialDistortionCoefficients', 2, 'WorldUnits', 'millimeters', ...
    'InitialIntrinsicMatrix', [], 'InitialRadialDistortion', []);

waitbar((numfiles+3)/(numfiles*2+4));
I=cameraParams.IntrinsicMatrix;
R=cameraParams.RotationMatrices;
T=cameraParams.TranslationVectors;
x = wp(:,1);
y = wp(:,2);

numfilesvalid=size(T,1);
WorldBoardCoord=cell(numfiles,1);
waitbar((numfiles+4)/(numfiles*2+4));
for i=1:numfiles
    try
        WorldBoardCoord{i}=bsxfun(@plus,R(:, :, i)'*[x,y,zeros(length(x),1)]',T(i,:))';
    catch

```

```

        end
        waitbar((numfiles+4+i)/(numfiles*2+4));
    end
    delete(h);
    disp(imagesUsed)
    j=sum(imagesUsed);
    disp(j)
    tracker=cell2mat(WorldBoardCoord(j,1)');
    disp(min(tracker(3,:)))
    disp(max(tracker(3,:)))
    viewExtrinsics_Callback([], [], handles);
    %msgbox('Completed!');

% --- Executes on button press in pickLaserPlane.
function pickLaserPlane_Callback(hObject, eventdata, handles)
% hObject      handle to pickLaserPlane (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
PickLaserImages

% --- Executes on button press in viewExtrinsics.
function viewExtrinsics_Callback(hObject, eventdata, handles)
%Plots extrinsic parameters of the camera
global cameraParams WorldBoardCoord
teste=cell2mat(WorldBoardCoord');
%Visualização parametros extrinsecos dos planos
axes(handles.axes1);
view([-55.5 18]);
scatter3(teste(1,:),teste(3,:),teste(2,:));
hold on;
showExtrinsics(cameraParams);
xlabel('X (mm)','FontSize',20,'FontName','Times');
ylabel('Z (mm)','FontSize',20,'FontName','Times');
zlabel('Y (mm)','FontSize',20,'FontName','Times');
title('Extrinsic Parameters Visualization','FontSize',20,'FontName','Times');
rotate3d on

% --- Executes on button press in calibLaser.
function calibLaser_Callback(hObject, eventdata, handles)

% hObject      handle to calibLaser (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

```

```

global coeffPlaneLaser cameraParams planeAngle ip bs calibPath calibFiles

% Images accepted by the calibration algorithm
imUsed = find(reshape(sum(sum(ip,1),2)>0,1,[])); %line vector
Xlaser = [];
for i = 1:size(imUsed,2)
    Ic = imread([calibPath calibFiles{imUsed(i)}]);
    [laser_y ,laser_x] = find(imbinarize(mat2gray(Ic), 0.8));
    laser = [laser_x'; laser_y'];
    laser = removePointsOutsideBoard(laser, ip(:, :, imUsed(i))', bs);
    Ri = cameraParams.RotationMatrices(:, :, i)';
    Ti = cameraParams.TranslationVectors(i, :)' ;
    Xlaser_i = planeIntersection(laser, [Ri Ti], cameraParams.IntrinsicMatrix');
    Xlaser = [Xlaser Xlaser_i];
end

axes(handles.axes1);
cla
%NOTE THAT YY AND ZZ AXES ARE SWAPPED
showExtrinsics(cameraParams);view(3);hold on;
%LaserPlanePoints=[il1(7:12,1),il1(7:12,3),il1(7:12,2);il2(19:24,1),il2(19:24,3),il2(19:24,2)];
%plot3(LaserPlanePoints(:,1),LaserPlanePoints(:,2),LaserPlanePoints(:,3),'ro')
xlabel('X (mm)','FontSize',20,'FontName','Times');
ylabel('Z (mm)','FontSize',20,'FontName','Times');
zlabel('Y (mm)','FontSize',20,'FontName','Times');
title('Extrinsic Parameters Visualization','FontSize',20,'FontName','Times');
rotate3d on

% Plane equation -> aX + bY + cZ = 1
coeff = Xlaser'\ones(size(Xlaser,2),1);
coeff(4) = -1;

coeffPlaneLaser=coeff/norm(coeff(1:3),2);

%Desenhar plano laser
%Para y=0 e z=0 x=-d/a
xplaneorigin=-coeffPlaneLaser(4)/coeffPlaneLaser(1);
zmax=max(Xlaser(3,:));
ymin=min(Xlaser(2,:));
ymax=max(Xlaser(2,:));
xplaneendbottom=(-coeffPlaneLaser(4)-coeffPlaneLaser(3)*zmax-coeffPlaneLaser(2)/ymin)/coeffPlaneLaser(1);
xplaneendtop=(-coeffPlaneLaser(4)-coeffPlaneLaser(3)*zmax-coeffPlaneLaser(2)/ymax)/coeffPlaneLaser(1);

plane=[xplaneorigin, ymin 0;xplaneorigin ymax 0;xplaneendtop ymax zmax;xplaneendbottom ymax 0];
axes(handles.axes1);
hold on;
fill3(plane(:,1),plane(:,3),plane(:,2),plane(:,3)*0);
planeAngle=atan2d(plane(3,2)-plane(4,2),plane(3,1)-plane(4,1));

```

```
disp(xplaneorigin)
```

```
% --- Executes on button press in run.
```

```
function run_Callback(hObject, eventdata, handles)
```

```
% hObject    handle to run (see GCBO)
```

```
% eventdata  reserved - to be defined in a future version of MATLAB
```

```
% handles    structure with handles and user data (see GUIDATA)
```

```
global I coeffPlaneLaser freq pointcloud planeAngle lastRunParams useRightEdge u
```

```
%Load images and specify the test parameters
```

```
prompt = {'Images frequency [frames/second]:','Enter Threshold (min=0; max=255)'};
```

```
dlg_title = 'Test Specifications';
```

```
num_lines = 1;
```

```
if (~isempty(lastRunParams))
```

```
    def = {num2str(lastRunParams.freq), num2str(lastRunParams.threshold)};
```

```
else
```

```
    lastRunParams.freq = 20;
```

```
    lastRunParams.threshold = 100;
```

```
    lastRunParams.lastFolder = '.';
```

```
    def = {'20','100'};
```

```
end
```

```
answer = str2double(inputdlg(prompt,dlg_title,num_lines,def));
```

```
if (isempty(answer))
```

```
    return;
```

```
end
```

```
freq=answer(1);
```

```
threshold=answer(2);
```

```
lastRunParams.freq = freq;
```

```
lastRunParams.threshold = threshold;
```

```
cd1=cd;
```

```
if (~isempty(lastRunParams.lastFolder))
```

```
    try
```

```
        cd(lastRunParams.lastFolder);
```

```
    catch
```

```
    end
```

```
end
```

```
[filename, pathname, ~] = uigetfile({'*.png'; '*.tiff'; '*.jpg'}, 'Pick a file', 'Mu
```

```
if (isempty(filename))
```

```
    cd(cd1);
```

```
    return;
```

```
end
```

```
cd(cd1);
```

```
lastRunParams.lastFolder = pathname;
```

```
[~, parentFolder] = fileparts(pathname(1:end-1));
```

```
numfiles=size(filename,2);
```

```
%imagem=cell(numfiles,1);

fx=I(1,1);
fy=I(2,2);
Pc=cell(numfiles,1);
```

Check laser plane

```
if (abs(planeAngle - 90) > 5)
    warning('Laser line deviates from the vertical by more than 5 degrees!');
    answer = questdlg('Laser line deviates from the vertical by more than 5 degrees!',
        'Laser line deviates from the vertical by more than 5 degrees!',
        'Yes', 'No', 'No');
    if (strcmp(answer, 'No'))
        return;
    end
end

h=waitbar(0,'A processar imagens!');
velx=str2double(get(handles.text6,'String'));
% vely=str2double(get(handles.text7,'String'));
vely=0;
% velz=str2double(get(handles.text8,'String'));
velz=0;

for i=1:numfiles

    aux=imread([pathname filename{i}]);
    aux=rgb2gray(aux);
    aux=double(aux>threshold); aux(aux==0)=NaN;
    imagem=double(aux);
    if (i == 1)
        umax=size(imagem,2);
        vmax=size(imagem,1);
        %U V são as matrizes gerais de coordenadas da imagem
        u=(-umax/2:1:umax/2);
        v=(-vmax/2:1:vmax/2);
        u(u==0)=[];
        v(v==0)=[];
        [u,v]=meshgrid(u,v);
    end
end
```

SINGLE PIXELS

```
if (useMax)
    for ky=1:vmax
        [val, ~] = max(imagem(ky,:));
        if (~isnan(val))
            imagem(ky,imagem(ky,:)~=val) = NaN;
            midPt = median(find(~isnan(imagem(ky,:)))));
            imagem(ky,~ismember(1:end,midPt)) = NaN;
        end
    end
end
```

```

        end
    end
    % for ky=1:vmax
    % [val, idx] = max(imagem(ky,:));
    % if (~isnan(val))
    %     imagem(ky,~ismember(1:end,idx)) = NaN;
    % end
    % end
    % Consider a single line of pixels
elseif (useRightEdge)
    lastIdx = [0 0];
    for ky=1:vmax
        notNan = ~isnan(imagem(ky,:));
        if (any(notNan))
            idx = find(notNan,1,'last');
            if (ky > 1) % Operation is implemented in the previous row
                if (lastIdx(1) > idx)
                    imagem(ky-1,1:idx-1) = NaN;
                elseif (lastIdx(2) < lastIdx(1) && lastIdx(2) > 0 )
                    imagem(ky-1,1:lastIdx(2)-1) = NaN;
                else
                    imagem(ky-1,1:lastIdx(1)-1) = NaN;
                end
            end
            %imagem(ky,1:idx-1) = NaN;
            lastIdx = [idx lastIdx(1)];
        elseif lastIdx(1) > 0
            imagem(ky-1,1:lastIdx(1)-1) = NaN;
            lastIdx = [-1 lastIdx(1)];
        end
    end
end

ui=u;
vi=v;
ui(isnan(imagem))=NaN;
vi(isnan(imagem))=NaN;
u0=ui(:);u0(isnan(u0))=[];
v0=vi(:);v0(isnan(v0))=[];

%4
%Equação do plano = ax+by+cz+d=0
a=coeffPlaneLaser(1);
b=coeffPlaneLaser(2);
c=coeffPlaneLaser(3);
d=coeffPlaneLaser(4);

Z=-d./(a*u0/fx+b*v0/fy+c);%Substituição na equação do plano dos valores de x
X=Z.*u0/fx+velx/freq*(i-1);

```

```

Y=Z.*v0/fy+vely/freq*(i-1);
Z=Z+velz/freq*(i-1);

%5
try
    Pc{i}=[X Y Z];
catch
    Pc{i}=[];
end
waitbar(i/numfiles)

end

```

Export point cloud

```

waitbar(0,h,'Exporting point cloud');
pointcloud=cell2mat(Pc);
waitbar(0.1)

% Choose the filename
if strncmpi(parentFolder,'snap',4)
    % If the folder is called snap, do not include it
    % If the folder has a suffix, include that suffix
    nameSuffix = parentFolder(5:end);
else
    % Else, use the folder name as a suffix with an underscore
    nameSuffix = ['_' parentFolder];
end

dlmwrite(['pointcloud' nameSuffix '.txt'],pointcloud);
waitbar(0.9)
assignin ('base','pointcloud_xyz',pointcloud)
waitbar(1)
delete(h);
cd(cd1);
msgbox('Export Complete');

% --- Executes on button press in calibrateSpeed.

function calibrateSpeed_Callback(hObject, eventdata, handles)

% hObject      handle to calibrateSpeed (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
%Load speed calibration images
global velx vely velz xx yy time zz calibPath speedFiles calibFiles

```

```
waitbar(0,h,'Calculating checkerboards...');

%Define image frequency and calibration pattern
ipvel=zeros((answervel(2)-1)*(answervel(3)-1),2,round(numfiles/fps)+k)*0;
totalSteps = round(numfiles/fps)+k;
issued=[];
for i=1:totalSteps
    try
        [ImagePoints,bs,imagesUsed]=detectCheckerboardPoints(velcalib{i});
        issued=[issued; imagesUsed];
        if bs(1)==boardsize(1) && bs(2)==boardsize(2)
            ipvel(:,:,i)=ImagePoints;
        end
    catch
        errordlg('Please make sure all images contain the full calibration pattern');
        break;
    end
end
waitbar(i/totalSteps);
```

end

```
j=0;
%counts calib_camera images used to detect image points
for i=1:k
    if issued(round(numfiles/fps)+i)==1
        j=j+1;
    end
end
end
```

```
waitbar(0,h,'Estimating camera parameters...');
[wp] = generateCheckerboardPoints(boardsize,squaresize);
[cameraParams, imagesUsed, estimationErrors] = estimateCameraParameters(ipvel,wp, ...
    'EstimateSkew', true, 'EstimateTangentialDistortion', false, ...
    'NumRadialDistortionCoefficients', 2, 'WorldUnits', 'millimeters', ...
    'InitialIntrinsicMatrix', [], 'InitialRadialDistortion', []);
```

```
m=0;
%counts calib_camera images used to estimate camera parameters
for i=length(imagesUsed)-j+1:length(imagesUsed)
    if imagesUsed(i)==1
        m=m+1;
    end
end
end
```

```
waitbar(0.2)
```

```
Rvel=cameraParams.RotationMatrices;
Tvel=cameraParams.TranslationVectors;
xvel = wp(:,1);
yvel = wp(:,2);
```

```
WorldBoardCoord_vel=cell(size(Rvel,3)-m,1);
for i=1:size(WorldBoardCoord_vel,1)
    try
        WorldBoardCoord_vel{i}=bsxfun(@plus,Rvel(:,:,i)'+[xvel,yvel,zeros(length(xvel))
    catch
    end
end
end
```

```
waitbar(0.5,h,'Calculating speed...');
%Define speed in the three directions
```

```

ww=cell2mat(WorldBoardCoord_vel);
xx=mean(ww(1:3:end,:),2);
yy=mean(ww(2:3:end,:),2);
zz=mean(ww(3:3:end,:),2);
time=squeeze(double(ipvel(1,1,:)==0));time(time==1)=NaN;time=time+[0:size(time,1)

aux=polyfit(time,xx,1);
velx=aux(1);

aux=polyfit(time,yy,1);
vely=aux(1);

aux=polyfit(time,zz,1);
velz=aux(1);

waitbar(0.6,h,'Calculating speed...');

set(handles.text6,'String',num2str(-velx));
set(handles.text7,'String',num2str(-vely));
set(handles.text8,'String',num2str(-velz));

figure1 = figure('Color',...
    [0.9333333333333333 0.9333333333333333 0.9333333333333333]);
axes1 = axes('Parent',figure1,...
    'Position',[0.13 0.111893939393939 0.213405797101449 0.815],...
    'FontName','Times');
set(figure1, 'Position', [100,300,1500,400])
box(axes1,'on');
hold(axes1,'all');
plot(time,xx,'Parent',axes1,'Marker','o','LineWidth',2);
xlabel('Time [s]','FontSize',10,'FontName','Times');
ylabel('X pattern displacement [mm]','FontSize',10,'FontName','Times');
subplot1 = subplot(1,3,2,'Parent',figure1,'FontName','Times');
box(subplot1,'on');
hold(subplot1,'all');
plot(time,yy,'Parent',subplot1,'Marker','o','LineWidth',2,'Color',[1 0 0]);
xlabel('Time [s]','FontSize',10,'FontName','Times');
ylabel('Y pattern displacement [mm]','FontSize',10,'FontName','Times');
subplot2 = subplot(1,3,3,'Parent',figure1,'FontName','Times');
box(subplot2,'on');
hold(subplot2,'all');
plot(time,zz,'Parent',subplot2,'Marker','o','LineWidth',2,'Color',[0 1 0]);
xlabel('Time [s]','FontSize',10,'FontName','Times');
ylabel('Z Pattern Displacement [mm]','FontSize',10,'FontName','Times');

waitbar(1,h,'Calculating speed...');
delete(h)

```

```

%clearvars -except velx vely velz xx yy zz time

% --- Executes on button press in pick_threshold_button.

function pick_threshold_button_Callback(hObject, eventdata, handles)
% hObject      handle to pick_threshold_button (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
choose_threshold

% --- Executes on button press in rightEdge.
function rightEdge_Callback(hObject, eventdata, handles)
% hObject      handle to rightEdge (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of rightEdge
global useRightEdge
useRightEdge = get(hObject,'Value');

% --- Executes on button press in checkbox2.
function checkbox2_Callback(hObject, eventdata, handles)
% hObject      handle to checkbox2 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of checkbox2
global useRightEdge useMax
useMax = get(hObject,'Value');
if (useMax)
    useRightEdge = false;
    set(handles.rightEdge,'Value',0);
end

```

Appendix B

Welded Plates specimen

```
% program which takes the CMM (CATIM) output and interpolates it over the
% plates for plotting the deformation
% 10/11/2011 last modified 11/11/2011

% prepare the MatLab workspace
clear all, close all, clc

% Interpolation type
type_inter=2; %1 linear interpolation
               %2 nearest interpolation
               %3 cubic interpolation

% Downsizing Point Grid inputs
% -----
type_remov=0; %Which way removes points
%0 does not remove any point
%1 removes interlock points
%2 removes entire lines
%3 removes entire columns
%4 removes entire columns and lines

removconst=2; %by how many times the mesh is downsized
               %2 removes 1/2
% -----
% Smoothing parameters
knots={41,1}; % number of knots in X and Y directions
x=-150:2:150;
y=0:0.5:9;
[X,Y]=meshgrid(x,y); % interpolation grid

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% -----
% parts of the code below were taken from:
% ML_PhD_VRT_CTsteel_contour_1_v01_simplifiedCodeForShowing.M
```

```

%
% Created V. Richter-Trummer 13/07/2010
% -----

part=3; % part 1 is the geoemtry creation for the FEM mesh,
        % part 2 writes the path for abaqus
        % part 3 reads the abaqus output

% input data
file{1}='raw_data/input_matlab_placaA';
file{2}='raw_data/input_matlab_placaB';

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

for k=1:2
    clear Data_raw Data
    % raw CMM measurement data

    Data_left=readmatrix(file{k},'Range', 'A3:C977');
    Data_right=readmatrix(file{k},'Range', 'E3:G977');

    Data(1:length(Data_left(:,1)),1)=Data_left(:,1);
    Data(1:length(Data_left(:,1)),2)=Data_left(:,2);
    Data(1:length(Data_left(:,1)),3)=Data_left(:,3);
    Data(length(Data_left(:,1))+1:length(Data_left(:,1))*2,1)=Data_right(:,1);
    Data(length(Data_left(:,1))+1:length(Data_left(:,1))*2,2)=Data_right(:,2);
    Data(length(Data_left(:,1))+1:length(Data_left(:,1))*2,3)=Data_right(:,3);

    Data(find(Data(:,3)<-1),:)=[];

    if k==1
        data_plate_A=Data;
    elseif k==2
        data_plate_B=Data;
    end
end

tol=0.2;
contador_X=0;

for x_sort=-149:2:149
    contador_X=contador_X+1;
    organizerA=find(data_plate_A(:,1)>x_sort-tol & ...
        data_plate_A(:,1)<x_sort+tol);
    [a,b]=sort(data_plate_A(organizerA,2),'descend');
    measured_A_matrix_y(:,contador_X)=a;
    organizerB=find(data_plate_B(:,1)>x_sort-tol & ...

```

```

        data_plate_B(:,1)<x_sort+tol);
        [a,b]=sort(data_plate_B(organizerB,2),'descend');
        measured_B_matrix_y(:,contador_X)=a;
    end

    count=0;
    for nn=1:length(measured_A_matrix_y(1,:))/2
        for mm=1:length(measured_A_matrix_y(:,1))
            count=count+1;
            measured_A_matrix_x(mm,nn)=data_plate_A(count,1);
            measured_A_matrix_z(mm,nn)=data_plate_A(count,3);
        end
    end
end

count=length(data_plate_A(:,1));
for nn=length(measured_A_matrix_y(1,:))/2+1:length(measured_A_matrix_y(1,:))
    for mm=1:length(measured_A_matrix_y(:,1))
        count=count-1;
        measured_A_matrix_x(mm,nn)=data_plate_A(count,1);
        measured_A_matrix_z(mm,nn)=data_plate_A(count,3);
    end
end

count=0;
for nn=1:length(measured_B_matrix_y(1,:))/2
    for mm=1:length(measured_B_matrix_y(:,1))
        count=count+1;
        measured_B_matrix_x(mm,nn)=data_plate_B(count,1);
        measured_B_matrix_z(mm,nn)=data_plate_B(count,3);
    end
end
end
count=length(data_plate_B(:,1));
for nn=length(measured_B_matrix_y(1,:))/2+1:length(measured_B_matrix_y(1,:))
    for mm=1:length(measured_B_matrix_y(:,1))
        count=count-1;
        measured_B_matrix_x(mm,nn)=data_plate_B(count,1);
        measured_B_matrix_z(mm,nn)=data_plate_B(count,3);
    end
end

% importing nodes
allnodes=csvread('FEM/Nodess.inp',1,0);
nodeset_=importdata('FEM/surface_nodess.inp');
nodeset=unique(nodeset_.data);
ins=[];
for n=1:length(nodeset)
    if isnan(nodeset(n))

```

```

        ins=[ins n];
    end
end
nodeset(ins)=[]; %Remove NaNs
for n=1:length(nodeset)
    nodes(n,:) = allnodes(find(allnodes(:,1)==nodeset(n)),:);
end

tol=0.2;
contador_X=0;
for x_sort=-149:2:-7
    contador_X=contador_X+1;
    aux_2=find(nodes(:,2)>x_sort-tol & ...
        nodes(:,2)<x_sort+tol);
    [a,b]=sort(nodes(aux_2,3),'descend');
    nodes_list=nodes(aux_2(b),1);
    node_ID_A(:,contador_X)=nodes_list;
end

contador_solda=0;
for x_sort=-5:2:-5
    contador_solda=contador_solda+1;
    aux_2=find(nodes(:,2)>x_sort-tol & ...
        nodes(:,2)<x_sort+tol);
    [a,b]=sort(nodes(aux_2,3),'descend');
    nodes_list=nodes(aux_2(b),1);
    node_ID_middle(:,contador_solda)=nodes_list;
end

contador_x=0;
for x_sort=7:2:149
    contador_x=contador_x+1;
    aux_2=find(nodes(:,2)>x_sort-tol & ...
        nodes(:,2)<x_sort+tol);
    [a,b]=sort(nodes(aux_2,3),'descend');
    nodes_list=nodes(aux_2(b),1);
    node_ID_B(:,contador_x)=nodes_list;
end
node_ID=[node_ID_A node_ID_middle node_ID_B];
for nn=1:size(node_ID,2)
    for mm=1:size(node_ID,1)
        c=find(nodes(:,1)==node_ID(mm,nn));
        matrix_x(mm,nn)=nodes(c,2);
        matrix_y(mm,nn)=nodes(c,3);
    end
end
end

```

```

% Downsizing grid
if type_remov == 1
    if removconst==4
        for p1 = 1:length(measured_A_matrix_x(1,:))
            if mod(p1,2) == 0
                for p2 = 1:length(measured_A_matrix_x(:,1))
                    if mod(p2,removconst) == 0
                        measured_A_matrix_x(p2,p1) = NaN;
                        measured_A_matrix_y(p2,p1) = NaN;
                        measured_A_matrix_z(p2,p1) = NaN;
                        measured_B_matrix_x(p2,p1) = NaN;
                        measured_B_matrix_y(p2,p1) = NaN;
                        measured_B_matrix_z(p2,p1) = NaN;
                    end
                end
            end
        end
    end
    if removconst==2
        for p1 = 1:length(measured_A_matrix_x(1,:))
            if mod(p1,2) ~= 0
                for p2 = 1:length(measured_A_matrix_x(:,1))
                    if mod(p2,removconst) == 0
                        measured_A_matrix_x(p2,p1) = NaN;
                        measured_A_matrix_y(p2,p1) = NaN;
                        measured_A_matrix_z(p2,p1) = NaN;
                        measured_B_matrix_x(p2,p1) = NaN;
                        measured_B_matrix_y(p2,p1) = NaN;
                        measured_B_matrix_z(p2,p1) = NaN;
                    end
                end
            end
        end
        if mod(p1,2) == 0
            for p2 = 1:length(measured_A_matrix_x(:,1))
                if mod(p2,removconst) ~= 0
                    measured_A_matrix_x(p2,p1) = NaN;
                    measured_A_matrix_y(p2,p1) = NaN;
                    measured_A_matrix_z(p2,p1) = NaN;
                    measured_B_matrix_x(p2,p1) = NaN;
                    measured_B_matrix_y(p2,p1) = NaN;
                    measured_B_matrix_z(p2,p1) = NaN;
                end
            end
        end
    end
end
elseif type_remov == 2

```

```

if removconst ~=1
    for p1 = 1:length(measured_A_matrix_x(:,1))
        if mod(p1,removconst) == 0
            measured_A_matrix_x(p1,:) = NaN;
            measured_A_matrix_y(p1,:) = NaN;
            measured_A_matrix_z(p1,:) = NaN;
            measured_B_matrix_x(p1,:) = NaN;
            measured_B_matrix_y(p1,:) = NaN;
            measured_B_matrix_z(p1,:) = NaN;
        end
    end
end
indX=find(isnan(measured_A_matrix_x(:,1)));
indY=find(isnan(measured_A_matrix_x(1,:)));
elseif type_remov == 3
    if removconst ~=1
        for p2 = 1:length(measured_A_matrix_x(1,:))
            if mod(p2,removconst) == 0
                measured_A_matrix_x(:,p2) = NaN;
                measured_A_matrix_y(:,p2) = NaN;
                measured_A_matrix_z(:,p2) = NaN;
                measured_B_matrix_x(:,p2) = NaN;
                measured_B_matrix_y(:,p2) = NaN;
                measured_B_matrix_z(:,p2) = NaN;
            end
        end
    end
    indX=find(isnan(measured_A_matrix_x(:,1)));
    indY=find(isnan(measured_A_matrix_x(1,:)));
elseif type_remov == 4
    for p1 = 1:length(measured_A_matrix_x(1,:))
        if mod(p1,removconst) == 0
            measured_A_matrix_x(:,p1) = NaN;
            measured_A_matrix_y(:,p1) = NaN;
            measured_A_matrix_z(:,p1) = NaN;
            measured_B_matrix_x(:,p1) = NaN;
            measured_B_matrix_y(:,p1) = NaN;
            measured_B_matrix_z(:,p1) = NaN;
        end
    end
    for p2 = 1:length(measured_A_matrix_x(:,1))
        if mod(p2,removconst) == 0
            measured_A_matrix_x(p2,:) = NaN;
            measured_A_matrix_y(p2,:) = NaN;
            measured_A_matrix_z(p2,:) = NaN;
            measured_B_matrix_x(p2,:) = NaN;
            measured_B_matrix_y(p2,:) = NaN;
            measured_B_matrix_z(p2,:) = NaN;
        end
    end
end

```

```

        end
    end
    indX=find(isnan(measured_A_matrix_x(:,1)));
    indY=find(isnan(measured_A_matrix_x(1,:)));
end

% Rebuilding original Data matrix
count=0;
for nn=1:length(measured_A_matrix_y(1,:))
    for mm=1:length(measured_A_matrix_y(:,1))
        count=count+1;
        new_data_plate_A(count,:)=[measured_A_matrix_x(mm,nn) ...
            measured_A_matrix_y(mm,nn) measured_A_matrix_z(mm,nn)];
        new_data_plate_B(count,:)=[measured_B_matrix_x(mm,nn) ...
            measured_B_matrix_y(mm,nn) measured_B_matrix_z(mm,nn)];
    end
end
ind=find(isnan(new_data_plate_A(:,1)));
new_data_plate_A(ind,:)=[];
ind2=find(isnan(new_data_plate_B(:,1)));
new_data_plate_B(ind2,:)=[];

% Alignment of both halves
% -----
% calculate centroid
x0 = mean(new_data_plate_A)';

% translate the points around the centroid
XYZ_trans = [(new_data_plate_A(:, 1) - x0(1)) (new_data_plate_A(:, 2) - x0(2)) ...
[U, S, V] = svd(XYZ_trans, 0);

% find the smallest singular value in S and extract the
% corresponding right singular vector from V
[s, ii] = min(diag(S));
a = V(:, ii); % Direction cosines of the normal to the best-fit
% plane.

for n=1:length(new_data_plate_A)
    plane_plate_A(n)=x0(3)-a(1)/a(3)*(new_data_plate_A(n,1)-x0(1))-a(2)/a(3)*...
        (new_data_plate_A(n,2)-x0(2));
end
% Alignment
new_data_plate_A(:,3)=new_data_plate_A(:,3) - plane_plate_A';

% calculate centroid
x0 = mean(new_data_plate_B)';

% translate the points around the centroid

```

```

XYZ_trans = [(new_data_plate_B(:, 1) - x0(1)) (new_data_plate_B(:, 2) - x0(2)) (new_data_plate_B(:, 3) - x0(3))];

[U, S, V] = svd(XYZ_trans, 0);

% find the smallest singular value in S and extract the
% corresponding right singular vector from V
[s, ii] = min(diag(S));
a = V(:, ii); % Direction cosines of the normal to the best-fit
% plane.

for n=1:length(new_data_plate_B)
    plane_plate_B(n)=x0(3)-a(1)/a(3)*(new_data_plate_B(n,1)-x0(1))-a(2)/a(3)*...
        (new_data_plate_B(n,2)-x0(2));
end
% Alignment
new_data_plate_B(:,3)=new_data_plate_B(:,3) - plane_plate_B';
% -----

figure(119), hold on
plot3(new_data_plate_A(:,1),new_data_plate_A(:,2),new_data_plate_A(:,3),'r.')
plot3(new_data_plate_B(:,1),new_data_plate_B(:,2),new_data_plate_B(:,3),'b.')

if type_inter==1
    Z_plate_A=griddata(new_data_plate_A(:,1),new_data_plate_A(:,2) ...
        ,new_data_plate_A(:,3),X,Y,'linear');
    Z_plate_B=griddata(-new_data_plate_B(:,1),new_data_plate_B(:,2) ...
        ,new_data_plate_B(:,3),X,Y,'linear');

elseif type_inter==2
    Z_plate_A=griddata(new_data_plate_A(:,1),new_data_plate_A(:,2) ...
        ,new_data_plate_A(:,3),X,Y,'nearest');
    Z_plate_B=griddata(-new_data_plate_B(:,1),new_data_plate_B(:,2) ...
        ,new_data_plate_B(:,3),X,Y,'nearest');

elseif type_inter==3
    Z_plate_A=griddata(new_data_plate_A(:,1),new_data_plate_A(:,2) ...
        ,new_data_plate_A(:,3),X,Y,'cubic');
    Z_plate_B=griddata(-new_data_plate_B(:,1),new_data_plate_B(:,2) ...
        ,new_data_plate_B(:,3),X,Y,'cubic');

elseif type_inter==4
    Z_plate_A=griddata(new_data_plate_A(:,1),new_data_plate_A(:,2) ...
        ,new_data_plate_A(:,3),X,Y,'v4');
    Z_plate_B=griddata(-new_data_plate_B(:,1),new_data_plate_B(:,2) ...
        ,new_data_plate_B(:,3),X,Y,'v4');
end

% calculates the average of both specimen halves

```

```

average_surface=(Z_plate_A+Z_plate_B)./2;
mean_average_surface_lines=mean(average_surface,1);

i=0;
for m=1:length(X(:,1))
    for n=1:length(X(1,:))
        i=i+1;
        new_Data(i,:) = [X(m,n) Y(m,n) average_surface(m,n)];
    end
end
XYZ=[new_Data(:,1),new_Data(:,2),new_Data(:,3)];

% calculate centroid
x0 = mean(XYZ)';

% translate the points around the centroid
XYZ_trans = [(XYZ(:, 1) - x0(1)) (XYZ(:, 2) - x0(2)) ...
              (XYZ(:, 3) - x0(3))];

[U, S, V] = svd(XYZ_trans, 0);

% find the smallest singular value in S and extract the
% corresponding right singular vector from V
[s, ii] = min(diag(S));
a = V(:, ii); % Direction cosines of the normal to the best-fit
               % plane.

for n=1:length(new_Data)
    average_plane(n)=x0(3)-a(1)/a(3)*(new_Data(n,1)-x0(1))-a(2)/a(3)*...
        (new_Data(n,2)-x0(2));
end

figure(103),hold on
title('Average Interpolated Surface with middle plane')
plot3(new_Data(:,1),new_Data(:,2),new_Data(:,3),'b.')
plot3(new_Data(:,1),new_Data(:,2),average_plane,'r.')
view([-29 40]);

figure(102),hold on
title('Middle planes before and after alignment')
plot3(new_Data(:,1),new_Data(:,2),average_plane,'b.')
view([-29 40]);

new_Data(:,3)=new_Data(:,3) - average_plane';

XYZ=[new_Data(:,1),new_Data(:,2),new_Data(:,3)];

% Testing again for allginment

```

```

% calculate centroid
x0 = mean(XYZ)';

% translate the points around the centroid
XYZ_trans = [(XYZ(:, 1) - x0(1)) (XYZ(:, 2) - x0(2)) ...
              (XYZ(:, 3) - x0(3))];

[U, S, V] = svd(XYZ_trans, 0);

% find the smallest singular value in S and extract the
% corresponding right singular vector from V
[s, ii] = min(diag(S));
a = V(:, ii); % Direction cosines of the normal to the best-fit
               % plane.

for n=1:length(new_Data)
    average_plane(n)=x0(3)-a(1)/a(3)*(new_Data(n,1)-x0(1))-a(2)/a(3)*...
        (new_Data(n,2)-x0(2));
end

i=0;
for m=1:length(X(:,1))
    for n=1:length(X(1,:))
        i=i+1;
        X(m,n)= new_Data(i,1);
        Y(m,n)= new_Data(i,2);
        average_surface(m,n)= new_Data(i,3);
    end
end
end
% % % % % %

figure(99),hold on
title('Both Interpolated Surfaces')
plot3(X,Y,Z_plate_A,'b.')
plot3(X,Y,Z_plate_B,'r.')
view([-29 40]);

figure(119),hold on
title('Average Interpolated Surface and both measured plates')
plot3(X,Y,average_surface,'k.')
set(gca,'FontSize',16)
set(gcf, 'Position', get(0,'Screensize')); % Maximize figure
xlabel('specimen width [mm]','Rotation',0)
ylabel('specimen thickness [mm]','Rotation',-35)
zlabel('surface relaxation [mm]','Rotation',90)
view([-12 20]);
saveas(gcf,'analyzed_data/Measured_plates_and_average','pdf')

```

```

figure(120),hold on
title('Average Interpolated Surface')
plot3(X,Y,average_surface,'k.')
view([-29 40]);

figure(121),hold on
title('Average Interpolated Surface')
plot3(X,Y,average_surface,'k.')
view([-29 40]);

figure(101),hold on
title('Average Interpolated Surface')
surf(X,Y,average_surface)
view([10 52]);

figure(102),hold on
title('Difference between before and after allignment of Average Surface')
plot3(new_Data(:,1),new_Data(:,2),average_plane,'r.')
view([-50 20]);
saveas(gcf,'analyzed_data/Before_and_after_allignment_of_average_surface','pdf')

if part ==1
% smooths the gridded data
k=3; % quadratic spline
surf_smooth_av=spap2(knots,[k k],[x,y],average_surface');
for nn=1:size(matrix_x,2)
    for mm=1:size(matrix_x,1)
        smooth_surface(mm,nn)=fnval(surf_smooth_av,...
            {matrix_x(mm,nn),matrix_y(mm,nn)});
    end
end
for nn=1:length(nodes)
    smooth_result(nn)=fnval(surf_smooth_av,{nodes(nn,2),...
        nodes(nn,3)});
    smooth_result_to_write(nn)=smooth_result(nn)+nodes(nn,4);
end

smooth_result=smooth_result-mean(smooth_result);

interpolated_smooth_surface=griddata(matrix_x,matrix_y ...
    ,smooth_surface,X,Y,'nearest');

% Goodness of fit
average_surface=inpaint_nans(average_surface,0);
sum_smooth=0;
sum_origin=0;
for nn=1:size(X,1)
    for mm=1:size(X,2)

```

```

        sum_smooth=sum_smooth+(interpolated_smooth_surface(nn,mm)-...
            average_surface(nn,mm))^2;
        sum_origin=sum_origin+(interpolated_smooth_surface(nn,mm)-...
            nanmean(nanmean(smooth_surface)))^2;

    end
end
GOF=1-sum_smooth/sum_origin;

figure(120), hold on
set(gca,'FontSize',16)
title('Average Smooth Surface and Interpolation Grid')
set(gcf, 'Position', get(0,'Screensize')); % Maximize figure
surf(matrix_x,matrix_y,smooth_surface)
xlabel('specimen width [mm]','Rotation',-25)
ylabel('specimen thickness [mm]','Rotation',15)
zlabel('surface relaxation [mm]')
view([48 37])
% colorbar('location','NorthOutside','FontSize',14)
saveas(gcf,'analyzed_data/Average_Smooth_Surface_and_Interpolation_Grid_3D','pdf')

figure(121), hold on
set(gca,'FontSize',16)
title('Average Smooth Surface and Interpolation Grid')
set(gcf, 'Position', get(0,'Screensize')); % Maximize figure
surf(matrix_x,matrix_y,smooth_surface)
xlabel('specimen width [mm]','Rotation',0)
ylabel('specimen thickness [mm]','Rotation',90)
zlabel('surface relaxation [mm]')
view([-0.5 90])
colorbar('location','EastOutside','FontSize',14)
saveas(gcf,'analyzed_data/Average_Smooth_Surface_and_Interpolation_Grid','pdf')

disp(strcat('The goodness of fit for this surface is ',...
    num2str(GOF*100),'%'))

% Writes the Abaqus input file with the obtained deformation
fid1 = fopen('FEM/1/displacements.inp', 'w');
fprintf(fid1, '*BOUNDARY, TYPE=DISPLACEMENT, OP=MOD\n');
for nn=1:length(smooth_result)
    fprintf(fid1, strcat('\t ','Assembly.Part-1-1.',num2str(nodes(nn,1)),'\t',...
        num2str(3),'\t',\t',num2str(smooth_result(nn)),'\n')); %alterei esta linha
end
fclose(fid1);

```

```
% Writes the Abaqus input file non deformed
fidl = fopen('FEM/1/originaldisplacement.inp', 'w');
fprintf(fidl, '*BOUNDARY, TYPE=DISPLACEMENT, OP=MOD\n');
for nn=1:length(smooth_result)
    fprintf(fidl, strcat('\t ', num2str(nodes(nn,1)), ', \t', ...
num2str(2), ', \t, \t', num2str(nodes(nn,4)), ' \n'));
end
fclose(fidl);
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
if part == 2
    % writes a path with all the nodes from the surface. Afterwards, the
    % stresses on this nodes will be retrieved
    fidl = fopen('FEM/pathB.txt', 'w');
    for j=1:length(nodes(:,1))
        fprintf(fidl, strcat(num2str(nodes(j,1)), ', ');
    end
    fclose(fidl);
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
if part == 3
    % import stress from the created abaqus report and plots them
    DELIMITER = ' ';
    HEADERLINES = 2;
    newData1 = importdata('FEM/abaqus.rpt', DELIMITER, ...
        HEADERLINES);
    stress33=-newData1.data(:,2);
    stress33_qq=-newData1.data(:, :);

    % Ploting
    figure(3)
    plot(nodes(:,2),nodes(:,3),'k.')

    aux_nodes=csvread('FEM/Nodess.inp',1,0);
    %FEMnodes=aux_nodes(find(aux_nodes(:,4)==190),:);
    tol=0.2;
    contador_X=0;
    for x_sort=-149:2:-7
        contador_X=contador_X+1;
        aux_2=find(nodes(:,2)>x_sort-tol & ...
            nodes(:,2)<x_sort+tol);
        [a,b]=sort(nodes(aux_2,3),'descend');
        nodes_list=nodes(aux_2(b),1);
```

```

        node_ID_A(:,contador_X)=nodes_list;
    end

    contador_X=0;
    for x_sort=-5:2:5
        contador_X=contador_X+1;
        aux_2=find(nodes(:,2)>x_sort-tol & ...
            nodes(:,2)<x_sort+tol);
        [a,b]=sort(nodes(aux_2,3),'descend');
        nodes_list=nodes(aux_2(b),1);
        node_ID_middle(:,contador_X)=nodes_list;
    end

    contador_X=0;
    for x_sort=7:2:149
        contador_X=contador_X+1;
        aux_2=find(nodes(:,2)>x_sort-tol & ...
            nodes(:,2)<x_sort+tol);
        [a,b]=sort(nodes(aux_2,3),'descend');
        nodes_list=nodes(aux_2(b),1);
        node_ID_B(:,contador_X)=nodes_list;
    end
    node_ID=[node_ID_A node_ID_middle node_ID_B];
    for nn=1:size(node_ID,2)
        for mm=1:size(node_ID,1)
            c=find(nodes(:,1)==node_ID(mm,nn));
            matrix_stress(mm,nn)=stress33(c);
            matrix_x(mm,nn)=nodes(c,2);
            matrix_y(mm,nn)=nodes(c,3);
        end
    end

    % Ploting
    figure(33)
    set(gca,'FontSize',16)
    surf(matrix_x,matrix_y,matrix_stress)
    xlabel('specimen width [mm]','Rotation',0)
    ylabel('specimen thickness [mm]','Rotation',0)
    zlabel('Normal stress [MPa]')
    colorbar('location','NorthOutside','FontSize',14)
    set(gcf, 'Position', get(0,'Screensize')); % Maximize figure
    saveas(gcf,'analyzed_data/Stress_Map','jpg')

    figure(34)
    set(gca,'FontSize',16)
    surf(matrix_x,matrix_y,matrix_stress)
    xlabel('specimen width [mm]','Rotation',0)

```

```

ylabel('specimen thickness [mm]','Rotation',90)
xlabel('Normal stress [MPa]')
view([-0.5 90])
colorbar('location','EastOutside','FontSize',14)
set(gcf, 'Position', get(0,'Screensize')); % Maximize figure
saveas(gcf,'analyzed_data/Stress_Map_Plane','jpg')

figure(35)
set(gca,'FontSize',16)
surf(matrix_x,matrix_y,matrix_stress,'LineStyle','none')
xlabel('specimen width [mm]','Rotation',0)
ylabel('specimen thickness [mm]','Rotation',90)
xlabel('Normal stress [MPa]')

view([-0.5 90])
colorbar('location','EastOutside','FontSize',14)
set(gcf, 'Position', get(0,'Screensize')); % Maximize figure
saveas(gcf,'analyzed_data/Stress_Map_Plane_no_lines','jpg')

save('stress.mat','matrix_x','matrix_y','matrix_stress')

tol=0.25;
y_sort=2.75;
    aux_3=find(nodes(:,3)>y_sort-tol & ...
        nodes(:,3)<y_sort+tol);
    [a,b]=sort(nodes(aux_3,3),'descend');
    nodes_list2=nodes(aux_3(b),1);
    node_sens(:,1)=nodes_list2;

for i=1:length(node_sens)
    aux_4=find(nodes(:,1)==node_sens(i,1));
    for j=1:length(aux_4)
        if nodes(aux_4(j),2)<-7 || nodes(aux_4(j),2)>7
            sens(i,1)=nodes(aux_4(j),2);
            sens(i,2)=stress33(aux_4(j),1);
        end
    end
end
end

mid_1=[503 4004 3994 3984 3974 3964 3954 287]';

for i=1:length(mid_1)

    aux_5=find(nodes(:,1)==mid_1(i));

```

```
mid_2(i,1)=nodes(aux_5,2);  
mid_2(i,2)=stress33(aux_5,1);  
end  
sens=[sens; mid_2];
```

```
end
```

Appendix C

Correction of the rigid rotations

```
close all, clear all, clc

file{1}='pointcloudA';
file{2}='cmmA';

cloud=readmatrix(file{1},'Range', 'A3:C40000');
bmark=readmatrix(file{2},'Range', 'A3:C9896'); %CMM cloud to use as benchmark

%Remove NaNs
ins=[];
for n=1:length(cloud(:,1))
    if isnan(cloud(n,1))
        ins=[ins n];
    end
end
cloud(ins,:)=[];

% %Remove NaNs
% ins2=[];
% for n=1:length(bmark(:,1))
%     if isnan(bmark(n,1))
%         ins2=[ins2 n];
%     end
% end
% bmark(ins2,:)=[];

%reference plane at z=0
ref=cloud;
ref(:,3)=0;
ref=ref';
```

```

cloud=cloud';

[ direction, rotationCentre, angle, R_1 ] = calculateAxis(ref, cloud, 2);
mRotationStruct.normalVector = direction;
mRotationStruct.rotationCentre = rotationCentre;

corrected = rotatePtsAroundAxis( cloud, mRotationStruct, -angle )';

```

aligns the CMM datapoints with a virtual horizontal plane

parallel to the best fit plane through the centroid of the deformed surface

```

XYZ=[bmark(:,1),bmark(:,2),bmark(:,3)];

% calculate centroid
x0 = mean(XYZ)';

% translate the points around the centroid
XYZ_trans = [(XYZ(:, 1) - x0(1)) (XYZ(:, 2) - x0(2)) ...
              (XYZ(:, 3) - x0(3))];

[U, S, V] = svd(XYZ_trans, 0);

% find the smallest singular value in S and extract the
% corresponding right singular vector from V
[s, ii] = min(diag(S));
a = V(:, ii); % Direction cosines of the normal to the best-fit
               % plane.

for n=1:length(bmark)
    plane(n)=x0(3)-a(1)/a(3)*(bmark(n,1)-x0(1))-a(2)/a(3)*...
              (bmark(n,2)-x0(2));
end

bmark(:,3)=bmark(:,3)-plane';

% xlswrite('correctedA.xlsx',corrected)

```

Plot

```

figure(4)
plot3(corrected(:,1),corrected(:,2),corrected(:,3),'r.')
grid on
xlabel('X [mm]');
ylabel('Y [mm]');
zlabel('Z [mm]');
legend('Scanned points')

```

```

figure(5), hold on
plot3(bmark(:,1),bmark(:,2),bmark(:,3),'b.')
plot3(corrected(:,1),corrected(:,2),corrected(:,3),'r.')
grid on
xlabel('X [mm]');
ylabel('Y [mm]');
zlabel('Z [mm]');
legend('CMM points','Scanned points')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   From "Correction of rigid body motion in deformation measurement   %
%   of rotating objects" (Sousa et al., 2018)                           %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [ rotatedPts ] = rotatePtsAroundAxis( originalPts, mRotationStruct, ang

%UNTITLED Summary of this function goes here
%   Detailed explanation goes here

```

Calculation of the Rotation Matrix

```
rotationMatrix = vrrotvec2mat([mRotationStruct.normalVector' angle*pi/180]);
```

Calculation of the Rotated Points

```

tiledRotationCentre = repmat(mRotationStruct.rotationCentre,1,size(originalPts,2)
rotatedPts = rotationMatrix*(originalPts - tiledRotationCentre) + tiledRotationC

end
function [ direction, rotCentre, angle, R ] = calculateAxis( originalPoints, rot

%calculateAxis LSQ fit to pure rotation around an arbitrary centre point
%   Calculates the axis, angle and centre of rotation using a modification
%   of the Least Squares approach by (Veldpaus et al., 1988) and (Rose and
%   Richards, 2012).
%
%   Inputs: original points and rotated points
%   Outputs: Axis of rotation, Centre of Rotation, Angle, Rotation Matrix

```

Verifies the size of the input vectors

```

a = originalPoints;
p = rotatedPoints;
if (any(size(a) ~= size(p)) || size(a,1) ~= 3)
    error('The size of the matrices is not compatible. Both originalPoints and r
end

```

Algorithm(Veldpaus et al., 1988), (Rose and Richards, 2012).

```

avg_p = mean(p,2);
avg_a = mean(a,2);
M = 1/size(a,2)*a*p'-avg_a*avg_p';
[V, D2] = eig(M' * M);
D = sqrt(D2);
m = M*V;
teste = cross(V(:,2),V(:,3));
if (all(teste./V(:,1) > 0 | teste == 0))
    R = [cross(m(:,2),m(:,3))/(D(2,2)*D(3,3)) , m(:,2)/D(2,2) , m(:,3)/D(3,3)] * V';
elseif (all(teste./V(:,1) < 0 | teste == 0))
    R = [-cross(m(:,2),m(:,3))/(D(2,2)*D(3,3)) , m(:,2)/D(2,2) , m(:,3)/D(3,3)] * V';
else
    error('Caught Exception: V2 x V3 != +/- V1');
end
R = R';
%v = avg_p - R * avg_a;
%p2 = R*a + repmat(v,1,size(a,2));
%difs = p2-p

% Calculation of the rotation axis
rotVect = vrrotmat2vec(R);
direction = rotVect(1:3)';
angle = rotVect(4)*180/pi;
% [Rv, Rd] = eig(R);
% realIdx = 0;
% for i=1:size(Rv,1)
%     if (isreal(Rd(i,i)))
%         realIdx = i;
%         break;
%     end
% end
% angle = acosd((trace(R)-1)/2); % Saída em graus
% direction = Rv(:,realIdx)/(2*sin(angle));
% direction = direction / norm(direction);

```

Calculation of the Rotation Center (Algoritmo próprio)

```

I = eye(3);
% DEDUCTION:
% avg_p=R*(avg_a-x0)+x0 => centro de rotação!
% => -x0 = R*avg_a - R*x0 - avg_p
% => (R - I)*x0 = R*avg_a - avg_p
% => x0 = (R - I) \ (R*avg_a - avg_p)
A = (R - I);
b = (R*avg_a - avg_p);
%rotCentre = A\b;

```

```

try
    rotCentre = lsqminnorm(A,b); % Minimum norm solution
catch
    warning('LSQMINNORM not available, falling back to PINV');
    rotCentre = pinv(A)*b; % Generates some higher-error results
end
if nargin == 3
    if zeroIdx==1 || zeroIdx==2 || zeroIdx==3
        if (abs(direction(zeroIdx)) > 0.1)
            rotCentre = rotCentre - direction*rotCentre(zeroIdx)/direction(zeroI
        end
    end
end
end
% As singular => multiple solutions, solution A\b is not correct.
% Implies the use of an alternative, the pseudo-inverse (pinv).
% SOURCE:
% https://www.mathworks.com/help/matlab/ref/mldivide.html
% If the rank of A is less than the number of columns in A, then x = A\b is
% not necessarily the minimum norm solution. The more computationally
% expensive x = pinv(A)*B computes the minimum norm least-squares solution.

% If >=R2017b
%rotCentre = lsqminnorm(A,b);
% SOURCE:
% https://www.mathworks.com/help/matlab/ref/pinv.html#d119e765044
% You can replace most uses of pinv applied to a vector b, as in pinv(A)*b,
% with lsqminnorm(A,b) to get the minimum-norm least-squares solution of a
% system of linear equations. lsqminnorm is generally more efficient than
% pinv, and it also supports sparse matrices.

end

```

Appendix D

CT specimen

```
% program which takes the CMM (CATIM) output and interpolates it over the
% plates for plotting the deformation
% Valentin, 13.7.10

% prepare the MatLab workspace
clear all, close all, clc

part=1; % part 1 is the geoemtry creation for the FEM mesh,
        % part 2 uses the FEM mesh for surface interpolation
        % part 3 writes the path for abaqus
        % part 4 reads the abaqus output

% input data
file{1}='raw_data/FEUP-MedPlaneza-ZonaCorteErosao_1-A';
file{2}='raw_data/FEUP-MedPlaneza-ZonaCorteErosao_1-B';

file_FEM='FEM/femap/FEM_nodes.inp';
fileID={'A','B'};

if part ==1||3||4
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    for k=1:2
        clear Data_raw Data
        % raw CMM measurement data

        Data(:,1)=readmatrix(file{k},'Range', 'A3:A300000');
        Data(:,2)=readmatrix(file{k},'Range', 'B3:B300000');
        Data(:,3)=readmatrix(file{k},'Range', 'C3:C300000');

        %remove NANS
        ins=[];
        for n=1:length(Data(:,1))
            if isnan(Data(n,1))
                ins=[ins n];
            end
        end
    end
end
```

```

        end
    end
    Data(ins,:)=[];

    % finds data outside of the area to be measured and deletes it
    index=find(Data(:,1)>105.4);
    Data(index,:)=[];

    % inverts the first plate in order to be able to take the average of
    % both
    if k==1
        Data(:,2)=-Data(:,2)+32;
    end

    % aligns the obtained datapoints with a virtual horizontal plane
    % parallel to the best fit plane through the centroid of the
    % deformed surface

    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    % -----
    % parts of the code below were taken from:
    % LSPLANE.M   Least-squares plane (orthogonal distance
    %             regression).
    %
    % Version 1.0
    % Last amended   I M Smith 27 May 2002.
    % Created       I M Smith 08 Mar 2002
    % -----
    XYZ=[Data(:,1),Data(:,2),Data(:,3)];

    % calculate centroid
    x0 = mean(XYZ)';

    % translate the points around the centroid
    XYZ_trans = [(XYZ(:, 1) - x0(1)) (XYZ(:, 2) - x0(2)) ...
        (XYZ(:, 3) - x0(3))];

    [U, S, V] = svd(XYZ_trans, 0);

    % find the smallest singular value in S and extract the
    % corresponding right singular vector from V
    [s, ii] = min(diag(S));
    a = V(:, ii); % Direction cosines of the normal to the best-fit
    % plane.

    plane=[];
    for n=1:length(Data)
        plane(n)=x0(3)-a(1)/a(3)*(Data(n,1)-x0(1))-a(2)/a(3)*...

```

```

        (Data(n,2)-x0(2));
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

figure(100+k), hold on
plot3(Data(:,1),Data(:,2),Data(:,3),'b.')
plot3(Data(:,1),Data(:,2),plane,'r.')

figure(10+k), hold on
set(gca,'FontSize',16)
plot3(Data(:,1),Data(:,2),Data(:,3),'k.')
xlabel('specimen width [mm]','Rotation',-25)
ylabel('specimen thickness [mm]','Rotation',15)
zlabel('surface relaxation [mm]')
view(50,25)
axis([0 105 0 32 -0.1 0.1])
saveas(gcf,...
        strcat('analyzed_data/G-PhD-VRT-CTsteelContour_1_surf_',...
        num2str(k)),'pdf')

Data(:,3)=Data(:,3) - plane';

figure(20+k), hold on
plot3(Data(:,1),Data(:,2),Data(:,3),'k.')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% use the mesh created in FEMAP for interpolation of the CMM data
% the mesh is made of linear brick elements
allnodes=csvread('FEM/nodes.inp',1,0);
nodeset_=importdata('FEM/surface_node_set.inp');
nodeset=unique(nodeset_.data); %transfere todos os dados para uma só coluna

%remove NaNs
ins=[];
for h=1:length(nodeset)
    if isnan(nodeset(h))
        ins=[ins h];
    end
end
nodeset(ins)=[];
for n=1:length(nodeset)

```

```

        nodes(n,:) = allnodes(find(allnodes(:,1)==nodeset(n)),:);
    end

    % interpolate the measured data points to a rectangular grid
    x=linspace(0,105.4,265);
    x(266)=109.1;
    y=linspace(0,32,45);
    [X,Y]=meshgrid(x,y);
    meshed_data{k}=griddata(Data(:,1),Data(:,2),Data(:,3),X,Y,...
        'nearest');

end

% calculates the average of both specimen halves
average_surface=(meshed_data{1}+meshed_data{2})./2;

% eliminates some errors
average_surface(find(average_surface(:,:)>0.04))=NaN;

if part ==1
    % smooths the gridded data
    k=3; % quadratic spline
    knots={3,5}; % number of knots in X and Y directions
    surf_smooth_av=spap2(knots,[k k],[x,y],average_surface');

    for nn=1:size(X,2)
        for mm=1:size(X,1)
            smooth_surface(mm,nn)=fnval(surf_smooth_av,...
                {X(mm,nn),Y(mm,nn)});
        end
    end

    for nn=1:length(nodes)
        smooth_result(nn)=fnval(surf_smooth_av,{nodes(nn,2),...
            nodes(nn,4)});
    end

    smooth_result=smooth_result-mean(smooth_result);

    % plotting
    figure(3)
    set(gca,'FontSize',16)
    surf(X,Y,average_surface)
    xlabel('specimen width [mm]','Rotation',-25)
    ylabel('specimen thickness [mm]','Rotation',15)

```

```

zlabel('surface relaxation [mm]')
view(50,25)
axis([0 105 0 32 -0.05 0.05])
saveas(gcf,...
    'analyzed_data/G-PhD-VRT-CTsteelContour_1_surf_av','pdf')

figure(4)
set(gca,'FontSize',16)
surf(X,Y,smooth_surface)
xlabel('specimen width [mm]','Rotation',-25)
ylabel('specimen thickness [mm]','Rotation',15)
zlabel('surface relaxation [mm]')
view(50,25)
axis([0 105 0 32 -0.05 0.05])
saveas(gcf,...
    'analyzed_data/G-PhD-VRT-CTsteelContour_1_surf_smooth','pdf')

% Goodness of fit
average_surface=inpaint_nans(average_surface,0);
sum_smooth=0;
sum_origin=0;
for nn=1:size(X,1)
    for mm=1:size(X,2)

        sum_smooth=sum_smooth+(smooth_surface(nn,mm)-...
            average_surface(nn,mm))^2;
        sum_origin=sum_origin+(average_surface(nn,mm)-...
            nanmean(nanmean(smooth_surface)))^2;

    end
end
GOF=1-sum_smooth/sum_origin;

clc
disp(strcat('The goodness of fit for this surface is ',...
    num2str(GOF*100),'%'))

% Writes the Abaqus input file with the obtained deformation
fid1 = fopen('FEM/1/displacement.inp', 'w');
fprintf(fid1, '*BOUNDARY, TYPE=DISPLACEMENT, OP=MOD\n');
for nn=1:length(smooth_result)
    fprintf(fid1, strcat('\t ','Assembly.Part-1-1.',num2str(nodes(nn,1)),'\t',
        num2str(2),',',\t',\t',num2str(smooth_result(nn)),'\n'));
end
fclose(fid1)

```

```

end

end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

if part == 3
    % writes a path with all the nodes from the surface. Afterwards, the
    % stresses on this nodes will be retrieved
    fid1 = fopen('FEM/pathB.txt', 'w');
    for j=1:length(nodes(:,1))
        fprintf(fid1, strcat(num2str(nodes(j,1)),',', ' '));
    end
    fclose(fid1)
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

if part == 4
    % import stress from the created abaqus report and plots them
    DELIMITER = ' ';
    HEADERLINES = 2;
    newData1 = importdata('FEM/abaqus_dataCT1_cmm.rpt', DELIMITER,...
        HEADERLINES);
    stress22=-newData1.data(:,2);

    % eliminates the extrapolated node values
    indexes=find(nodes(:,2)>106);
    stress22(indexes)=[];
    nodes(indexes,:)=[];

    figure(3)
    plot3(nodes(:,2),nodes(:,4),stress22,'k.')

    % interpolate the obtained data to a rectangular grid
    x=linspace(0,105,265);
    y=linspace(0,32,45);
    [X,Y]=meshgrid(x,y);
    stressRS=griddata(nodes(:,2),nodes(:,4),stress22,X,Y,'nearest');

    figure(31)
    set(gca,'FontSize',16)
    contourf(X,Y,stressRS,20)
    axis equal
    xlabel('specimen width [mm]')
    ylabel('specimen thickness [mm]')
    colorbar('location','NorthOutside','FontSize',16)
    set(gca,'CLim',[-400 400])

```

[illegible]