

QUANTIFICATION OF PROFILE MEASUREMENT DATA

by

Farid Javidpour

A dissertation submitted to the faculty of
The University of North Carolina at Charlotte
in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in
Mechanical Engineering

Charlotte

2019

Approved by:

Dr. Edward Morse

Dr. Gert Goch

Dr. Jimmie Miller

Dr. Konstantinos Falaggis

Dr. Shen-En Chen

ABSTRACT

FARID JAVIDPOUR. Quantification of Profile Measurement Data. (Under the direction of DR. EDWARD MORSE)

Parts with complicated freeform shapes have been attracting more attention in industrial applications in recent decades. In most systems, the functional performance of these parts is directly related to the conformity of their profile with the designed position, form, shape and size of the part surface. Traditional profile measurement methods were primarily developed for parts with simple prismatic shapes; the increased use of freeform profiles has highlighted the need for new rapid measurement methods to validate these profiles with an acceptable level of uncertainty. Metrological uncertainty is a task-specific parameter that quantifies the variability of a measurement, and is influenced by numerous factors. To analyze the effect of these factors, it is necessary to have both a well-quantified metric to describe the profile measurement error, as well as a quantification of errors in the measurement data.

This research has two main components. In the first stage, the common Least Squares (LS) fitting method is extended to two different fitting algorithms identified as Weighted Least Squares (WLS) and Highly Weighted Least Squares (HWLS). These algorithms can accommodate variable density in the point distribution, are developed to analyze the effect of non-uniform point cloud density on the fitting process, which is more likely with newer, optical measuring systems. In these algorithms, the weight of individual point deviations is defined based on distances from surrounding points. These algorithms are implemented in Matlab software, and are applied to simulated theoretical data sets as well as experimental data sets obtained by tactile and optical measurement methods.

In the second stage of this work, a convolution averaging filter is proposed as a method for quantification of profile measurement data. By applying this filter to the point deviation data, the Mean Local Deviation (MLD) of every profile segment is calculated. This quantified local error can be used as a criterion for comparing different profile scans performed with one or multiple measurement methods. The point clouds—fitted by WLS and HWLS algorithms—are filtered and then their MLDs are compared with MLDs of data fitted by LS method. Primary results of this stage show that the proposed filtering method can effectively be used to compare measurement data with very different density, and can also remove the negative effect of noise in data. With effective filtering, the MLD values of a part are almost constant (they are independent of the measurement method).

ACKNOWLEDGEMENTS

I'm always thankful to my mother and sister because they were my first teachers and all I have in my life is from them. Secondly, I thank Holly Martin and tutors of Writing Resources Center at UNC Charlotte for editing my texts and improving the fluency of this dissertation.

I also should thank my excellent advisor Dr. Edward Morse for his unconditional support. He taught me the basics of Metrology so patiently and provided me with good opportunities to advance in this way and deepen my knowledge of this branch of science. I will be his lifelong student and will not forget his dedicated assistance.

I would like to thank all of my committee members as well. It was really my honor to be their student and find the correct way with their guidance.

I offer my appreciation to the Affiliate members of CPM (Center for Precision Metrology) at UNC Charlotte for funding my projects during my study and allowing me to grow scientifically by presenting the results of my research and having productive discussions with them.

Jennifer Chastain and Tracy Beauregard are great ladies making student life easy as two caring "school moms." I thank both of them faithfully from the depth of my heart.

Furthermore, I would like to thank the staff of Hexagon Metrology and Capture 3D (Doug Sutphin, Scott Grumbles, Tori Welty, Jared Williams and David Linford) who measured some of my parts and taught me how to work with their instruments.

Finally I appreciate every assistance from alumni and students of CPM especially Mario Valdez, Yue Peng and Jesse Groover. No achievements could be reached in this research without their outstanding support.

TABLE OF CONTENTS

LIST OF FIGURES	ix
LIST OF TABLES	xv
LIST OF ABBREVIATIONS.....	xvi
CHAPTER 1: INTRODUCTION	1
1.1. Importance of part profile measurement	1
1.2. Profile measurement for quality control	2
1.3. Part model generation for reverse engineering	4
1.4. Available profile measurement methods.....	6
1.5. Importance of knowledge about measurement uncertainty.....	7
1.5.1. Necessity of profile error quantification for uncertainty assessment.....	8
1.6. Research objectives.....	9
1.7. Scope of work	9
1.8. Dissertation outline	10
CHAPTER 2: LITERATURE REVIEW AND STATE OF THE ART	11
2.1. Profile measurement for form error evaluation.....	12
2.1.1. Different fitting methods.....	12
2.1.2. Effect of fitting method on point deviations	18
2.2. Profile measurement for part model generation.....	18
2.2.1. Effect of data processing method on reconstructed surfaces	25

2.3. Mean Local Deviation (MLD) as a method for profile quantification.....	25
CHAPTER 3: POINT DENSITY AND PROFILE FITTING PROCESS.....	27
3.1. Weighted Least Squares (WLS) and Highly Weighted Least Squares (HWLS) fitting algorithms.....	27
3.1.1. Weight definition for two-dimensional profiles.....	30
3.1.2. Weight definition for three-dimensional profiles.....	30
3.2. Theoretical simulations	32
3.2.1. Square profile.....	32
3.2.2. Quarter circle profile.....	34
3.2.3. Saddle profile.....	35
3.3. Experimental measurements	36
3.3.1. Measurement and analysis of 2D profiles.....	36
3.3.2. Measurement and analysis of 3D profiles.....	41
CHAPTER 4: QUANTIFICATION OF PROFILE DATA.....	49
4.1. Quantification of 2D profile data	49
4.2. Quantification of 3D profile data	51
4.2.1. Segmentation of circular surface (wavy profile).....	51
4.2.2. Segmentation of square surface (saddle profile).....	53
CHAPTER 5: RESULTS AND DISCUSSION.....	56
5.1. Results of simulations	56
5.1.1. Square profile.....	56
5.1.2. Quarter circle profile.....	62

5.1.3. Saddle profile	69
5.2. Results of experimental measurements	76
5.2.1. Measurement and analysis of 2D profiles	76
5.2.2. Measurement and analysis of 3D profiles	78
5.2.2.1. Statistical analysis of point deviations	90
CHAPTER 6: CONCLUSIONS AND FUTURE WORKS	95
REFERENCES	98
APPENDIX A: OPTIMIZING FITTING PARAMETERS	105
APPENDIX B: A COLLECTION OF MATLAB CODES	106

LIST OF FIGURES

Figure 1.	Uniform profile tolerance zone for 2D (left) and 3D (right) profiles [15].....	2
Figure 2.	Non-uniform tolerance zone at a segment of profile [5].....	4
Figure 3.	Cycle of product design and development [14]	5
Figure 4.	Examples of point cloud and reconstructed surface; a) human's face; b) inside of a shoe [3], [4]	5
Figure 5.	Different types of part profile measurement sensors [8].....	7
Figure 6.	Error components leading to profile measurement uncertainty [8].....	7
Figure 7.	Detection and removal of outliers from the point cloud of a horse model; a) The original point cloud with outliers b) The detected outliers are shown in red color c) The result of outlier removal procedure [35]	20
Figure 8.	a) segmented point cloud b) initial meshes c) optimized meshes of a scanned rabbit model [35]	20
Figure 9.	Mesh smoothing on the noisy meshed surface of a 8-shaped part [34]	21
Figure 10.	Approximation of a polygonised surface with B-spline surface patches a) segregated surface zones b) fitted B-spline surfaces to the specified zones c) fitted surface patches [36].....	21
Figure 11.	Measured points (P_1, P_2, \dots), midpoints (MP_1, MP_2, \dots) and weights (W_1, W_2, \dots) of deviations (D_1, D_2, \dots) for a simple profile	30
Figure 12.	Steps of Delaunay triangulation of scattered points [64]	31
Figure 13.	Delaunay triangulation with varying point density [65]	31
Figure 14.	Twelve different point layouts used for the analysis of square profile	33
Figure 15.	Two types of point distribution over quarter circle profile; a) Uniform point distribution; b) Non-uniform point distribution	34
Figure 16.	(a) uniform and (b) non-uniform point distribution over saddle profile	36

Figure 17.	Sample parts with (a) outer profile and (b) slot profile for two-dimensional profiles and their general dimensions [66].....	37
Figure 18.	Profile measurement with a Cartesian CMM.....	38
Figure 19.	(a) Optical scan of the outer profile and (b) tactile measurement of slot profile with Romer arm	38
Figure 20.	Measurement of the outer profile with a FARO ION laser tracker.....	39
Figure 21.	Offset in tactile data; (a) general profile shape (b) detail of dashed zone; measured points are shown with dot and calculated contact points are represented with cross [66].....	39
Figure 22.	(a) a point cloud obtained by optical scan (b) a 2D profile created by sectioning and editing the point cloud	40
Figure 23.	Outlier points over the profile data	40
Figure 24.	Aluminum sample parts with freeform top surfaces; (a) wavy profile; (b) saddle profile	41
Figure 25.	Optical scan of wavy profile with Romer arm (linear triangulation sensor).....	42
Figure 26.	Optical scan of freeform surfaces (a) wavy profile, and (b) saddle profile with Blue Light Scanner (fringe projection technique).....	43
Figure 27.	Progress from measured data (blue) to contact points (red) and points matched to the wavy profile surface (orange)	45
Figure 28.	Top view of measured (a) tactile and (b) optical data of wavy profile	46
Figure 29.	Projection of a grid of points onto a polygonised surface to create a uniform sample (a) 3D view, (b) top view.....	48
Figure 30.	(a) measured points matched to the slot profile; (b) graph of point deviations versus profile length [66].....	50
Figure 31.	(a) graph of point deviations of the slot profile; (b) graph of local deviations after applying the convolution filter [66]	51

Figure 32.	Decrease of the circular sectors angle from the center to the outside of the projected area	52
Figure 33.	Deviation of individual measured points from the wavy profile; (a) 3D view; (b) top view	53
Figure 34.	MLD plot of a point cloud measured by the Romer arm; (a) 3D view; (b) top view	53
Figure 35.	Segmentation of the projected square surface.....	54
Figure 36.	Point deviations of a fitted data sample from the saddle profile; (a) 3D view; (b) top view	55
Figure 37.	MLD plot of a data set resulted from measurement of the saddle profile with the Blue Light Scanner; (a) 3D view; (b) top view	55
Figure 38.	Graphs of optimum fitting parameters when the origin is on the corner of the square (a) rotation angle, (b) translation in x axis, (c) translation in y axis.....	58
Figure 39.	Graphs of optimum fitting parameters when the origin is at the center of the square (a) rotation angle, (b) translation in x axis, (c) translation in y axis.....	59
Figure 40.	(a) Minimum and maximum point deviations, (b) range of deviations related to different point layouts (origin at the corner).....	60
Figure 41.	(a) Minimum and maximum point deviations, (b) range of deviations related to different point layouts (origin at the center)	61
Figure 42.	Two test samples with constant point deviations; (a) uniform point distribution; (b)non-uniform point distribution.....	62
Figure 43.	Graphs of point deviations vs. profile length associated to different fitting algorithms; (a) uniform and (b) non-uniform point distribution.....	63
Figure 44.	Relative graph of fitting parameters (a) rotation angle (b) translation along x (c) translation along y belonging to data with uniform structure	66

Figure 45.	Relative graph of fitting parameters (a) rotation angle (b) translation along x (c) translation along y belonging to data with non-uniform structure	67
Figure 46.	Histograms of differences in fitting parameters related to the uniform point distribution (vertical axes show the percentage of probability)	68
Figure 47.	Histograms of differences in fitting parameters related to the non-uniform point distribution (vertical axes show the percentage of probability)	68
Figure 48.	CDF plots of point deviation ranges obtained from various fitting algorithms (a) uniform point distribution; (b) non-uniform point distribution.....	69
Figure 49.	Relative graphs of optimum fitting parameters obtained by applying various algorithms to uniform point samples	71
Figure 50.	Relative graphs of optimum fitting parameters obtained by applying various algorithms to non-uniform point samples	72
Figure 51.	Histograms of differences in fitting parameters related to the uniform point distribution over the saddle profile; (a) difference between WLS and LS; (b) difference between HWLS and LS (vertical axes show the percentage of probability).....	73
Figure 52.	Histograms of differences in fitting parameters related to the non-uniform point distribution over the saddle profile (a) difference between WLS and LS; (b) difference between HWLS and LS (vertical axes show the percentage of probability).....	74
Figure 53.	CDF plots of point deviation ranges obtained from various fitting algorithms (a) uniform point distribution; (b) non-uniform point distribution.....	75
Figure 54.	MLD graphs of the optically measured slot profile (profile 2) fitted by (a) LS algorithm; (b) WLS algorithm	76
Figure 55.	Difference between MLDs originated in WLS and LS fitting algorithms	76
Figure 56.	The lowest range of the difference in MLDs obtained from (a) tactile and (b) optical sensor on the Romer arm instrument	77

Figure 57.	Individual point normal deviations from the wavy profile originated in measurement with the optical sensor on the Romer arm; (left column: top view; right column: isometric view) ..	79
Figure 58.	MLDs on segments of the wavy profile originated in measurement with the optical sensor on the Romer arm; (left column: top view; right column: isometric view) ..	80
Figure 59.	Differences of MLD data originated in measurement with the optical sensor on the Romer arm (left column: front view; right column: isometric view).....	81
Figure 60.	(a) individual point deviations and (b) MLDs obtained from tactile measurement of the wavy profile (left column: front view; right column: isometric view).....	83
Figure 61.	Differences of MLD data originated in measurement with the tactile CMM (left column: front view; right column: isometric view) ..	84
Figure 62.	Individual point deviations resulted from scan of coated and uncoated surface (left column: side view; right column: isometric view).....	85
Figure 63.	Difference between MLDs obtained from measurement of the sprayed wavy surface and the surface with no coating (a) side view; (b) isometric view.....	85
Figure 64.	High deviation from the profile at the location of reference stickers (a) point deviations after fitting by LS algorithm; (b) photo of the scanned part (left column: top view; right column: 3D view) ..	86
Figure 65.	Relative graphs of optimum fitting parameters obtained by applying various algorithms to randomly selected samples with non-uniform structure ..	88
Figure 66.	Relative graphs of optimum fitting parameters obtained by applying various algorithms to samples with uniform structure.....	89
Figure 67.	Histograms of differences in fitting parameters related to randomly selected samples with non-uniform point distribution (a) difference between WLS and LS; (b) difference between HWLS and LS ..	90

Figure 68.	Individual point deviations from the saddle profile obtained from the WLS fitting algorithm; (a) measured with the GOM camera; (b) measured with the Romer arm	91
Figure 69.	Histogram of deviations shown in previous figure; (a) measured with the GOM camera (b) measured with the Romer arm.....	91
Figure 70.	The CDF plot of point deviations derived from two separate scans with the Romer arm	93
Figure 71.	Distribution of point deviations obtained from fitting a sample with uniform structure; (a) histogram; (b) CDF plot	93

LIST OF TABLES

Table 1.	Optimum fitting parameters associated to uniform and non-uniform point distribution	
	63

LIST OF ABBREVIATIONS

AACMM	Articulated Arm Coordinate Measuring Machine
ASME	American Society of Mechanical Engineers
CAD	Computer Aided Design
CCD	Charged Coupled Device
CDF	Cumulative Distribution Function
CMM	Coordinate Measuring Machine
CNC	Computer Numerical Control
CO	Chaos Optimization
CT	Computed Tomography
DOF	Degrees Of Freedom
DZE	Deviation Zone Evaluation
EPF	Exponential Penalty Function
GD&T	Geometric Dimensioning and Tolerancing
HTR	Hybrid Trust Region
HWLS	Highly Weighted Least Squares
HWSS	Highly Weighted Sum of Squares
ICP	Iterative Closest Point
IGA	Immune Genetic Algorithm
ISO	International Organization for Standardization
LM	Levenberg Marquardt
LS	Least Squares
MCC	Minimum Circumscribed Circle
MIC	Maximum Inscribed Circle
MLD	Mean Local Deviation

MLS	Moving Least Squares
MZ	Minimum Zone
NIST	National Institute of Standard and Technology
OCS	Orthogonal Cross Sections
PCA	Principal Component Analysis
PCS	Part Coordinate System
PMP	Point Measurement Planning
PSO	Particle Swarm Optimization
PTFE	Poly Tetra Fluoro Ethylene
PV	Peak to Valley
QC	Quality Control
SGE	Substitute Geometry Estimation
SMR	Spherically Mounted Retroreflector
SQP	Sequential Quadratic Program
SS	Sum of Squares
SVD	Singular Value Decomposition
WLS	Weighted Least Squares
WSS	Weighted Sum of Squares

CHAPTER 1: INTRODUCTION

1.1. Importance of part profile measurement

Manufacturing of mechanical parts with various shapes is a complicated procedure that is usually composed of different processes, such as casting, molding, forming, machining, grinding, polishing, and heat treatment. There are numerous parameters in each process that affect the final workpiece shape, and it is not possible to control them all during manufacture, so the manufactured workpiece is never identical to the desired designed shape. Generally speaking, common universal machines are used for manufacturing of parts with simple prismatic shapes such as plane or cylinder. In this condition, the control of part shape and size is easier than when a complicated freeform shape is desired, and traditional methods may not be applicable to checking new profile shapes. The usage of parts with complicated shapes has grown noticeably in industrial applications, and it highlights the need for new methods of shape and size evaluation. To ensure a satisfactory product functionality, usually a range of shape and size variation is defined in the form of part profile tolerance, which describes a zone around the considered profile segments. Therefore, the first application of part profile measurement is to check the part compatibility with its desired shape and size.

The second application of profile measurement is in reverse engineering when no CAD data are available, and engineers would like to create a digital model from the physical product. This approach is also quite common in maintenance when a part of an outdated system should be replaced with a new one. In addition, reverse engineering is used when conceptual design models need to be directly digitized for rapid prototyping, design evaluation and optimization [1],[2]. In all of these cases, the initial model is created physically, and newer design versions are generated by editing the digital CAD model associated with the primary design.

1.2. Profile measurement for quality control

As stated before, the geometry and dimensions of the manufactured workpiece is never completely identical to the nominal desired size and geometry. This unavoidable variability must be acknowledged at the design level. To do this, design engineers use geometric tolerances to define a range of shape variation that is permitted in the product specifications. The goal of tolerancing is to permit only variation that does not impair the function of the part, while not over-constraining the geometry so that manufacture is prohibitively expensive. One of the tasks of the quality control (QC) department is to check the product variation against the design specification to ensure manufactured workpieces have the desired functionality.

The profile tolerance is used in both ISO and ASME standards to specify a tolerance zone, within which the actual part surface must fit. Depending on the design geometry, this controls a combination of size, orientation and location of features relative to the design (nominal) profile [5], [6]. This zone can be in the form of a volume along the part's outer surface for three-dimensional profiles or an area around two-dimensional part profiles (Figure 1).

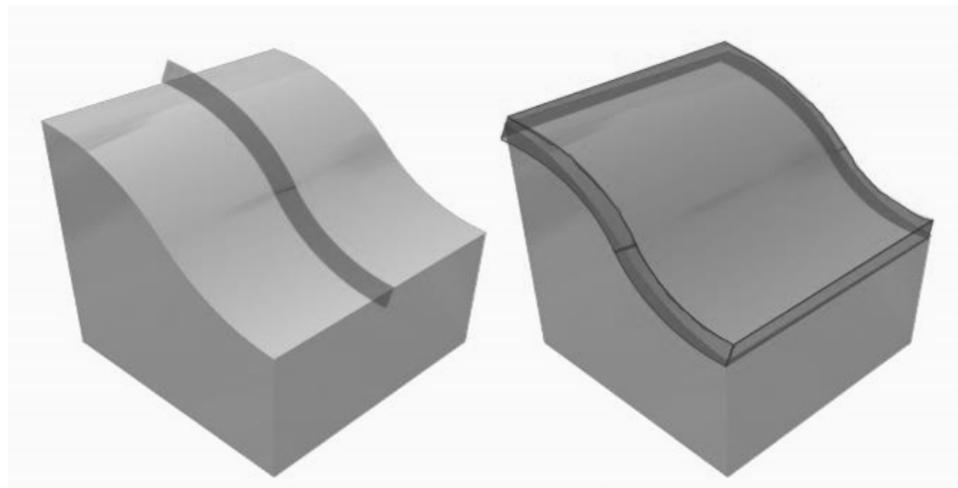


Figure 1. Uniform profile tolerance zone for 2D (left) and 3D (right) profiles [15]

Depending on the designer's intention, the profile tolerance zone can be uniform with constant width (Figure 1) or non-uniform with variable width (Figure 2). It may also be specified at all around the profile or only at a segment of surface or profile perimeter.

In the quality assessment process, to provide a comparison between the manufactured profile and the nominal profile, these steps are executed:

- 1) Collecting points from the surface
- 2) Preparation and filtering of part profile data (2D or 3D)
- 3) Part profile fitting (finding the optimum location and orientation of measured points)
- 4) Calculation of point deviations and comparison with the profile tolerance width

Degrees of freedom have a basic role in the third step, because datums and their associated constraints should be considered before fitting, and they have a significant influence on the result of this process. If any measured surface point does not lie within the tolerance zone, then the part is non-conforming with its specifications, and it should be corrected or recycled. Either of these scenarios would cause a major loss to the manufacturing department because significant effort and energy has been used to manufacture the part. Usually in this situation, manufacturing engineers try to find the main sources of form deviations and correct the existing manufacturing processes. Therefore, it is essential to not only identify the location of non-conforming areas on the workpiece, but also to know the sign and amount of deviation at all profile segments. Utilization of more precise manufacturing methods can be another solution for this problem, but this incurs additional cost.

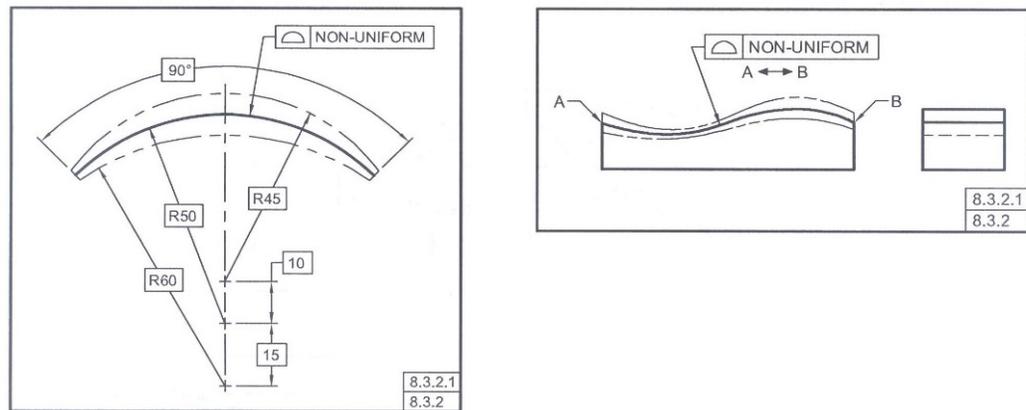


Figure 2. Non-uniform tolerance zone at a segment of profile [5]

1.3. Part model generation for reverse engineering

Figure 3 shows the cycle of product design and development. It usually starts from an initial concept that is the base of design procedure. In the next levels, the concept is brought to reality by prototyping. If the result of evaluations is satisfactory, it may pass other procedures and become a commercial product. The main purpose of reverse engineering is to reverse this chart and reach the design data from the final product. One of the fundamental types of design information is the geometric CAD data, and profile measurement has an important role in providing this type of information. Measurement instruments are able to measure the point coordinates on the workpiece surface, and their output is typically in the form of a cloud of points¹. There are several different techniques for mixing point clouds that originate from multiple sensors or are taken from the same sensor in multiple orientations. Additional operations, such as filtering and polygonization, may be applied to the combined point cloud to construct the CAD model of the physical object. The details of these procedures will be described in the next chapter. Two examples of point clouds and associated reconstructed surfaces are shown in Figure 4. It is realized from these two examples that points may have an organized structure with a constant density or may have a non-uniform distribution over the point cloud.

¹ A set of points in 2D or 3D space

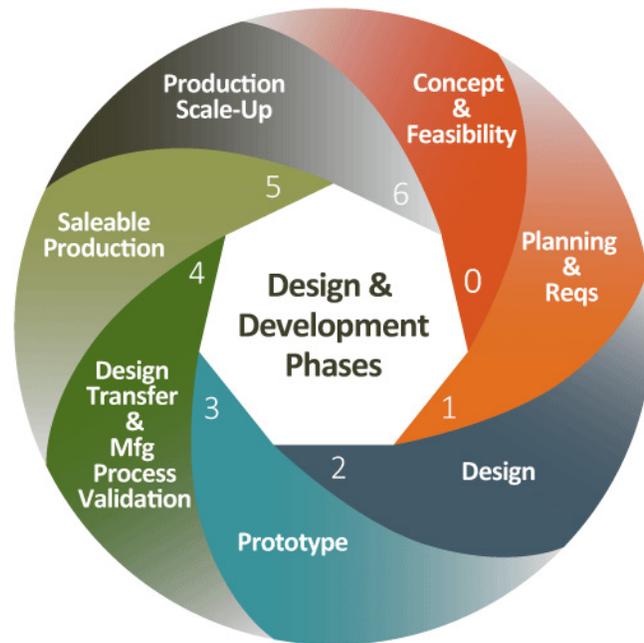


Figure 3. Cycle of product design and development [14]

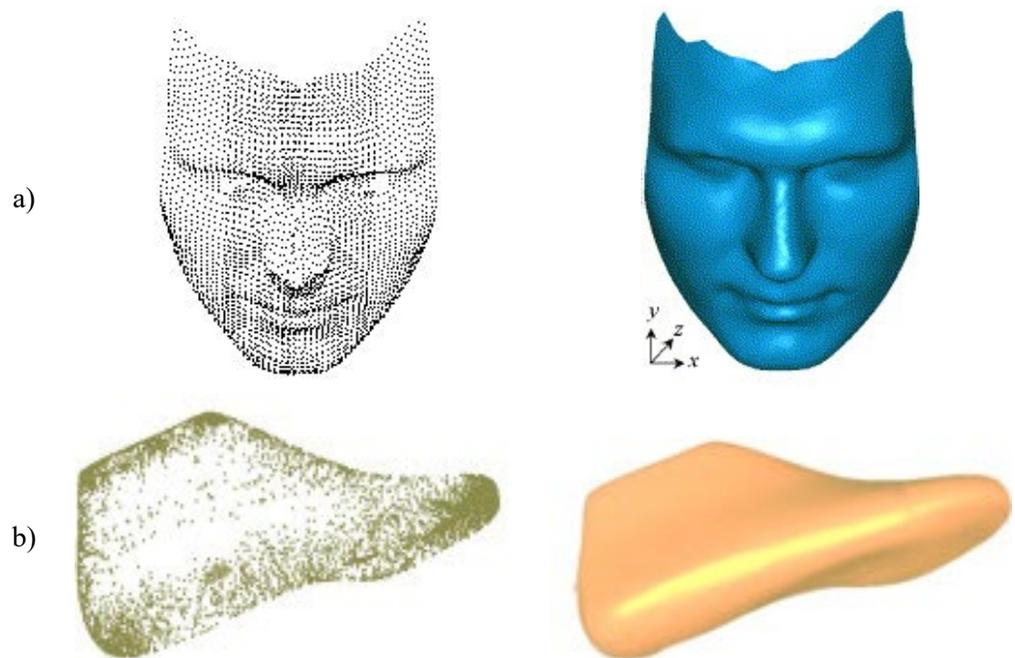


Figure 4. Examples of point cloud and reconstructed surface; a) human's face; b) inside of a shoe [3], [4]

1.4. Available profile measurement methods

There are abundant different methods for measuring points on the workpiece's surface. In each of these methods, the measurement sensor may either apply some energy to the workpiece or may touch it physically. From the first condition, the measurement method can be active or passive [7], while the second condition categorizes the methods into tactile and non-contact groups [8]. In other words, in the passive methods, the collecting tool retrieves information about the environment or other surrounding objects without exerting any energy to them like atomic microscopy or photogrammetry. In contrast, active sensors apply energy to the workpiece to collect its surface data. Laser scanners, ultrasonic sensors and tactile probes are examples of this group of sensors.

Figure 5 shows a detailed categorization of measurement sensors by Christoph and Neumann in [8]. Each of these methods has its own advantages and flaws; that is why there is no universal sensor and many parameters such as environment, measurement range, geometry, surface characteristics, workpiece material properties and more can affect the accuracy and precision of measurement. It should be considered here that measurement is not an ideal procedure and there is a certain level of uncertainty in any kind of measurement. Figure 6 shows the relationships between parameters influencing the results of measurements on a coordinate measuring system; this research focuses on the specific task of profile measurement and evaluation.

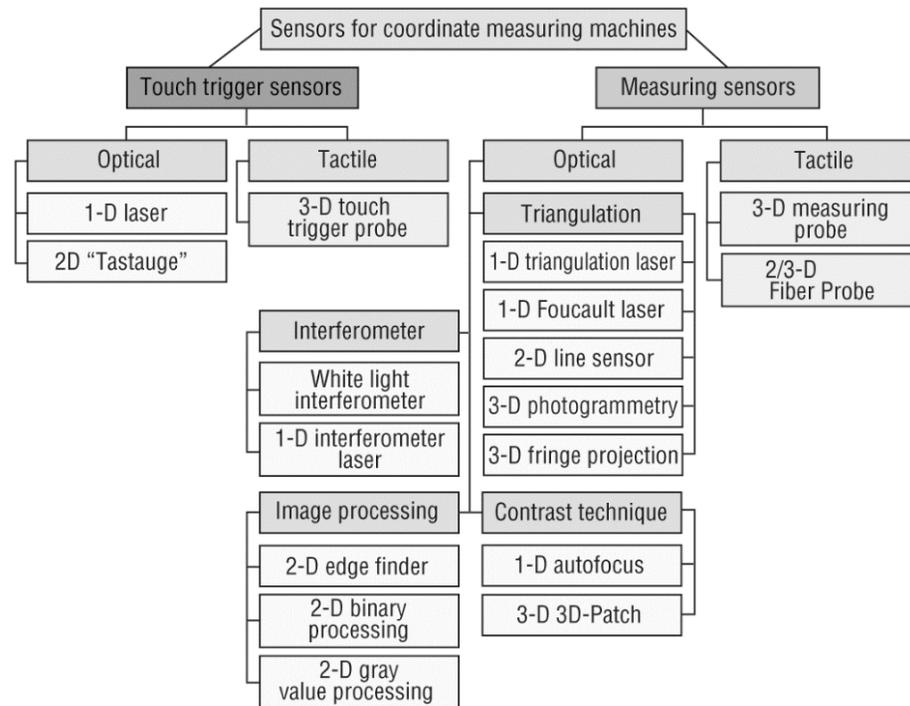


Figure 5. Different types of part profile measurement sensors [8]

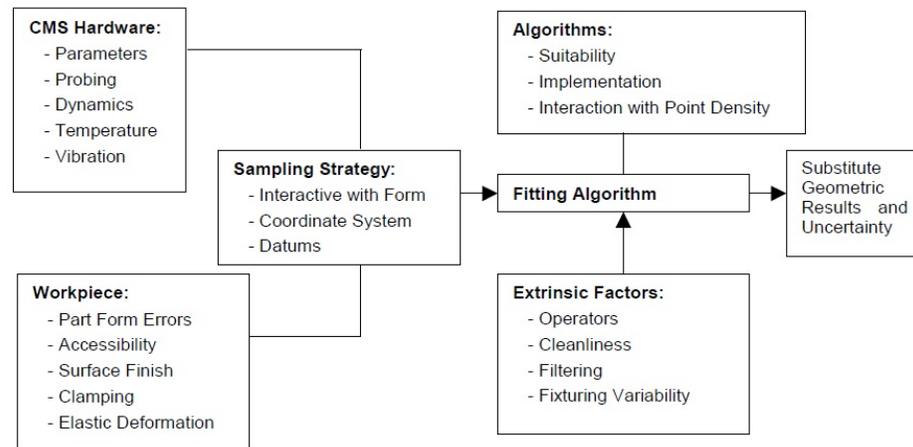


Figure 6. Error components leading to profile measurement uncertainty [8]

1.5. Importance of knowledge about measurement uncertainty

For any part profile measurement there will be a range within which the measured point coordinates will vary due to the measurement method and the conditions under which the measurement is performed. For example, a part profile might be determined to be non-conforming when it is measured with the optical linear triangulation sensor, but the same part might be accepted

when measured with the tactile probe. In this case, it is possible that a conforming part may be rejected due to high task-specific uncertainty of profile measurement with the optical method. The term task-specific here is used to clarify that the optical method may have low uncertainty for some applications, but the particular measuring task of interest is the determination of conformance (or non-conformance) to the part's profile specification. As always, the measurement uncertainty plays a basic role in the part evaluation process and it must be considered when evaluating that process.

Recent research articles that focus on performance characterization or task-specific uncertainty assessment of different measurement methods [9] - [13] describe a fundamental trade-off between tactile and optical methods. Optical scanning methods provide a vast amount of surface data in a very short period of time. However, they are extremely vulnerable to environment conditions and part surface characteristics such as slope, roughness and shininess. That is why tactile probing is widely used for precise applications, although it has much slower scanning speeds. Recent advancements in optical scanning has made metrologists hopeful about the future of this group of contactless methods. Seemingly, if the environment is controlled well enough and the measured data are analyzed in a correct way, the majority of outliers and noisy data can be removed and the measurement uncertainty could be improved noticeably.

1.5.1. Necessity of profile error quantification for uncertainty assessment

The first step toward the profile measurement uncertainty assessment is the estimation of profile measurement error, and for that purpose, the profile measurement data need to be quantified. In other words, quantification of profile measurement data is a prerequisite for the realization of profile measurement uncertainty. The quantification of profile measured data is not only useful for instrument users in improving the profile measurement precision, but also can help the instrument producers design and make more reliable instruments with better performance for part profile measurement. Another application of profile measurement data quantification could be in comparing the performance of two or more measuring instruments. So far, there has not been any

suitable criterion for profile measurement evaluation which can be applicable to scan data of parts with complicated shapes such as sculpted or freeform surfaces. Although– with the increasing use of freeform surfaces in design– the importance of a clear quantification of profile measurement data is becoming better understood in industry. There is still no consensus regarding how a measured profile data set can be quantified and what the main restrictions are of available quantification methods.

1.6. Research objectives

The main objective of this research is to develop a new method for the quantification of part profile measured data, and to use this method to generate a criterion for comparison between various scans with one instrument or a group of instruments equipped with tactile and non-contact optical sensors. A well-designed criterion will make it possible to understand the main differences between tactile and optical data, and to have a view about the variation in fitting parameters as well as deviations at separate profile segments when various fitting methods are applied to profile measured data.

1.7. Scope of work

To reach the objectives mentioned in the previous section, it is planned to simulate points with uniform and non-uniform distribution around different nominal profiles in the 2D and 3D space. The deviation of these points can be constant or randomly selected from a specific range. These simulated points will be fitted to considered profiles by means of various fitting algorithms, and optimized fitting variables derived from utilized algorithms will be compared together as well as the deviation of fitted points. In the next stage, various 2D and 3D part profiles will be measured with various instruments, and the same type of analysis will be performed on experimentally measured data. In this way, it is possible to compare the behavior of various fitting algorithms, but point deviations will not be comparable. To compare the result of various profile measurements, a convolution filter will be applied to profile data, and the Mean Local Deviation (MLD) of profile

segments will be used as a criterion for the comparison between separate measured data sets. Additionally, the selection of samples with an organized structure and the application of various fitting algorithms to them, will highlight the influence of the point density variation on the result of the fitting process.

1.8. Dissertation outline

After a short introduction about profile measurement methods and the necessity of the data quantification in this chapter, the second chapter will include the background and the literature review as well as the state of the art of this research. Various profile data fitting methods will be introduced, and drawbacks and strong points of each method will be discussed. Furthermore, previous research works will be classified, and the role of the profile data quantification will be described. Chapter 3 will cover the explanation of three fitting algorithms (LS, WLS, and HWLS) that will be used for data fitting in the 2D and 3D space. In addition, measured profiles and measuring instruments will be described in that chapter. In the following, it will be described that how simulated data sets are generated and how measured tactile and optical data are analyzed. The quantification process of profile data, which is the main part of this research, will be explained in the fourth chapter. Generally speaking, it will be presented in that chapter that how the measured profile can be divided into segments. For 2D profiles, segments will have equal lengths, and with a 3D profile, segments will have similar areas on the xy plane.

Results of theoretical simulations will be presented in chapter 5, as well as results of experimental profile measurements. Two groups of results will be compared together, and probable reasons for differences in shown data will be discussed. Some applications of the quantification idea will be presented, and it will be discussed that how the quantification process influences measurement data. Following that, a statistical analysis of individual point deviations will be performed on fitted points measured with different instruments. Conclusions and suggestions for future research works will be presented in the sixth chapter.

CHAPTER 2: LITERATURE REVIEW AND STATE OF THE ART

Profile measurement of manufactured parts has been a tedious challenge for humankind for centuries. This action could be executed for a wide range of purposes such as checking the workpiece shape quality, realization of the variation in products, as well as manufacturing processes or copying the shape of an accessible physical item. Primary profile measurement methods were mostly manual and suitable for simple shapes such as flat planes or cylindrical holes. Using Go/No-go gauges is a good example of size evaluation for simple shapes that is still favorable in many industrial applications. However, traditional measurement methods are not applicable for complex profiles. Simple prismatic or axisymmetric shapes are described by simple mathematical equations, and their CAD model is made by sweeping a 2D profile along a line or curve, or revolving around an axis. In contrast, freeform surfaces are defined by a set of control points that are fitted to some equations, so a set of mathematical equations are used for the description of a freeform shape.

By advancement of more modern mechanical systems and manufacturing methods, parts with complicated shapes could attract more attention and were used more in mechanical assemblies. Nowadays it can be seen that sculptured and free-form surfaces are used more often than before in countless systems. Optical lenses, turbine blades, pump propellers, air ducts, biomedical instruments, and artificial organs are proper examples for various implementation of surfaces with complicated shapes. In the majority of these cases, the crucial workpiece feature has an interface with a fluid or light, and the system efficiency has a high dependence to the accuracy of part profile. Therefore, shape complexity and need for more accuracy were the main motivations to move toward newer profile measurement methods such as measurement with Cartesian CMMs with various structures. The traditional CMMs had a tactile stylus that would act as an electronic switch for recording the coordinates of probe center point. However, the tactile measurement was too slow and it was not suitable for large workpieces. New profile measurement methods such as triangulation laser scanning, structured light projection and X-ray Computed Tomography (CT) are

implemented to improve the drawbacks of tactile measurement, but data of these methods are not as reliable as tactile data. Numerous efforts have been done to decrease the uncertainty of these measurement methods, but tactile measurement has still the lowest possible uncertainty for most applications.

2.1. Profile measurement for form error evaluation

As stated before, in this occasion the design information including the nominal profile is available, and the main purpose of profile measurement is to check the conformity of the real part with the nominal profile. In the standards such as ASME Y14.5, the tolerance zone is defined as the area between two limiting curves or the volume between two threshold surfaces. The form deviation of a measured profile is defined in the direction that is perpendicular to the nominal profile, and the distance between limiting parameters which are typically parallel must be minimum. The big issue is that there is no clear description in standards for fitting the measured data to the nominal profile and finding the minimum distance between border features. However, if the orthogonal deviations are defined mathematically, minimization of maximum distance can be found through the concept of coordinate transformation and parallel bodies [16]. Therefore, the choice of fitting method has an extensive influence on the resulted form deviation of measured points.

2.1.1. Different fitting methods

Among several fitting methods that have been developed so far, the Least Squares (LS) and Minimum Zone (MZ) or Chebyshev fitting methods are more common than others. The main reasons for this vast usage of these methods are easy mathematical definition and existence of numerous methods for their optimization. It should be mentioned here that the Chebyshev method is used because it can replicate the requirement of a zone around the profile. The majority of other methods are different extensions of these two methods. They have been developed due to their better performances at special conditions. For example, one-sided fitting is derived from

Chebyshev fitting. This method has found significant application for circular shapes in the form of Maximum Inscribed Circle (MIC) or Minimum Circumscribed Circle (MCC). Another famous application of this method is in fitting planar surfaces for datum definition. The results of this fitting method has satisfactory compatibility with experimental tests.

Research has mostly been conducted in three main categories of computational tasks of coordinate metrology: i. Point Measurement Planning (PMP) ii. Substitute Geometry Estimation (SGE) iii. Deviation Zone Evaluation (DZE) [17].

In PMP, the focus is on the number of measured points and their locations. The number of points is limited and the workpiece geometry must be estimated via these discrete points. Therefore, the objective of PMP is to achieve a more precise representation of measured geometries by using fewer points [17]. Edgeworth and Wilhelm [18] developed a sampling process for interpolating curves between sample points. This sampling strategy is relevant to min-max fitting algorithms and uses surface normal data. It is realized that as the sample size increases, the error surface generated by using cubic splines converges to match the error of the actual surface. Since their algorithm generates sample points concentrated at areas of large error, the total number of required points will be less than a dense uniform pattern. Another interesting research in PMP belongs to Summerhayes et al. [19]. They presented a new pattern for optimum, reasonably sized probing for measurement of internal cylindrical surfaces (holes) under time and economic constraints. In the applied extended zone method, a dense pattern of several holes is measured and then different subsets with various point numbers are selected and fitted to the nominal feature by using a linear combination of Fourier series, Chebyshev polynomials and Principal Component Analysis (PCA). The GD&T¹ parameters derived from the sparse sets are compared to the values determined by the dense sample. Regarding the possible form errors, an optimized sampling pattern was generated. The composite error of this type of sample merges rapidly to the composite error of dense data set

¹ Geometric Dimensioning and Tolerancing

even at low number of points. Their results show significant improvements in the uncertainty of derived dimension with small or no increase in measurement time [19].

Generally speaking, the desired sampling pattern is uniform, and if fitting multiple surface patches is desired, higher point densities are needed at surface areas with more shape complexity. In tactile measurement, it is possible to select the location of target points, but this option is not available in optical scanning methods. However, the point density can be set via exposure time or light intensity, so the point density varies over a point cloud. Another approach is to select specific points from the measured point cloud to obtain a uniform point density, but a portion of data would be neglected in this technique.

The main objective of the SGE task is to find the best substitute geometry based on the fitting criteria from the inspection process. The fitting criteria is typically based on the least squares or maximum of Euclidian distances of measured points from the substitute geometry. A numerical optimization method is required to optimize one of these parameters [17]. Some of the optimization methods find the global optimum condition, but the majority of them are useful for finding the local minimum point. In the employment of local optimization algorithms, it should be considered that the measured points must be close enough to the substitute geometry before running the optimization algorithm, otherwise the output of optimization process may differ from the desired fitting condition.

In 1993, Drieschner [16] used the concept of parallel body to present a new algorithm for efficient approximation of finite point sets by geometric elements. Although this algorithm was closely related to Chebyshev approximation but its results were slightly different from the results of Chebyshev fitting. In 1998, Shakarji [20] implemented a set of least squares fitting algorithms associated to various geometries and combined them together as reference software for NIST²

² National Institute of Standard and Technology

algorithm testing system. Most of those fitting routines used Levenberg-Marquardt (LM) optimization method, because it is an unconstrained method and the initial guess does not influence the calculated global minimum [20]. In a similar work by Shakarji [22], reference fitting algorithms are described at NIST. These algorithms can be used for Chebyshev fitting of lines, circles, spheres, cylinders and cones. At first, a coarse fitting condition is found through least squares method and then it is refined to reach the desired Chebyshev, maximum inscribed or minimum circumscribed fit. In this approach, the simulated annealing method is used to find the local minimum of the defined function. Although the proposed algorithms are not very fast, their results are reliable. The repeatability of the proposed algorithms was tested, and their results were compared with known solutions and industrial results as well. In some cases, it was found out that the result of fittings performed by industrial software is significantly different from results of this reference set of algorithms [22].

In 2002, Dall’Osso [21] applied a variant version of least squares fitting method to two-dimensional data. In this method the functions and known points are not directly related and a residual function is added to the sample data to improve the predictive power of the numerical model. In 2006, Jiang et al. [23] presented a new algorithm based on profile confidence level for evaluating roundness from discrete coordinate data. A good review for roundness evaluation is available in this paper. Testing on data with uniform point distribution showed results are significantly consistent and they are close to the minimum-zone solution of high sample size data [23]. In 2013, Zhang et al. implemented an improved version of Chebyshev fitting method on complex surfaces. By applying the Exponential Penalty Function (EPF) to the original optimization problem, it is converted into an unconstrained differentiable minimization problem. A complete review for different optimization methods is provided in this reference. Comparison of Peak to Valley (PV) values obtained by the new algorithm with PVs from the least squares fitting shows a

notable improvement in them [24]. However, it seems this criterion is not sufficient for the performance evaluation of fitting algorithms.

In a research in 2013 [25], Smitka and Štroner could develop a new noise reduction method based on surface continuity. A Weighted Least Squares (WLS) method is used to fit Chebyshev polynomials to the measured data and to reduce the amount of noise as well. The weight of point deviations was calculated based on the received light intensity or surface slope. Application of this method on sample parts with various shapes (flat, sphere, statue and sharp corner) verified that the weight and the degree of polynomial have slight effect on the noise reduction process.

In Vemulapalli's thesis in 2014 [25], the standard definitions for each tolerance are interpreted to understand the most suitable feature fitting method. It is also found that some of manual inspection practices are not compatible with standard definitions. Following that, normative procedures are proposed for metrological parameters of line, plane, circle, cylinder and width. Based on interpreted data, a library of different feature fitting algorithms is developed and applied to a software in C++ [25]. A valuable review of various fitting methods with statement of strong and weak points of every method is available at this reference.

In a novel research in 2015, Mohan et al. [27] presented a reference library of fitting algorithms for linear, circular and planar features. To reduce the number of analyzed points, two and three dimensional convex hulls are considered that can improve the convergence speed of applied geometric search algorithms. The presented algorithms were tested for accuracy and robustness by virtual method. In this method, a virtual geometry is simulated and imperfect random points are considered around it. One of the interesting results of this work is that different fitting methods need to be used in a certain order to verify the desired tolerance classes and datum simulation [27]. In reference [28] Zhang et al. introduced the Moving Least Squares (MLS) approximation method. This method combines the concept of moving window and compact support weighting function. Due to its local approximation, MLS not only acquire higher precision with

low order functions but also has a good stability. A thorough review for different Least Squares (LS) fitting methods is available at this reference. Application of the new method to the measured electric field intensity and the magnetic flux density data approved the fitting stability of MLS for complex surfaces [28]. In 2016, a general method (modified 2D LS) for automatic surface fitting was presented by Al-Shebeeb et al. [29]. Three main approaches (linear, quadratic and cubic) are applied to this method and their results are compared statistically. It is realized that the number of input samples does not improve the accuracy of models. However, it is effective on the speed of fitting algorithms [29]. In 2016, Moroni et al. [30] used the Chaos Optimization (CO) method to improve the initial solution which is required to be fed to the iterative Levenberg-Marquardt (LM) algorithm. The main application of this method is in fitting of non-linear geometries such as circle, sphere and cylinder, because the optimization of linear geometries can be solved by analytical methods. Different one-dimensional map functions were implemented to find the initial solution. Results show that in general, the logistic map is the most effective one in terms of radius with some drawbacks in higher computational time in certain cases [30].

Recently, in 2018 [31], Hutzschenreuter et al. used Sequential Quadratic Program (SQP) for finding the solution of minimax fitting of cylindrical holes. Sample part was a flange with multiple holes, and the maximum deviation from the reference solution (TraCIM Chebyshev element test data) was about $1 \times 10^{-14} m$ [31]. In reference [32] a modified version of Chebyshev fitting method named as Hybrid Trust Region (HTR) is compared with Exponential Penalty Function (EPF) method. Application on reference data showed HTR returns Peak to Valley (PV) values ten times more accurate than EPF. Also, tests on experimental data from optical aspherical lens showed that PV calculated by HTR is 9 nm less than PV of EPF [32].

The third category of computational metrology tasks (Deviation Zone Evaluation) is about modeling the deviation zone of the entire manufactured surface by using the available limited data

associated to sample points [17]. Research in this category is mainly about various methods of mesh generation and surface reconstruction. This topic will be described thoroughly in section 2.2.

2.1.2. Effect of fitting method on point deviations

From the above literature review it can be understood that the fitting method has a noticeable effect on point deviations. Chebyshev fitting has better compliance with standard tests, and it always returns smaller amounts of maximum deviations. The main disadvantages of this method are time consuming optimization and vulnerability to noise in data. Recent advancements in optimization methods could help with the process time of algorithms and also the accuracy of the optimum situation. However, due to susceptibility to noise, this fitting method is not suitable for the noisy data resulted from optical scan methods. On the other hand, the optimization of Least Squares (LS) fitting method has less complexity, and it is executed faster. This characteristic of LS method makes it useful for profiles with freeform shape. It is important to consider that the maximum deviation is usually over-estimated by LS, but the shape of deviation contour is almost similar to the deviation contour generated by Chebyshev fitting.

As mentioned in some research works in the literature review ([18], [23], [25]), a uniform sampling is often the best case for fitting, but the point distribution is not uniform when an optical scan method is used. One of the weak points of LS fitting method is its susceptibility to non-uniform point distributions because the profile fitted by LS is more heavily influenced by the high density areas. No research has been found that is evaluating the effect of non-uniform point density on the fitting quality of LS method, so this topic was selected as one of the motivations of this research. This parameter can affect the result of profile measurement significantly; therefore, it must be one of the important items of the uncertainty budget.

2.2. Profile measurement for part model generation

In contrast to the previous sections, in the case of profile measurement for part model generation there is no information about the actual part profile, and the main effort is to realize the

actual workpiece surface by using the discrete data of measured points. Optical measurement methods have become popular for this purpose because they are capable of providing more data, but denoising methods are necessary to remove the points with errors from the real profile. In a higher level of surface reconstruction that is mostly related to reverse engineering, sometimes additional simplification procedures are needed to achieve the design data out of the measured profile.

The entire procedure of CAD model generation out of measured data consists of five steps: data collection, data pre-processing, point cloud segmentation, initial meshing and mesh optimization [7].

The first step, data collection, includes all of the actions required for part profile scanning. The output of this phase is one or multiple point clouds that cover the outer surface of workpiece. In the data pre-processing phase, the point clouds are combined together (registration process), and coordinates of measured points are transformed to a common global coordinate system. An initial removal of outliers and noisy data may take place in this phase of CAD model generation process (Figure 7). In the point cloud segmentation phase, more advanced filtering procedures are performed to isolate the target object from the surrounding environment. Additionally, if there are multiple separate bodies in the point cloud, they are distinguished in this phase, so they will not be connected incorrectly in the meshing step. The initial meshing procedure deals with determining how the measured points should be linked together to present the workpiece's surface [7].

The output of initial meshing usually consists of coarse polygonised surfaces that need special treatments such as filling empty areas (bridging and hole-filling) and smoothing to become closer to the real workpiece's surface. Figure 8 presents the output of three last steps of the surface reconstruction process performed on a point cloud resulted from optical scan of a rabbit model. For parts that have complicated shapes such as freeform or sculpture shape, separated surface zones are specified at the mesh optimization phase, and polynomial surface patches are fitted to the mesh

data (Figure 10). This phase of work has a significant effect on the quality of the reconstructed surface. Finally, the surface model is wrapped and converted to a solid CAD model.

Numerous researchers have worked on data filtering, mesh generation and surface reconstruction methods, but this literature review is focused only on recent notable works and those that have overlap with this dissertation.

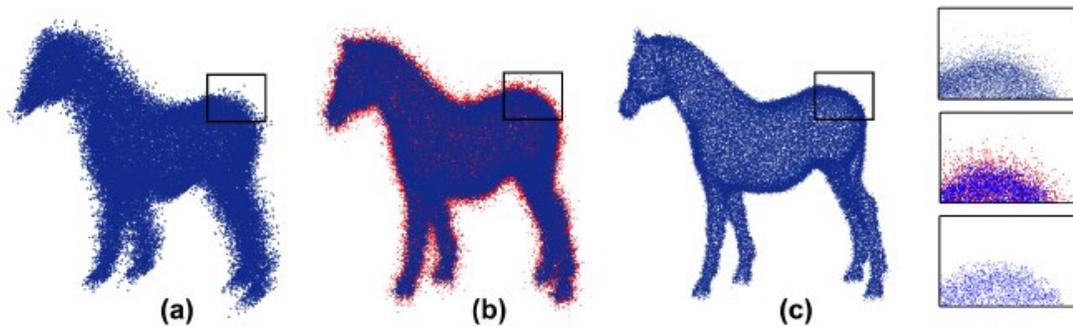


Figure 7. Detection and removal of outliers from the point cloud of a horse model;
 a) The original point cloud with outliers b) The detected outliers are shown in red color c) The result of outlier removal procedure [33]

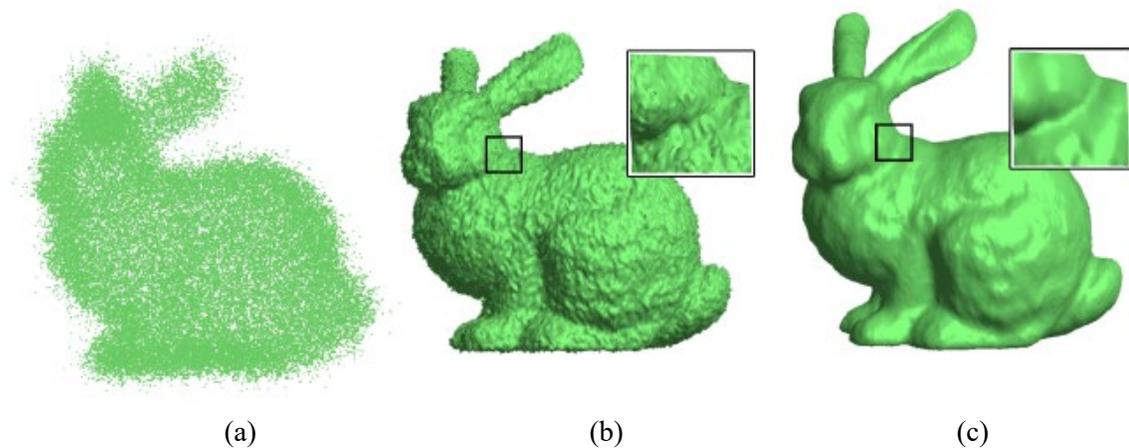


Figure 8. a) segmented point cloud b) initial meshes c) optimized meshes of a scanned rabbit model [33]

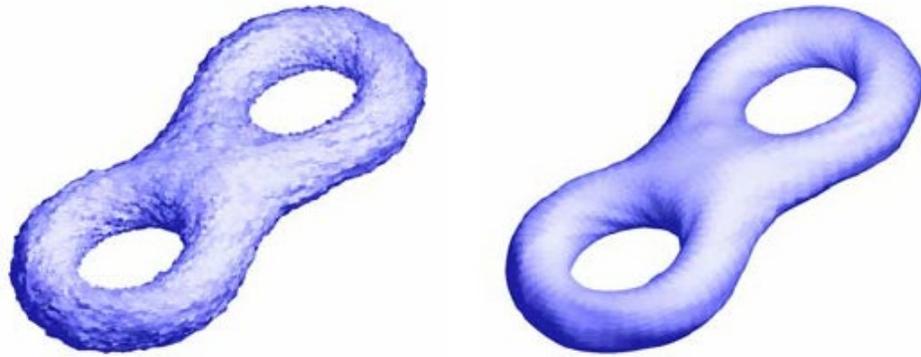


Figure 9. Mesh smoothing on the noisy meshed surface of a 8-shaped part [7]

In an introductory review paper, Varady et al. [35] present different surface reconstruction strategies and describe the advantages and weak points of every data acquisition method. Additionally, specific issues related to characterization of geometric models, surface representation, segmentation, and surface fitting for simple and freeform shapes are addressed. Problems of combining multiple point clouds and creating accurate B-rep models are reviewed, and the limitations of some solutions are discussed [35].

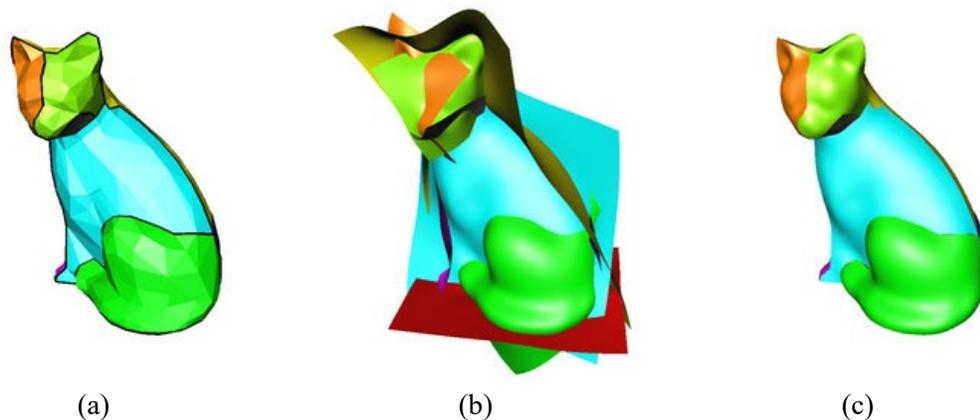


Figure 10. Approximation of a polygonised surface with B-spline surface patches
 a) segregated surface zones b) fitted B-spline surfaces to the specified zones c) fitted surface patches [34]

In data collection, Shen et al [36] integrated a vision system with a motorized probe and could develop a highly automated and precise three-dimensional coordinate acquisition system. In their research, feature recognition algorithms are applied to the data obtained by the vision system

to provide the general information of the unknown object such as location, orientation and topology. The provided information is subsequently used to plan the probing process for high precision and rapid sampling of critical surface areas [36]. In more recent works, the effort was to use the low density high quality data of tactile measurement to correct the high density low quality data associated to the optical measurement. In reference [37], a statistical model of Gaussian Process is developed to correct optical data sets and combine them with tactile data. The corrections are applied locally and the uncertainty of the reconstructed geometry can be calculated at any location. Application of this method to sample parts with free-form surfaces approved that it can improve the precision of point clouds [37]. In reference [38], the concept of virtual markers is used to combine two sets of data. Characteristic points from tactile measurement are used as reference to correct their corresponding points from the optical scan. This method was evaluated by simulation and measurement of three sample parts: flat plane, turbine blade and engine cover. Results showed that the improvement in the distance of optical points from the reference points was from 45% to 99% [38].

A significant portion of research work in data preparation deal with the registration of multiple point clouds obtained from various orientations. In reference [39], a new approach is presented that registers successive views with enough overlapping area to calculate an accurate transformation between views. In reference [40], the Iterative Closest Point (ICP) is used for registration of 3D point clouds to a model shape including free-form surfaces. The model shape can be a set of points, polylines, parametric curves, implicit curves, triangles or parametric surfaces. Turk et al. [41] presented a new method for combination of multiple meshed surfaces into a single polygonal mesh that is conforming to the object surface and has a satisfying performance on large data sets. In reference [43] the Orthogonal Cross Sections (OCS) model is used to register multiple polygonised surfaces together. Then, by fitting B-Spline wrap-around surfaces, an approach for smoothing the seams of registered surface patches is developed. Similar to [40], references [42],

[43], [45], [46] utilized the ICP method for alignment of point clouds, but their approaches for calculation of overlap areas differ from each other. In reference [47], a new strategy for data segmentation is presented and new methods are developed based on non-linear LS fitting for sphere, cylinder, cone and tori shapes.

For initial mesh generation, Mavriplis et al. [48] combined point placement and boundary integrity techniques with Delaunay triangulation to develop a new advancing-front-type mesh generation strategy. In reference [49] Ball-Pivoting algorithm is introduced as an advancing-front algorithm for an interpolating triangulation of a given point cloud. Recently, a new generalized algorithm is developed to convert point clouds with variable resolution, accuracy and occlusion level to uniform meshes that are suitable for part solid modeling [7].

A substantial number of research works in the field of mesh optimization, are about development of new surface fitting methods or polynomial approximation of discrete data points. An ideal fitting method is robust against noisy data and generates a smooth continuous surface. Also, it keeps the sharp features and has reasonable accuracy on freeform shapes.

Hiroshi [51] used the cubic B-spline fitting and defined a new smoothing function to minimize the L_2 norm composed of the data residuals and the first and second derivatives. In reference [52], a new algorithm based on Singular Value Decomposition (SVD) of a 3 by 3 matrix is developed for finding the LS solution of fitting two point sets with equal number of points. Levin [53] expressed that the moving least-squares method works very well for polynomial interpolation, smoothing and derivative approximation. Amenta and Bern [54] developed an algorithm that approximates a smooth surface to a finite set of sample points. This algorithm uses Voronoi vertices to remove triangles from the Delaunay triangulation. It is shown that on high density areas the output of this algorithm is topologically valid and converges to the original surface [54]. Ohtake et al. [55] proposed a new automatic mesh smoothing method that can preserve sharp features. Wagner et al. [34] presented a new method for generating a smooth surface that approximates a

given manifold triangulation with arbitrary topology. In this work, a set of necessary conditions are derived that guarantee the continuity of the result. Then by using these conditions, a new algorithm is developed for a piecewise polynomial surface reconstruction that generates a small number of patches. In reference [56], multiple mesh smoothing methods were compared together, and a new method based on linear diffusion of mesh normals is developed. This method demonstrates the best performance in denoising meshes with sharp features and high resistance to oversmoothing [56]. Fleishman et al. [57] introduced a new robust moving least-squares technique for reconstruction of a piecewise smooth surface from a noisy point cloud. The result is a unified framework that not only reduces the noise in data, but also enables modeling of surfaces with sharp edges [57]. In reference [58], Vieira developed a new method for creating a piecewise-smooth surface from the surface patches extracted by region growing method. In this method the measured points are projected iteratively onto polynomial surfaces for a progressive approximation of larger neighborhoods. The constructed surface has a satisfying visual quality that is comparable to a B-rep surface [58]. Zhao et al. [4] applied the IGA (Immune Genetic Algorithm) and presented a new algorithm for fitting B-spline surface patches to a noisy 3D point cloud. B-spline surfaces are reconstructed by using LS approximation and then the surfaces are optimized by applying IGA-based algorithm to adjust the knots. Finally, the surface patches are stitched together with the Particle Swarm Optimization (PSO) method to keep the surface continuity [4].

In an interesting work, Huang accomplished all of the five required steps for CAD model reconstruction from the scan data. In this dissertation, a new procedure is developed to extract the topological and geometrical information of the workpiece from the measured point cloud and generate a CAD model associated to the measured profile. This procedure starts with reconstruction of workpiece shape by interpolating manifold meshes with an efficient growing process and then applying a mesh optimization algorithm to refine them and to decrease the difference between the original surface and the polygonised surface. In the second step mesh segmentation is executed.

This algorithm performs the border detection and region growing. Afterwards, B-Spline surface patches are fitted to the segmented mesh sets by using a non-iterative method. Finally, the CAD model is generated from the established mesh topology and feature surface representation [59].

2.2.1. Effect of data processing method on reconstructed surfaces

In the previous section, it was understood that the measured data sets cannot be used directly for creation of an object model, and several processes are needed to convert them to a suitable digitized model. Each of these processes apply some changes to the original data, so all of them have a direct influence on the quality of reconstructed surfaces. If these affecting factors are executed appropriately the measurement uncertainty can be reduced; if the uncertainty is too great, the generated model may have major differences with the real workpiece. By considering the literature review of this field, it can be understood that the uncertainty of measured profiles has been evaluated rarely. Only references [17],[18],[19],[23] and [37] discussed the task specific uncertainty of measured profiles in their research, and in all of them except [37], the part profile has a simple shape such as flat plane or cylinder. Therefore, it can be understood that there is no general standard method for quantification of profile measurement uncertainty. This method is necessary for measurement evaluation, but quantification of profile measurement data is a prerequisite for that goal. In summary, there is an extensive need for a quantification method that arranges profile measurement data to a specific standard structure and works well for any profile shape even freeform surfaces.

2.3. Mean Local Deviation (MLD) as a method for profile quantification

Previous sections of this chapter discussed different purposes of profile measurement, and described various methods for reaching to the planned goal briefly. Finally, the importance of quantification of profile measurement data was discussed, but no general quantification method has been proposed in any research so far. The desired quantification method must be applicable to any profile shape. Furthermore, it should provide a criterion for comparison of multiple measured data

sets. It is important to consider that data sets may have different number of points, density, spatial organization, associated noise, and various coverage on the measured surface. So single points belonging to different point clouds may not fall into the same location on the workpiece surface [44]. This fact that is referred to as “lack of co-localization” [44] is the main issue in the profile data quantification process. To solve this issue, using the Mean Local Deviation (MLD) of profile segments is proposed in this research. For this purpose, the whole part profile is divided into segments with equal length (2D profiles) or area (3D profiles). After that the optimum fitting condition is found, the deviation of every individual point is calculated. In the next step, point deviations of every segment are averaged. The Mean Local Deviation (MLD) not only decreases the effect of noise in data, but also provides a quantified criterion that can be used for comparison of different measured data sets. More characteristics of the proposed quantification method will be discussed in the next chapters.

CHAPTER 3: POINT DENSITY AND PROFILE FITTING PROCESS

In the previous chapter, it was understood that a uniform point sampling pattern is the most ideal case for fitting methods. It not only does cover the profile consistently but also has a neutral effect on the fitting result; that is why it is normally selected for profile tactile measurement, and only the distance between measured points is set before the execution of measurement program. The measurement time of tactile methods is longer than the time of optical scanning methods, and it is the main reason for popularity of optical methods. In addition to that, optical methods provide extensive amounts of data of the part profile, but their data also encompass higher amount of noise in comparison with associated tactile data. Another flaw of optical measurement methods relates to their non-uniform point distribution. In profile scanning methods, the position of points on the workpiece's surface is measured from reflected light rays. Since the reflected light scatters out in various directions its intensity varies at different spots on the CCD camera, so it causes a variation in the density of measured points. The non-uniform pattern becomes troublesome when fitting methods such as Least Squares (LS) are used because they are vulnerable to the density of point clouds, and the final fitted result is shifted toward profile segments with high point density. Due to this vulnerability, some commercial software packages use down-sampling strategies and neglect a portion of measured data. Depending on the size of the kept data, this strategy can improve the speed of the fitting process significantly, but certainly its accuracy is lower than the accuracy of using the whole measured data set. The main goal in this chapter is to analyze the effect of non-uniform point distribution on the LS fitting method and to develop algorithms with more sensitivity to point positions on the profile rather than the reoccurrence number of point deviations.

3.1. Weighted Least Squares (WLS) and Highly Weighted Least Squares (HWLS) fitting algorithms

In order to understand the influence of non-uniform point density on the result of the fitting process, two different fitting algorithms are defined based on the LS fitting method. In both of these

algorithms, a weight is defined for every point deviation based on distances between projected points on the profile (2D data) or the xy plane (3D data). So it is necessary that the point cloud is fitted roughly to the nominal profile, otherwise it is not possible to find the coordinates of the projected points. In this condition, points farther from other ones would have higher weights, and points in high density areas would have lower weights. The only difference between WLS and HWLS algorithms is in the power of assigned weights in the weighted sum of squares of point deviations. Equations 1 to 3 present the target function of LS, WLS and HWLS fitting algorithms. To equalize the dimension of output parameters it is needed to calculate the square root of the Sum of Squares (SS), Weighted Sum of Squares (WSS), and Highly Weighted Sum of Squares (HWSS) of individual points.

$$\sqrt{SS} = (D_1^2 + D_2^2 + \dots + D_n^2)^{\frac{1}{2}} = \sqrt{\sum_{i=1}^n D_i^2} \quad (1)$$

$$\sqrt{WSS} = (W_1 D_1^2 + W_2 D_2^2 + \dots + W_n D_n^2)^{\frac{1}{2}} = \sqrt{\sum_{i=1}^n W_i D_i^2} \quad (2)$$

$$\sqrt{HWSS} = ((W_1 D_1)^2 + (W_2 D_2)^2 + \dots + (W_n D_n)^2)^{\frac{1}{2}} = \sqrt{\sum_{i=1}^n W_i^2 D_i^2} \quad (3)$$

D is the deviation of each point (normal distance to the nominal profile), and n represents the total number of measured points. With regard to profile constraints the transformation matrix is defined, and input variables (rotational and translational parameters) are linked to the output of the target function. Assuming α , β and θ as rotation angles around coordinate axes and dx , dy and dz as translational elements, the input matrix of functions is in this form:

$$RT_{in} = \begin{bmatrix} \alpha & \beta & \theta \\ dx & dy & dz \end{bmatrix} \quad (4)$$

Since point clouds are fitted roughly, the rotation angles will be small, so rotation matrices around x , y and z axes are defined as equations 5 to 7.

$$Rot_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha) & -\sin(\alpha) \\ 0 & \sin(\alpha) & \cos(\alpha) \end{bmatrix} \quad (5)$$

$$Rot_y = \begin{bmatrix} \cos(\beta) & 0 & \sin(\beta) \\ 0 & 1 & 0 \\ -\sin(\beta) & 0 & \cos(\beta) \end{bmatrix} \quad (6)$$

$$Rot_z = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (7)$$

With small angle approximations these equations are converted to equations 8 to 10. It should be considered that rotation angles are in radian in the latter set of equations.

$$Rot_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & -\alpha \\ 0 & \alpha & 1 \end{bmatrix} \quad (8)$$

$$Rot_y = \begin{bmatrix} 1 & 0 & \beta \\ 0 & 1 & 0 \\ -\beta & 0 & 1 \end{bmatrix} \quad (9)$$

$$Rot_z = \begin{bmatrix} 1 & -\theta & 0 \\ \theta & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (10)$$

The general matrix of rotation around three coordinate axes is calculated by the product of the above matrices. If rotation angles are large their sequence matters, but with small angles transformations can be in any sequence. Therefore, the transformed coordinates of every point are calculated as below:

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = Rot_x \times Rot_y \times Rot_z \times \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} dx \\ dy \\ dz \end{bmatrix} \approx \begin{bmatrix} 1 & -\theta & \beta \\ \theta & 1 & -\alpha \\ -\beta & \alpha & 1 \end{bmatrix} \times \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} dx \\ dy \\ dz \end{bmatrix} \quad (11)$$

In the next step, an optimization method is required for finding the optimum condition of the defined target functions. For this problem, in this research, *fminsearch* —a Matlab built-in function— is used. It is a nonlinear programming solver that is utilized for finding the local

minimum of functions by means of an iterative derivative-free method. It means that the mentioned function uses only discrete values of the target function, and there is no need to know derivative values.

3.1.1. Weight definition for two-dimensional profiles

In new weighted fitting methods, all measured points are projected onto the nominal profile, and then the middle points between the projected points are found by calculation. Every midpoint has equal distance to its neighbor projected points. The length between the midpoints prior and after each measured point is considered as the weight of its deviation. Where a particular feature ends can also be used as a limit for the length influenced by a measurement point. Figure 11 presents a schema of how the deviation weights are calculated for the considered simple profile shape.

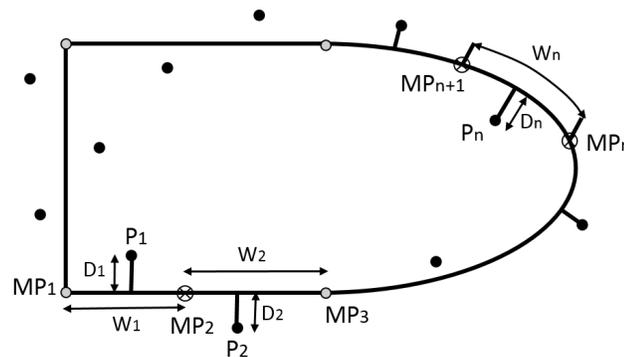


Figure 11. Measured points (P_1, P_2, \dots), midpoints (MP_1, MP_2, \dots) and weights (W_1, W_2, \dots) of deviations (D_1, D_2, \dots) for a simple profile

3.1.2. Weight definition for three-dimensional profiles

The weight definition for the three-dimensional space is slightly different from weighting 2D profiles. First of all, by using the *delaunayTriangulation* function in Matlab, a 2D Delaunay triangulation is applied to the considered point set. In the next step, coordinates of nodes are used to calculate the area of each triangle. Every triangle area is shared between three nodes, so every node receives one third of that area. The weight of each node is defined as its total share from

surrounding triangles. In this method, a less compact area means larger triangles that are equal to higher weights and vice versa. The sum of all weights should equal the projection of whole measured area over the projection 2D plane (its default in Matlab is xy plane). It can be tested after that deviation weights are assigned. Figure 12 shows the steps of Delaunay triangulation of a 3D point cloud. As it is seen in Figure 13, the density of points may vary over generated triangle meshes.

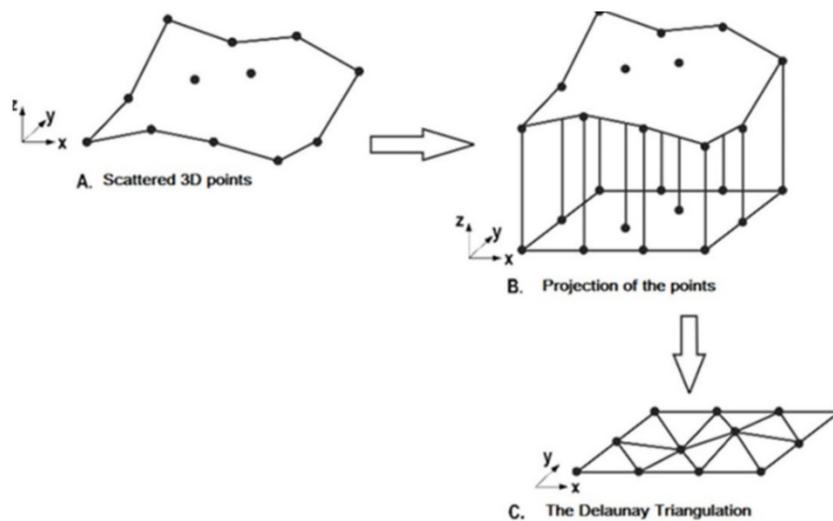


Figure 12. Steps of Delaunay triangulation of scattered points [64]

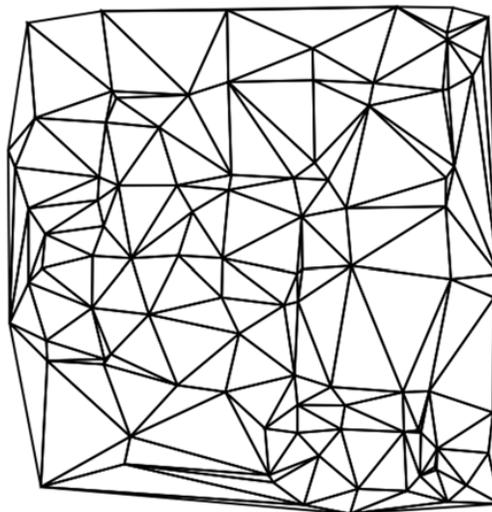


Figure 13. Delaunay triangulation with varying point density [65]

3.2. Theoretical simulations

To understand the behavior of the three introduced algorithms some theoretical simulations are performed. In these simulations, a profile with a simple shape is considered, and then a set of imperfect points with random deviations is generated around it. The deviation of points is randomly selected from a specific range that is close to the typical range of deviation in experimental data. The introduced algorithms are used to fit the virtual data to the assumed profile. This procedure is repeated several times with uniform and non-uniform point distributions, and the fitting parameters as well as maximum point deviations are analyzed.

3.2.1. Square profile

Initially, to understand the effect of point positions and symmetry, a square with dimension of 4 millimeters and seven points on each side (28 points in total) is assumed. Projected points have 0.5 millimeters distance from their neighbor points (feature point or projected point). Point deviations are constant, and in each case some points are removed to generate a different point layout. In this work, positive deviation is used for when the fitted point is outside of the profile and vice versa. Twelve generated layouts are shown in Figure 14. A different color coding is used for each side for easier identification of points belonging to every feature.

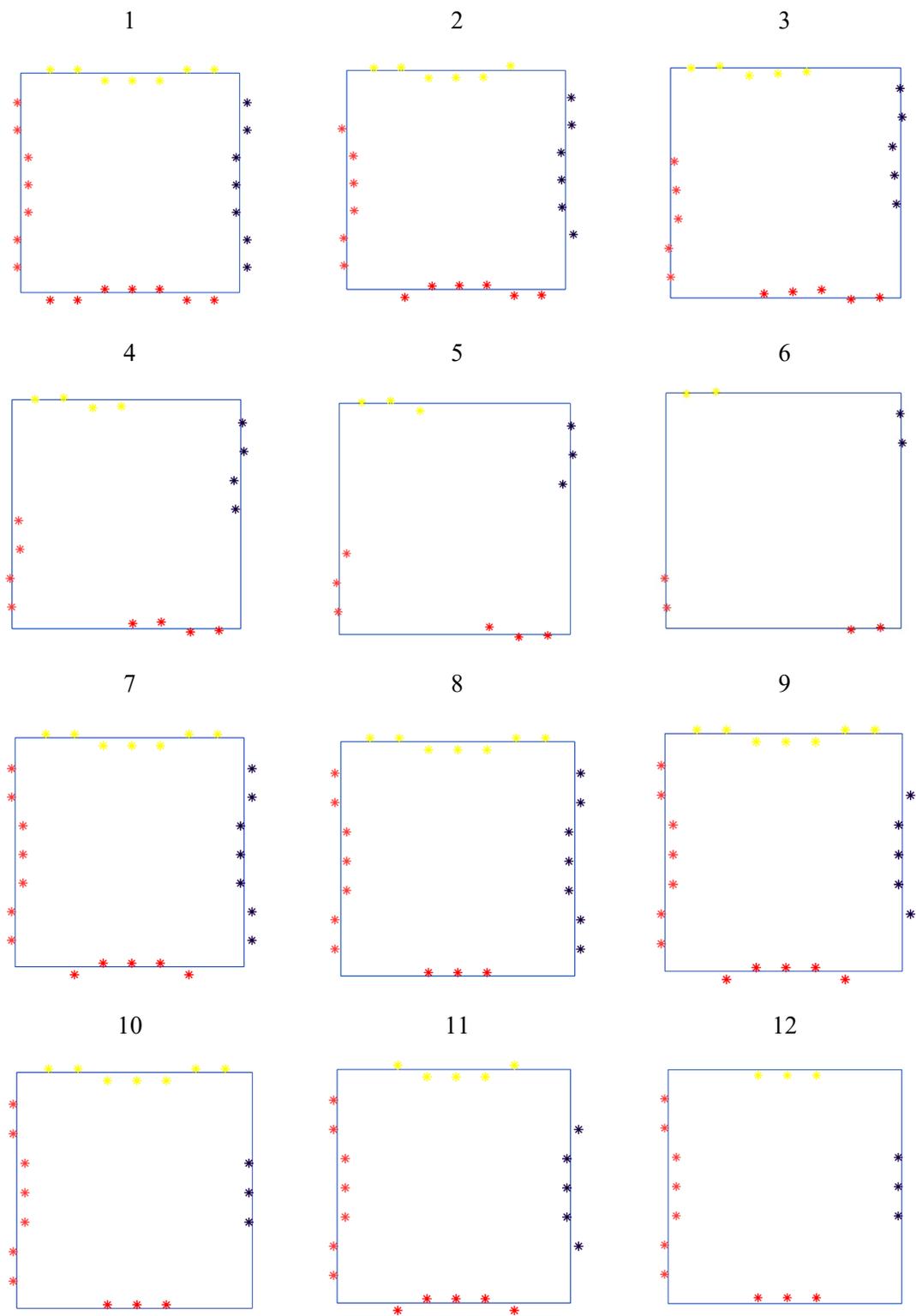


Figure 14. Twelve different point layouts used for the analysis of square profile

3.2.2. Quarter circle profile

To analyze the statistical behavior of LS, WLS, and HWLS fitting algorithms, virtual point clouds with added random deviations from a quarter circle profile with radius of 5 millimeters are created. For uniform point distribution, projection points with a constant distance from each other are used, and random deviations are imposed to their positions. In contrast, for non-uniform point distribution, the location of projected points is not fixed, but the number of points belonging to every feature is kept constant.

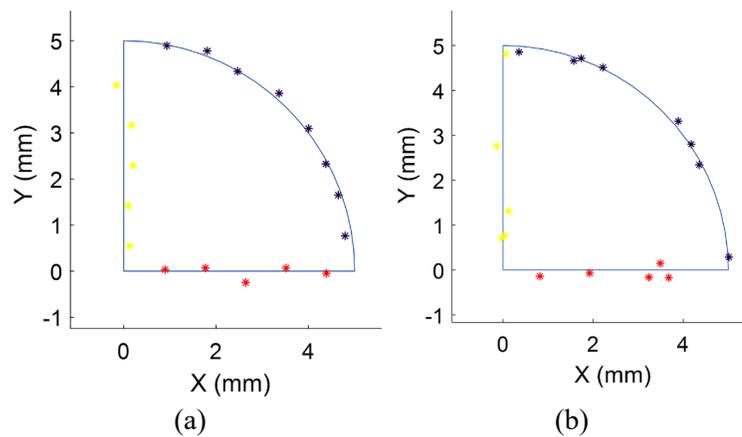


Figure 15. Two types of point distribution over quarter circle profile; a) Uniform point distribution; b) Non-uniform point distribution

For additional evaluation, both of normal and uniform distribution of random deviations are applied. The range of deviations for the uniform distribution model is from -0.2 to 0.2 millimeters, while the normally distributed model has a wider range of about -0.345 to 0.345 millimeters. In this condition, the standard deviation of both models equals 0.115 millimeters. Random point sets are created and fitted to the quarter circle profile by using the three fitting algorithms, and this action is repeated 100 times for each type of point distribution. To acquire a general picture about the statistical behavior of each algorithm, obtained fitting parameters are

saved as well as maximum deviations. Histograms of variables and relation graphs will be discussed in Chapter 5.

3.2.3. Saddle profile

For comparing the performance of three introduced fitting algorithms in 3D space, a saddle surface is assumed, and virtual point clouds with random deviations are generated and fitted to this freeform surface. Equation 9 represents the saddle surface mathematically.

$$z = \frac{x^2 - y^2}{250} \quad -50 < x < 50 \quad , \quad -50 < y < 50 \quad (12)$$

Similar to the quarter circle profile, two types of point sampling strategy (uniform and non-uniform) are simulated. Also, point deviations are selected randomly from one uniform and one normal distribution with the constant standard deviation of 0.115 millimeters. Points of uniform sampling have an organized grid structure on xy plane with 5 millimeters distance intervals and covers a 90 by 90 millimeters square area (Figure 16.a). As seen in Figure 16.b, the x and y coordinates of non-uniform points are randomly selected from the range of $[-45, 45]$. The total number of simulated points in both cases is 361 (19×19). It can be understood from above that there is no sample point on the edge areas because, in reality, data of these zones are typically full of noise, so point deviations are higher on zones close to sharp edges. For each type of point cloud, 100 samples are generated and fitted by means of LS, WLS, and HWLS algorithms. It is important to mention that NO datum is considered in these fittings, so there are 6 optimized variables associated to the degrees of freedom that need to be found for each virtual point set. Obtained fitting parameters and maximum deviations will be reported and discussed in Chapter 5.

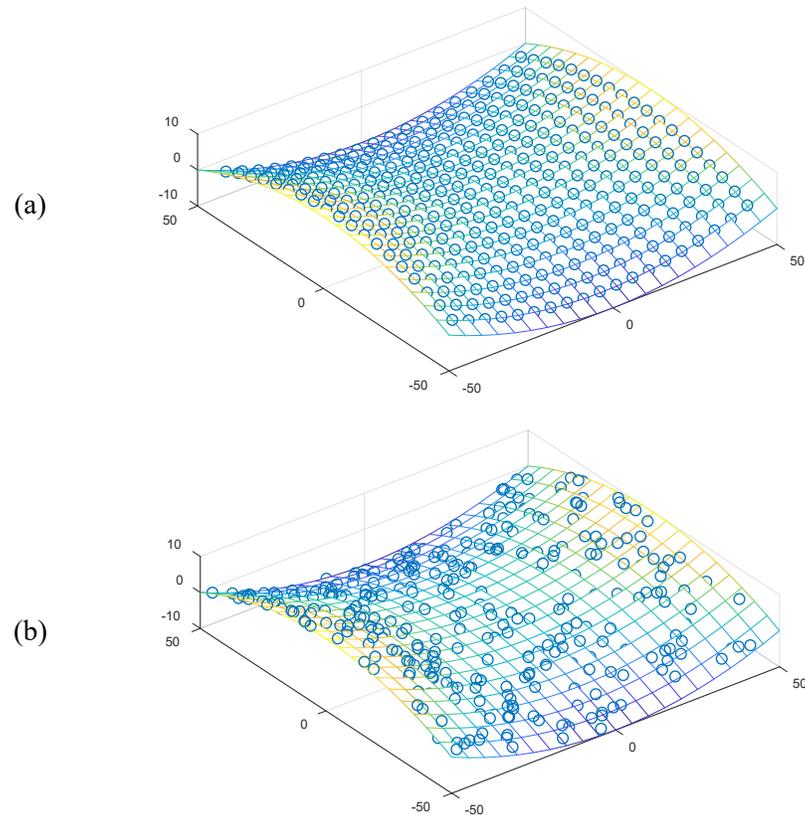


Figure 16. (a) uniform and (b) non-uniform point distribution over saddle profile

3.3. Experimental measurements

This section describes the scanning and analysis of both 2D and 3D profiles. After workpieces are cleaned they are positioned on a worktable (either the CMM base or a free-standing table) and clamped properly. The reference and profile surfaces are scanned using either tactile or optical sensors. Before any scan, the desired datums should be defined, and a part coordinate system is created based on those datums. This establishes a part coordinate system for the initial reporting of data.

3.3.1. Measurement and analysis of 2D profiles

Two sample parts with non-regular profile shapes made of rolled steel plate were selected (Figure 17). The profile of these parts was manufactured by CNC machining and all the edges were

deburred. The measurand on part 1 (outer profile) is the position (x and y coordinates) of points on the outer surface around the part. The initial Cartesian system is defined by the top surface and two perpendicular edges (datums B and C). For part 2 the measurand (slot profile) is the coordinates of the projected points of the inner surface of slot onto the xy plane. The x and y axes are defined by two perpendicular edges on part profile. The total set of profile points will be rotated and translated to fit to the nominal profile, so the initial coordinates will change in the next step.

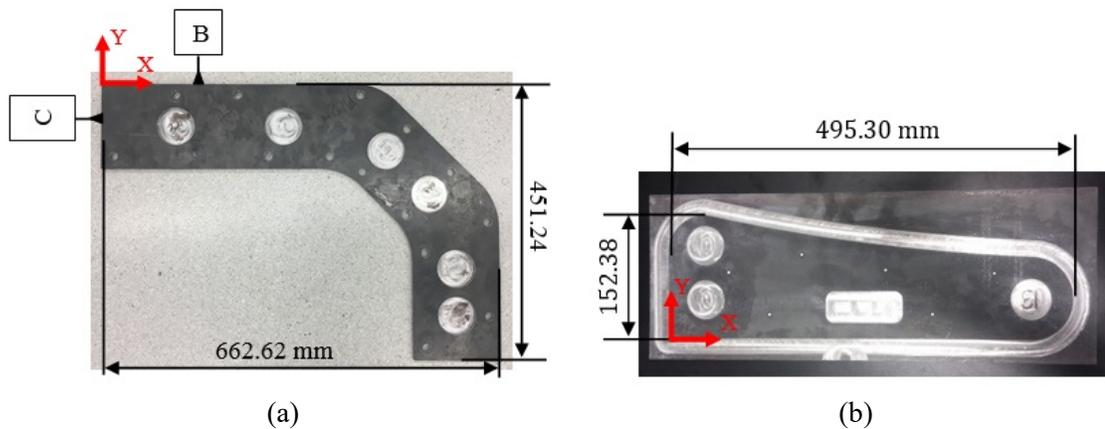


Figure 17. Sample parts with (a) outer profile and (b) slot profile for two-dimensional profiles and their general dimensions [62]

The above parts were scanned using five different techniques: a Cartesian CMM (touch-trigger probe), a Laser Tracker (contact of SMR pin nest), a Blue Light Scanner (fringe projection technique) and a measuring arm (using both a tactile probe and an optical triangulation sensor). Some of these instruments are shown in Figure 18 to Figure 20. In tactile measurement the instrument reports the position of probe center, and it is therefore necessary to calculate the position of contact point, since profile tolerances (like other tolerances) control the surface of the part. This can be performed if the probe radius is assumed constant along different directions (ideal probe) that is not true in reality, but it is important to know that errors in probe radius are negligible in comparison with other types of error. One of the main steps of data preparation in tactile measurement is to offset the measured points in the profile plane (xy) toward the profile line, but this offset direction has to be known. Some advanced tactile probes are able to measure the direction

of contact force that is exerted normal to profile surface. Even though it seems that the output data of measurement with this type of probes do not need any offset, but in fact they are not accurate enough to perform good compensation. Therefore, it is required that prior to offsetting the measured data, a scaled profile is fitted to the measured points and then an offset equal to the probe radius is applied to the points (Figure 21) normal to the fitted profile [62].

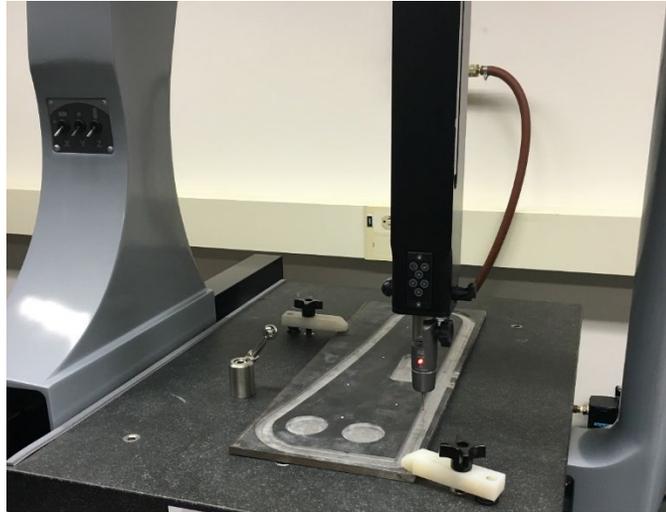


Figure 18. Profile measurement with a Cartesian CMM

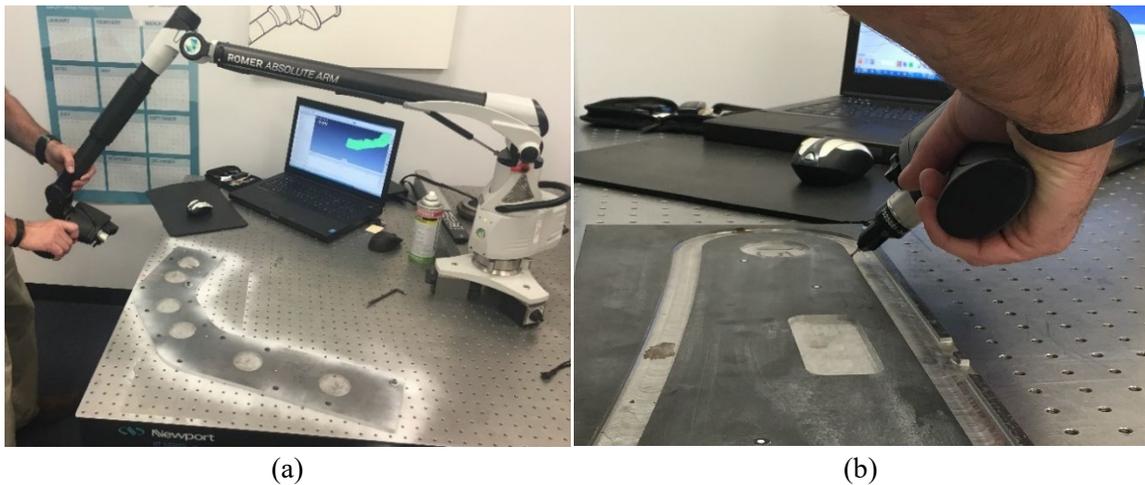


Figure 19. (a) Optical scan of the outer profile and (b) tactile measurement of slot profile with Romer arm



Figure 20. Measurement of the outer profile with a FARO ION laser tracker

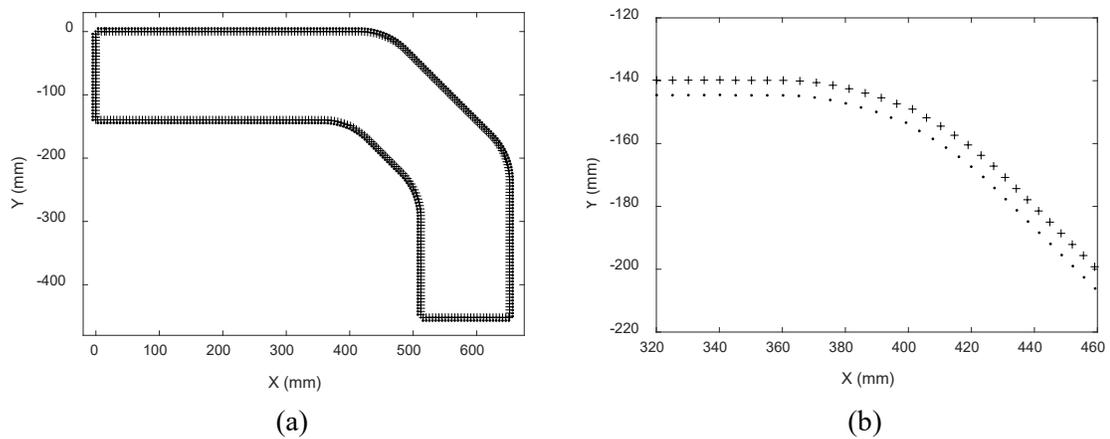


Figure 21. Offset in tactile data; (a) general profile shape (b) detail of dashed zone; measured points are shown with dot and calculated contact points are represented with cross [62]

Similar to the tactile measurement, the output of optical measurement is coordinates of points on the part surface, but they form a disorganized 3D point cloud, and profile points appropriate for analysis need to be extracted from that cloud. One common way to extract points for the evaluation of a 2D profile is to section the cloud with a horizontal plane that is parallel to xy plane (Figure 22). Due to mostly multiple reflections on the part surface, there are typically some outlier points that should be removed (Figure 23) before evaluating the profile for conformance.

One method to remove outliers is by defining a threshold for the point deviations from the nominal geometry and deleting the points whose deviations are larger than the considered threshold. Another method is to remove points that are significantly different in location from their neighbor points.

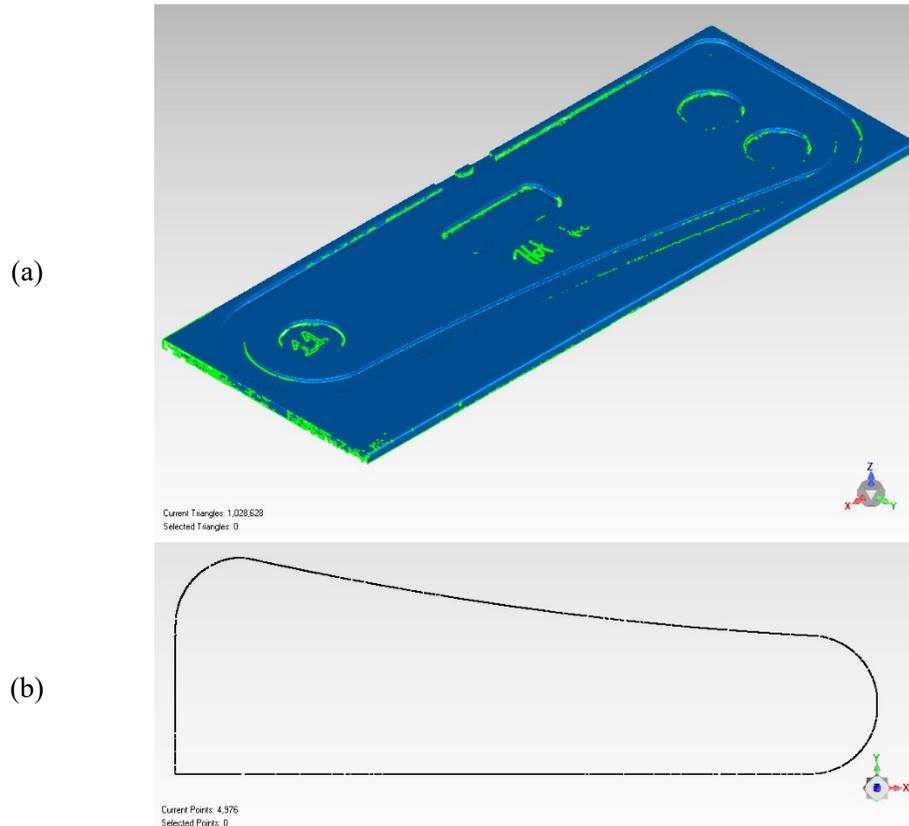


Figure 22. (a) point cloud obtained by optical scan (b) 2D profile created by sectioning and editing the point cloud

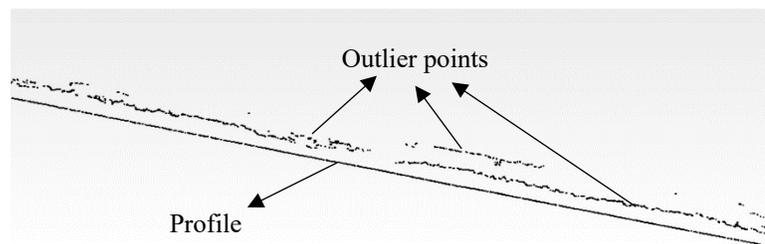


Figure 23. Outlier points over the profile data

After that measured data of 2D profiles are prepared, they are fitted to their associated nominal profile with LS, WLS, and HWLS fitting algorithms, and point deviations are calculated through the code developed in Matlab software. It should be considered here that since measured points are free within xy plane, they do not have more than three degrees of freedom (translation along x and y , and rotation around z). Results of applying different fitting algorithms to measured 2D data will be presented in Chapter 5.

3.3.2. Measurement and analysis of 3D profiles

Two sample parts with dissimilar freeform shapes are selected and scanned to provide real 3D data for evaluating the performance of three presented fitting algorithms (LS, WLS, and HWLS). Both of these parts are made of aluminum, and they are machined to a freeform shape on their top section. The first part has a wavy shape, and the nominal equation of its top surface is presented in cylindrical coordinate system as the below equation.

$$z = \frac{1}{75000} r^3 (\cos(\theta) + \sin(5\theta)) \quad 0 < r < 50, \quad 0 < \theta < 2\pi \quad (13)$$

The second workpiece has a saddle shape and the formula of its top surface is identical to the equation of the saddle surface in section 3.2.3 (Equation 12). Figure 24 presents pictures of these sample parts.

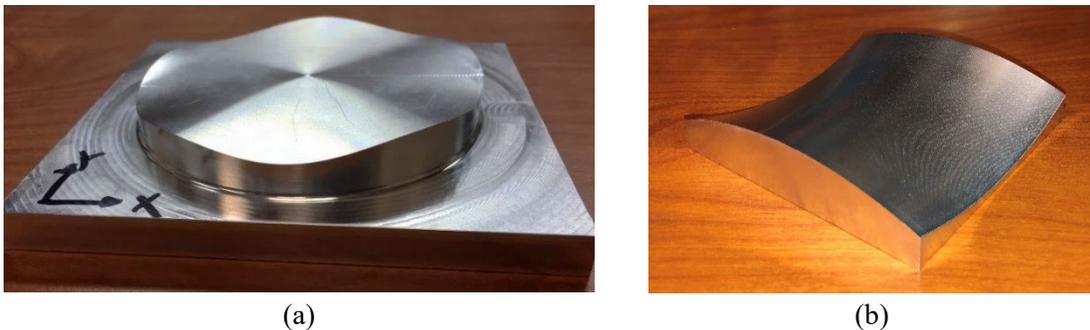


Figure 24. Aluminum sample parts with freeform top surfaces; (a) wavy profile; (b) saddle profile

The top surface of the wavy part is measured with a Cartesian CMM (tactile probe), Romer arm (optical), and Blue Light Scanner (optical). The saddle profile is scanned with both of the mentioned optical methods (no tactile measurement). Since both parts have shiny metal surfaces, their optical scan is tedious and time-consuming. In industrial applications, this type of surfaces are typically sprayed to decrease their shininess and improve the scatter of light from considered surfaces. To understand the effect of spraying on the measurement result, both sample parts are scanned once without any spray (shiny metal surface) and one more time with a coating of powder on their top surface (Miconazole nitrate powder on the wavy surface and PTFE powder on the saddle surface). Figure 25 and Figure 26 show the position of optical instruments and their distances to the profile surface during the scan process.

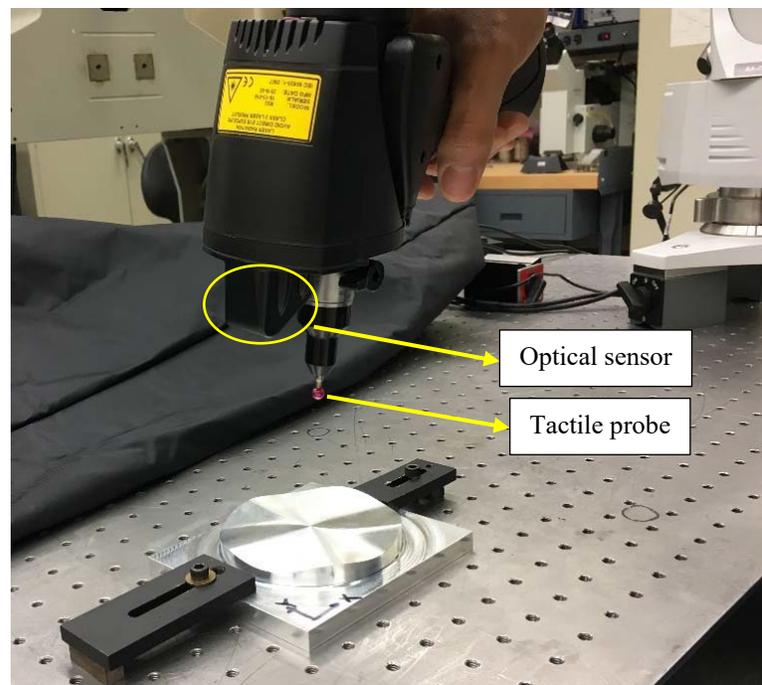


Figure 25. Optical scan of wavy profile with Romer arm (linear triangulation sensor)

As mentioned before, an offset needs to be applied to tactile pure data to find contact points on the wavy surface. For every individual measured point all distances to other points are calculated and the two closest points are selected. By using coordinates of the neighbor points and building two vectors starting from the measured point the normal direction can be calculated by vector

product of the two vectors. Then moving the point along the normal direction by assumed constant dimension of probe radius would specify the coordinates of contact point.

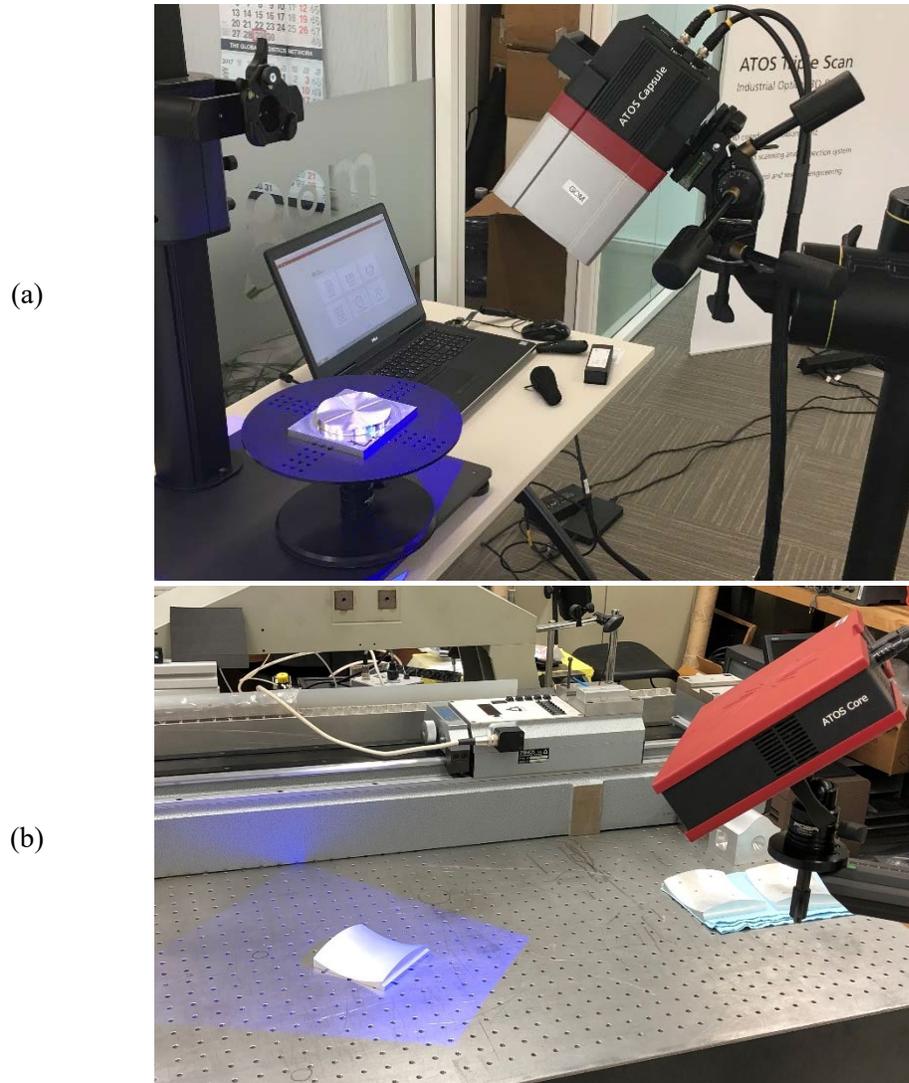


Figure 26. Optical scan of freeform surfaces (a) wavy profile, and (b) saddle profile with Blue Light Scanner (fringe projection technique)

Optical data do not have the offset issue, but since part surfaces are shiny the amount of noise would be high especially on edge areas. To avoid this issue, the *pcdenoise* function is used to reduce the noisy data. This function calculates all distances between measured points, and removes outlier points. A point is considered as an outlier if the average distance to its neighbors is above a threshold. The default value of this threshold is one standard deviation from the mean of

average distances of all points. It is required to be stated here that by applying the *pcdenoise* function to some data sets measured by Romer arm, a noticeable portion of data at less dense areas is removed, so this function is not used for denoising of those data sets. Additionally, after fitting, edge data points are filtered out, and from wavy surface data only points with radial distances less than 49.7 mm from the origin of PCS (Part Coordinate System) are kept. However, a wider edge zone is assumed on the saddle profile, and from fitted data sets, measured points that are in a square area ($-45 < x < 45$ and $-45 < y < 45$) are saved for further analysis. Therefore, points that are located on a 5 mm wide band from the edges are removed from measured point clouds.

It is important to know that data matching is performed in three steps. In the first step, only deviations in z direction are considered to reach an initial match. The second step consists of fine matching by using the surface equation and calculation of perpendicular vectors and point distances (regular LS fitting). The third step is to rematch measured points considering their normal deviations and weights (WLS and HWLS fitting). Figure 27 presents how the matched data set (fitted by LS algorithm) is generated from pure tactile data. The initial PCS is defined on the corner of the top flat surface on the workpiece. After that contact points (red points in the figure) on the wavy surface are found, the coordinate system is translated to the center of the wavy surface. That is the reason that points matched to the wavy surface, comparing with contact points, are about 11 millimeters shifted in the negative z direction.

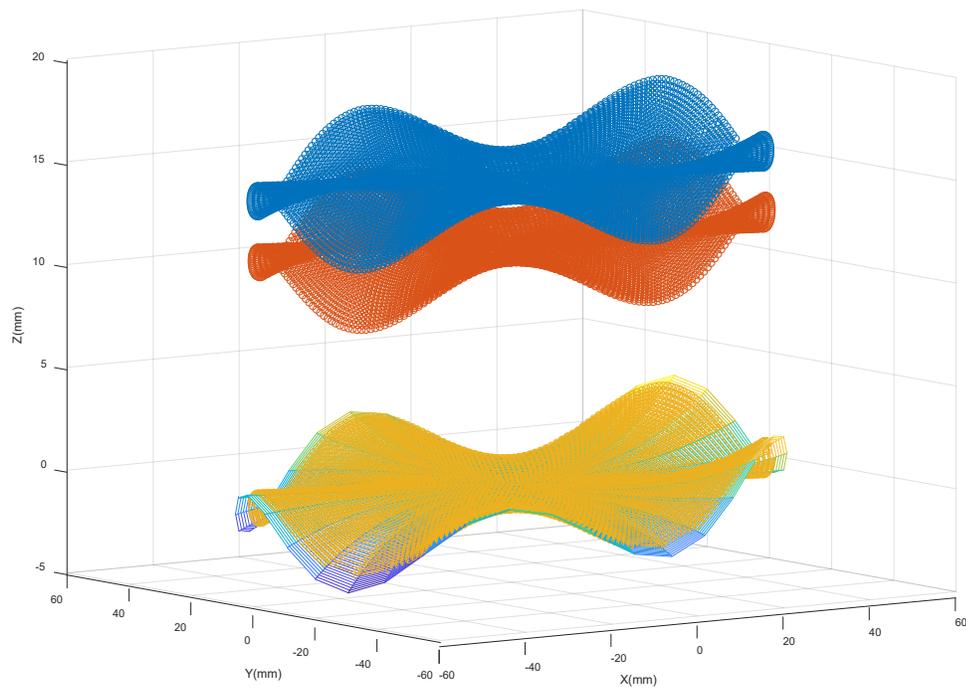


Figure 27. Progress from measured data (blue) to contact points (red) and points matched to the wavy profile surface (orange)

Figure 28 shows the top view of two data sets associated to the wavy surface. From comparison of these two pictures the difference in point distribution between tactile and optical measurement can be understood.

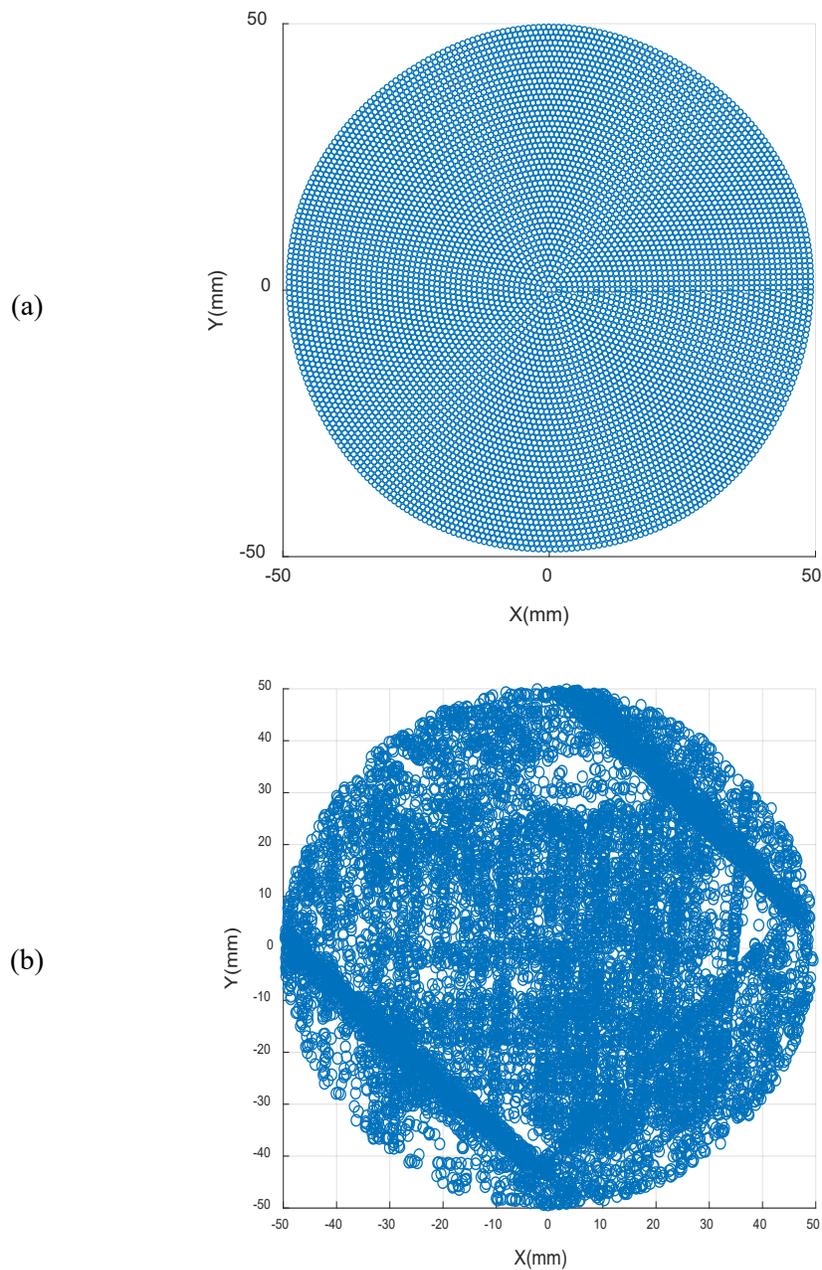


Figure 28. Top view of measured (a) tactile and (b) optical data of wavy profile

To provide a uniform point distribution over the saddle surface, a grid of points with 5 mm distance intervals is created on xy plane, and then this grid is projected in z direction onto the surface made by polygonization of scanned data (Figure 29). These projected points are saved separately

as data with a uniform structure. In the next step, the whole data and the sampled one are matched to the nominal saddle surface by using LS, WLS, and HWLS algorithms. Results related to the quantification of measured profile data and comparison of different fitting algorithms will be presented in Chapter 5.

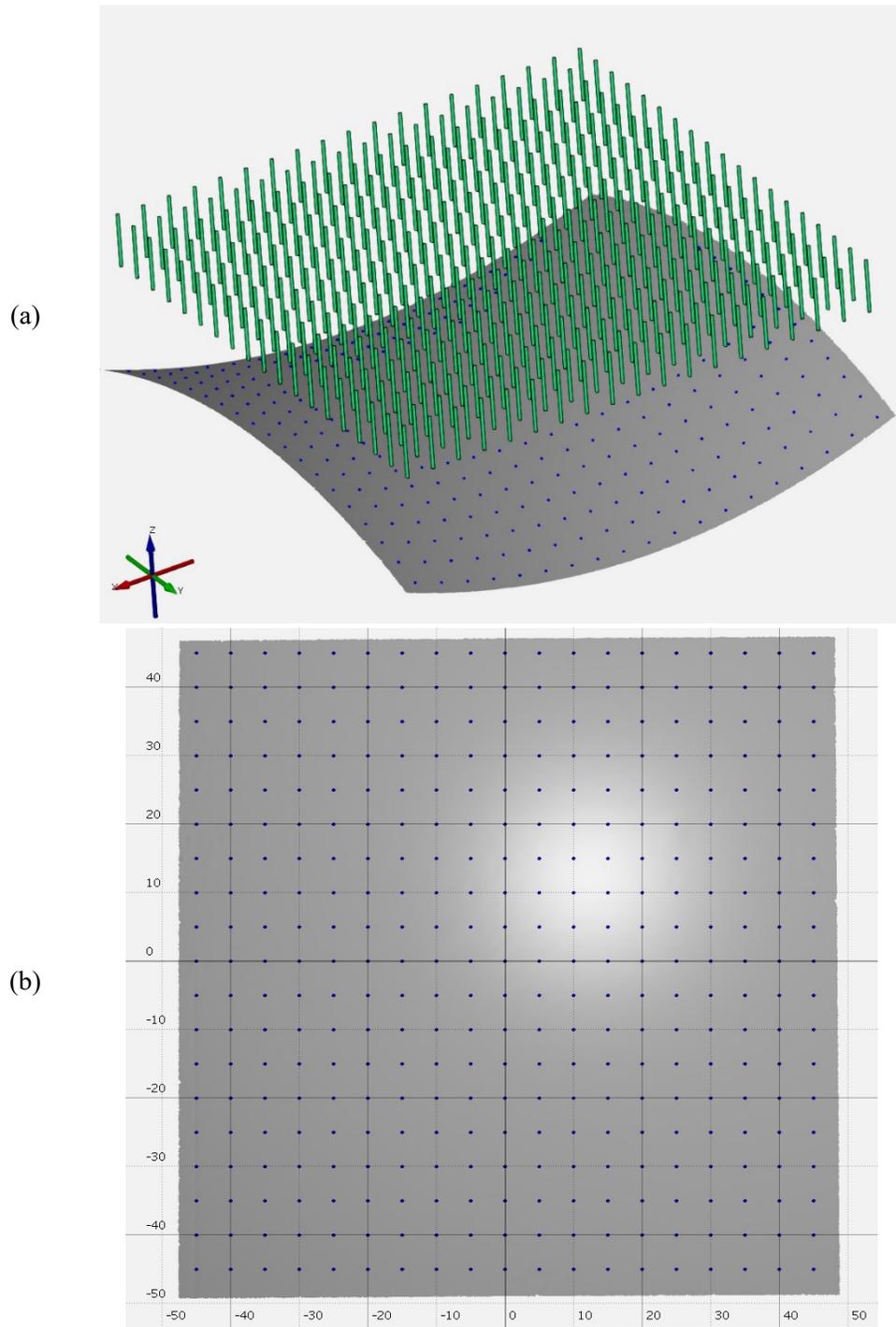


Figure 29. Projection of a grid of points onto a polygonised surface to create a uniform sample (a) 3D view, (b) top view

CHAPTER 4: QUANTIFICATION OF PROFILE DATA

In the previous chapter, it was discussed that the number and position of measured points may vary when multiple instruments or sensors are used. Due to this lack of co-localization, measured profile data sets associated to one sample part cannot be compared directly. Therefore, a criterion is needed to fill this gap, and to make the profile data comparable. The main idea of this research is to investigate methods— such as a convolution averaging filter— that will convert the individual point deviations to Mean Local Deviations (MLDs), and then use the MLD on each profile segment as a criterion for comparison of multiple profile scans. This criterion can be used for quantification of measured profile data and the profile measurement error as well.

4.1. Quantification of 2D profile data

By matching point clouds to the nominal profile, the position of projected points and the deviation of measured points are available. The point deviation is defined as the distance of a point to the profile element along the direction perpendicular to the profile. For closed profiles, the direction inside the profile is described as a negative deviation, and the direction outside of the profile is positive. Following the application of this fitting technique, and plotting the deviations versus the distance around the profile, it is possible to show the deviation of each segment of the profile regarding its position on the profile. The result of this procedure is similar to unbending the profile and showing measured points along a straight line that represents the distance to the origin point. In Figure 30, point deviations of the measured slot profile are shown. Color coding is used to distinguish between the different features. One of the main advantages of this technique is the ability to isolate zones that contain large deviations, which can be corrected in the manufacturing process.

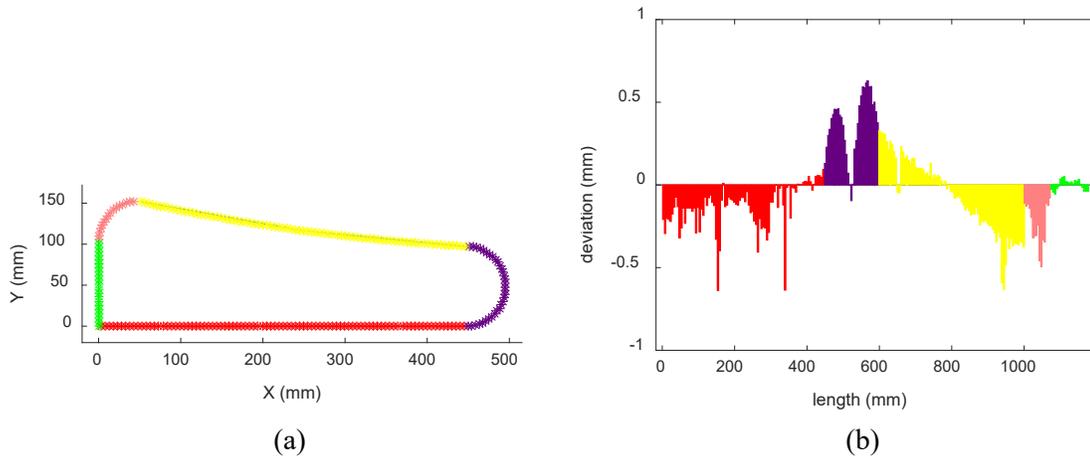


Figure 30. (a) measured points matched to the slot profile; (b) graph of point deviations versus profile length [62]

To calculate the MLD of each profile segment, a window with a fixed length is assumed. This window is moving at fixed steps along the horizontal axis in the point deviation graph, and averages of deviations inside the window are calculated and considered as the deviation of each segment. As an example, Figure 31 shows a point deviations graph and the associated local deviations graph belonging to the measurement of the slot profile with the Blue Light Scanner (BLS). In this example, the window length and step length are assumed respectively as 30 mm and 10 mm. The proper selection of window and step lengths makes it possible to calculate the described mean local deviation of each segment and use it as a criterion for the profile measurement error quantification. It can be understood that sudden jumps and drops of point deviation are filtered by this quantification method. Because of this effect, the MLD plot has a smoother shape [62]. More discussions about how filter parameters influence the MLDs of profile segments are available in reference [62]. Another benefit of using MLD criterion is in the verification of instrument repeatability, when a single part profile is measured with the same instrument several times. In this condition, it can be better understood how much variation occurs in profile measurement data due to the instrument, without requiring that the exact same points be measured each time.

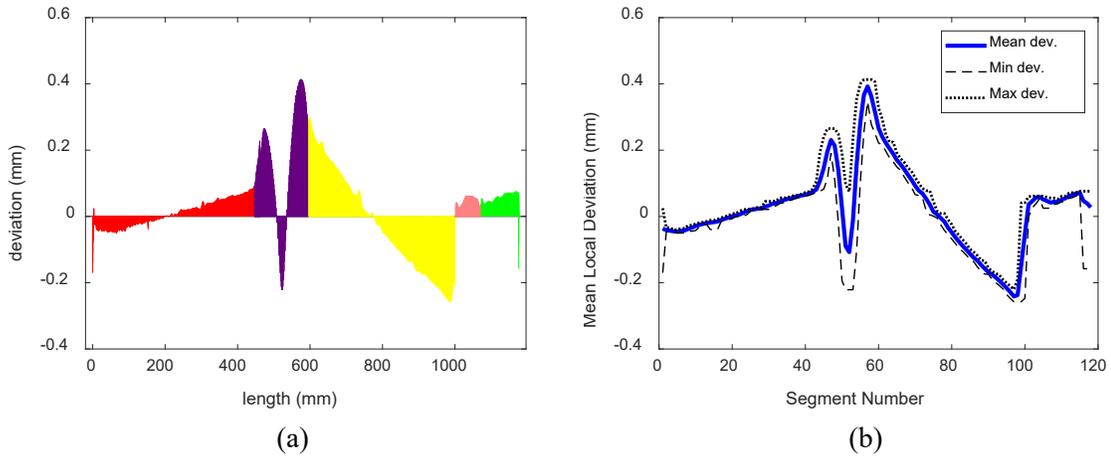


Figure 31. (a) graph of point deviations of the slot profile; (b) graph of local deviations after applying the convolution filter [62]

4.2. Quantification of 3D profile data

The main idea in this type of quantification is nearly similar to the quantification of 2D profile data. In this approach, after the measured points are matched to the nominal profile, the projection of the profile surface on the xy plane is divided into segments with equal area, and then the average of all point deviations within every profile segment is calculated. Since the projected surface of wavy and saddle profiles have circular and square shapes, it is recommended to segment them in different methods.

4.2.1. Segmentation of circular surface (wavy profile)

For the wavy profile, circular sectors with constant width of 10 mm are considered, so the inner and outer radius of sectors are selected from below series:

$$R_i = 0, 5, 10, \dots, 45 = 5i - 5 \quad (i = 1 \text{ to } 10) \quad (11)$$

$$R_o = 10, 15, 20, \dots, 55 = 5i + 5 \quad (i = 1 \text{ to } 10) \quad (12)$$

One quarter of the central circular area ($25\pi \text{ mm}^2$) is considered as the constant reference, and angles of next sectors were defined to keep their area equal to this reference amount. Since the width of segments are constant, their angle becomes smaller by radial movement from the center to the outside (Figure 32). The angular step is defined as two-third of the segment angle while the

radial step is 5 mm. By this method, the projected circular surface is divided into 328 segments with equal areas.

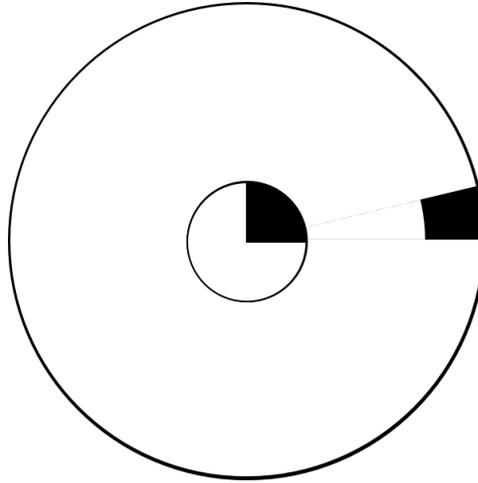


Figure 32. Decrease of the circular sectors angle from the center to the outside of the projected area

If the average of point deviations located on every segment is shown at the center of the segment, a constant uniform structure will be achieved that can be used for presentation of all measured data sets related to the wavy profile. Figure 33 shows an example of normal point deviations from the nominal wavy profile. This data set is measured with the optical sensor of the Romer arm and is matched by the LS method. By applying the described quantification method to this data set, the MLD data of profile segments are obtained and are presented in Figure 34. The constant structure of quantified data is visible in Figure 34(b).

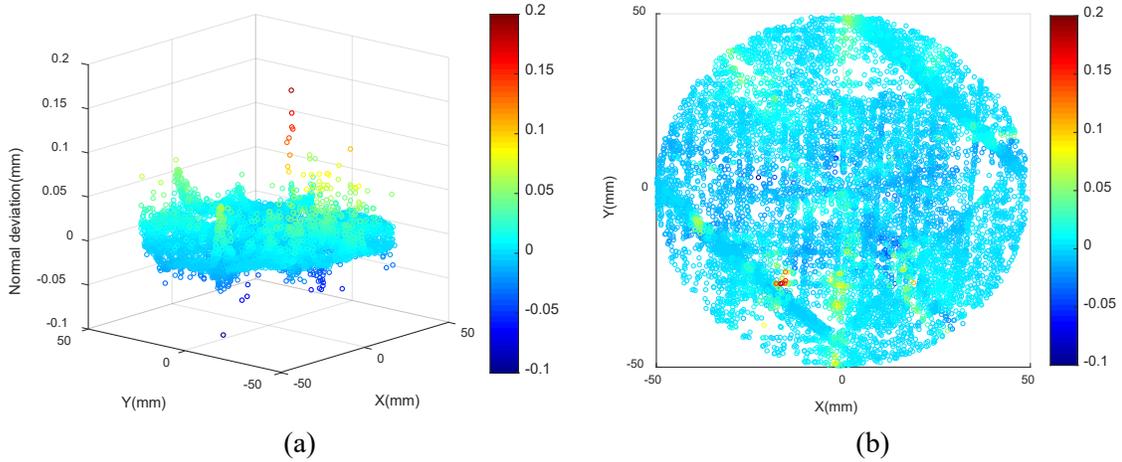


Figure 33. Deviation of individual measured points from the wavy profile; (a) 3D view; (b) top view

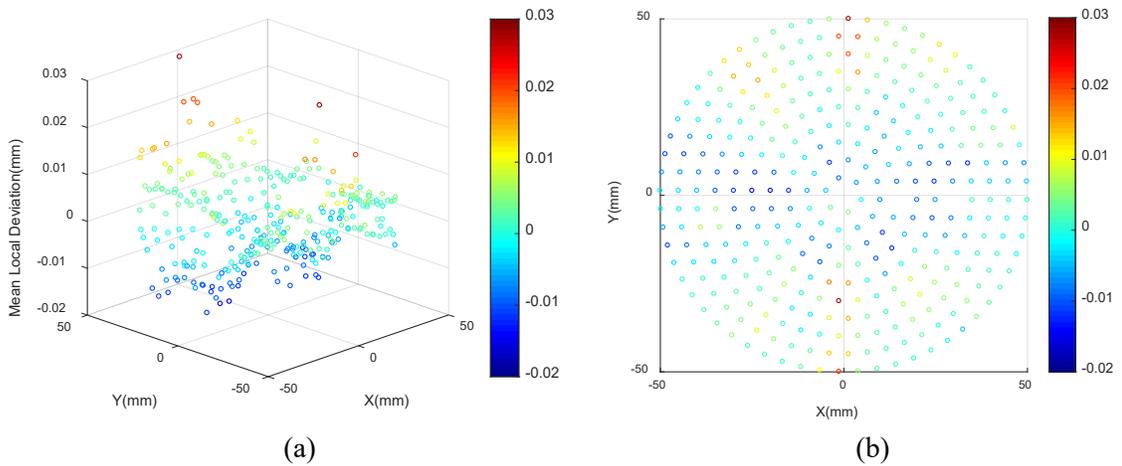


Figure 34. MLD plot of a point cloud measured by the Romer arm; (a) 3D view; (b) top view

4.2.2. Segmentation of square surface (saddle profile)

Segmentation of the saddle profile is executed in a different way from partitioning of the wavy profile. Since the projected surface has a square shape, it will be easier to consider segments with a square shape. For the saddle part, looking from the top, a square segment with the dimension of 12 mm is assumed. In the first step, this square covers the area with the minimum x and y (bottom left corner), and the average of point deviations belonging to points locating inside the square is

calculated. For this case, horizontal and vertical steps are set as one-third of the segment dimension. Therefore, this square sector moves along the x axis (direction 1 in Figure 35) by step lengths of 4 mm, and the MLD of each segment is computed until the first row is covered. For the second row, all squares of the first row are shifted along the y axis (direction 2 in Figure 35) by 4 mm, and related MLDs are calculated separately. The same procedure is repeated for the next rows until the whole projected surface is covered by square segments. The described segmentation method is presented schematically in Figure 35. In this figure, four corner profile segments are identified with their segment numbers.

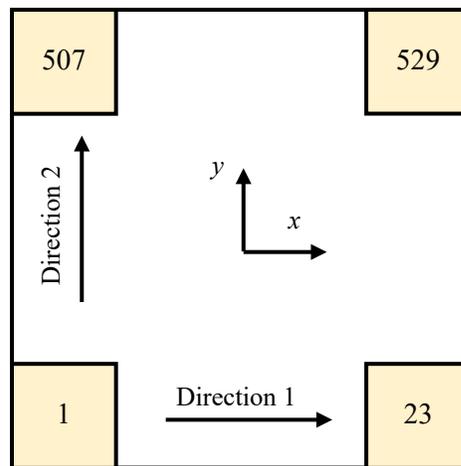


Figure 35. Segmentation of the projected square surface

In the next chapter, the described method will be used for the quantification of data sets (theoretical and experimental) relating to the saddle profile. For example, if this method is applied to point deviation data shown in Figure 36, quantified data presented in Figure 37 will be obtained.

It can be understood by comparing the range of data (original and quantified) and the continuity of deviation graphs that the unfavorable effect of noise in original data is vastly removed or decreased in quantified data. However, the performance of filtering severely depends on the number of points at noisy segments.

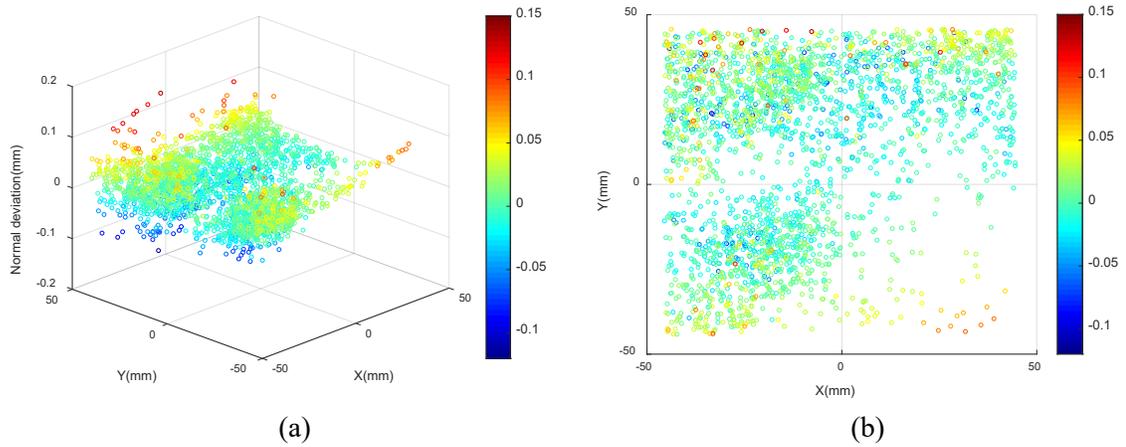


Figure 36. Point deviations of a fitted data sample from the saddle profile; (a) 3D view; (b) top view

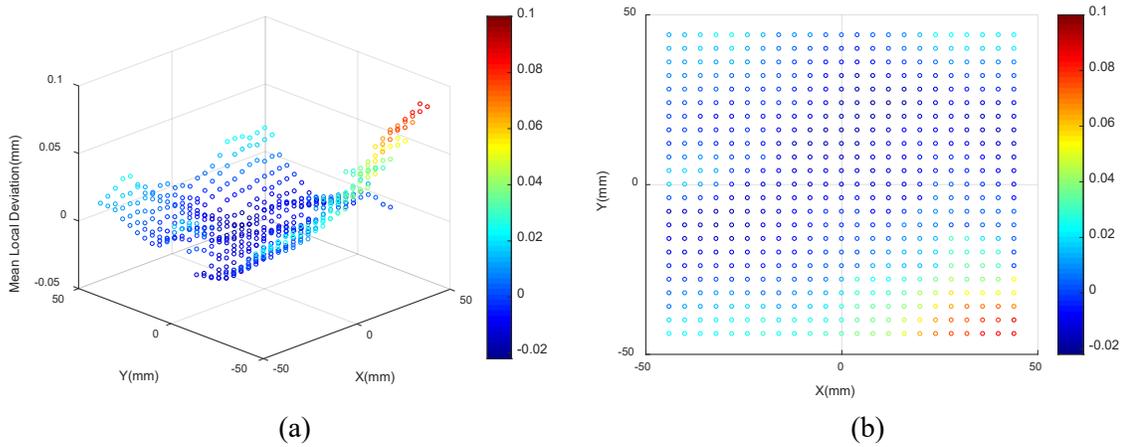


Figure 37. MLD plot of a data set resulted from measurement of the saddle profile with the Blue Light Scanner; (a) 3D view; (b) top view

CHAPTER 5: RESULTS AND DISCUSSION

In this chapter, at first results of theoretical simulations will be presented and discussed. Then, the second part of this chapter will be about results of experimental measurements, performed on 2D and 3D profiles.

5.1. Results of simulations

5.1.1. Square profile

As mentioned in Chapter 3, twelve different point layouts are considered around the nominal square profile, and these point sets are matched by LS, WLS, and HWLS methods. The profile dimension is assumed as 4 mm, and before fitting, all points have 0.1 mm (positive and negative) deviation from the profile. To have a more general analysis, the coordinate system origin is located initially on the corner and then at the center of the square. In both cases, to test the uniqueness of found optimum conditions, different values for the initial guess as well as the stop limit are fed into fitting algorithms. Some of the found fitting parameters are summarized in the Appendix A. It can be understood that if the target stop limit is not small enough, the found optimum condition may vary by initial guess. Therefore, a tiny stop limit and a high iteration limit are needed to ensure that found parameters are unique. It is comprehensive from the data shown at the table that 10^{-12} is a reasonable threshold for the stop limit of final variables.

Optimum fitting parameters associated to 12 point distribution layouts are available on graphs shown in Figure 38 and Figure 39. The first general result that is perceivable from the range of data in 3 pairs of graphs is that rotations in the second case (origin at the center) are generally higher than rotations of the first case (origin at the corner), while the ratio of translations are opposite. Furthermore, it is visible that fitting parameters resulting from the WLS method are between fitting parameters of LS and HWLS methods.

Regarding the robustness of three fitting algorithms, the variation of data in graphs prove that in the first case (origin at the corner), except for translation in x axis, there is not a significant difference between various fitting methods. However, HWLS has more robustness in the second parameter (dx). In contrast, the robustness of LS method is remarkably higher in the second case (origin at the center), but this result is only about the robustness of fitting parameters.

For a complete analysis of robustness, it is necessary to evaluate the data of point deviations as well as fitting parameters, because smaller deviations with the same data set are interpreted as higher quality of fitting. Graphs Figure 40 and Figure 41 present the maximum, minimum, and range (maximum minus minimum) of point deviations related to various layouts fitted by LS, WLS, and HWLS algorithms. As expected, graphs shown in both figures are identical, which means the result of fitting processes is independent from the position of the origin. In other words, the obtained fitting parameters are equivalent to two different ways for the description of optimum conditions of target functions. Additionally, from the first graph (min-max deviations), it can be understood that HWLS algorithm has a bias toward negative direction, while deviations resulted from LS algorithm are biased to more positive values, and deviations related to WLS algorithm are between those two (LS and HWLS). Most importantly, the consideration to the range of deviations declares that in the majority of layouts (10 out of 12), the WLS algorithm conveys a deviation range lower than or close to the deviation range carried by the LS algorithm. Nonetheless, these results show that the HWLS algorithm has an extensive level of sensitivity to how points are distributed, and it ensues to deviation ranges mostly higher than ranges of data derived by LS and WLS algorithms.

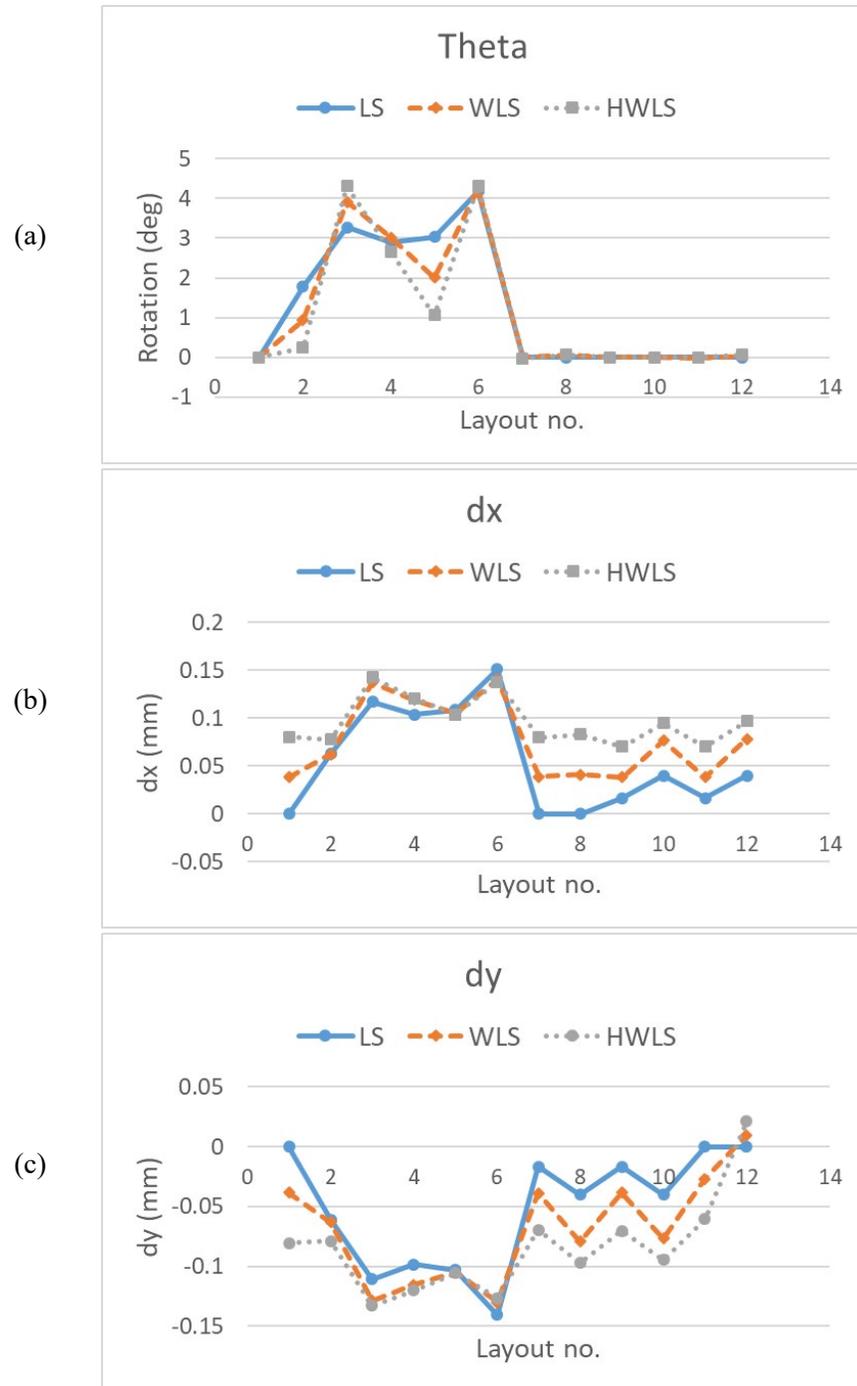


Figure 38. Graphs of optimum fitting parameters when the origin is on the corner of the square (a) rotation angle, (b) translation in x axis, (c) translation in y axis

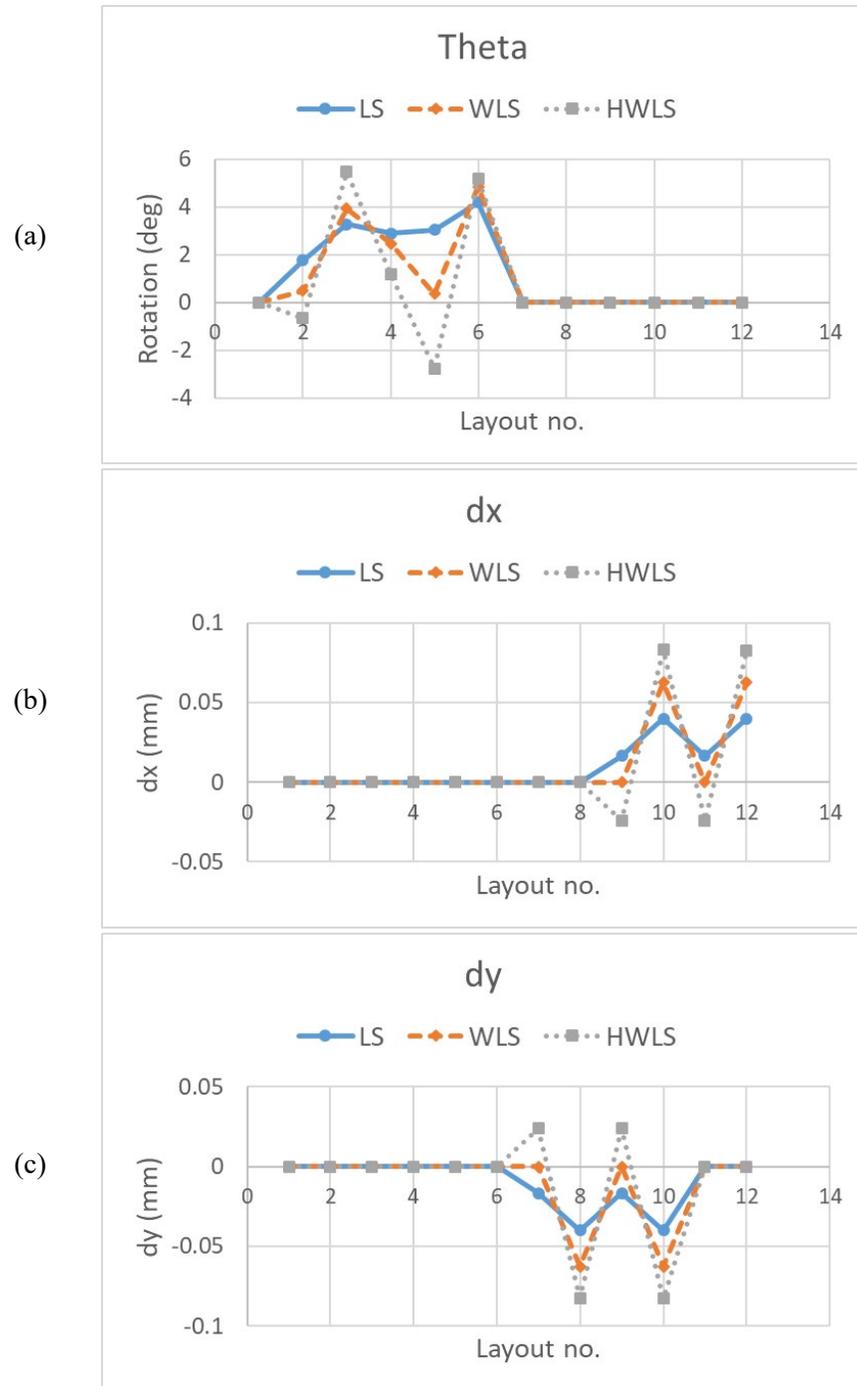


Figure 39. Graphs of optimum fitting parameters when the origin is at the center of the square (a) rotation angle, (b) translation in x axis, (c) translation in y axis

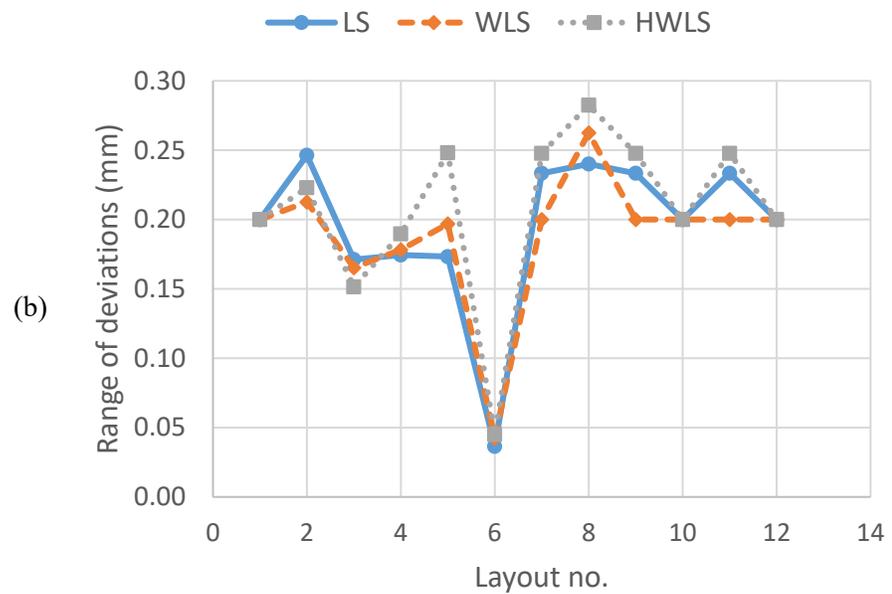
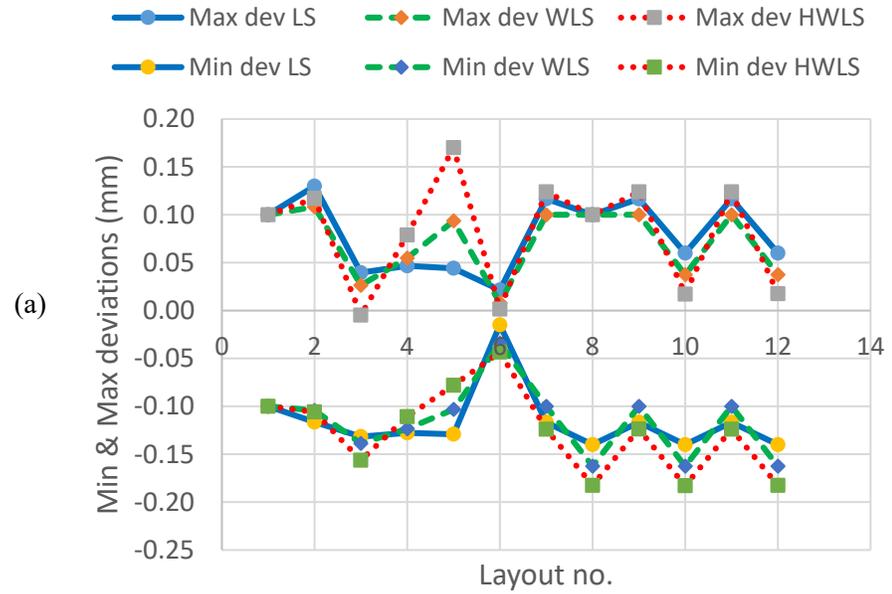


Figure 40. (a) Minimum and maximum point deviations, (b) range of deviations related to different point layouts (origin at the corner)

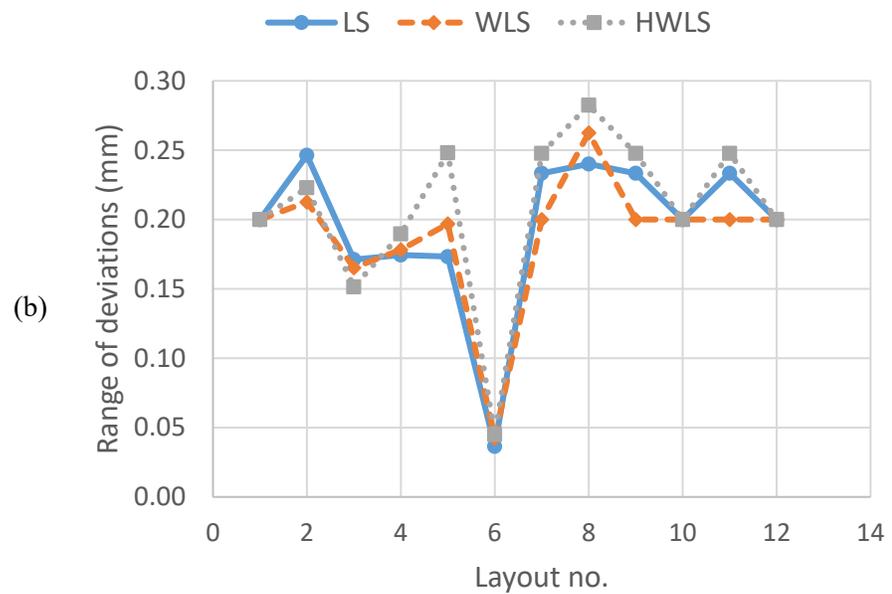
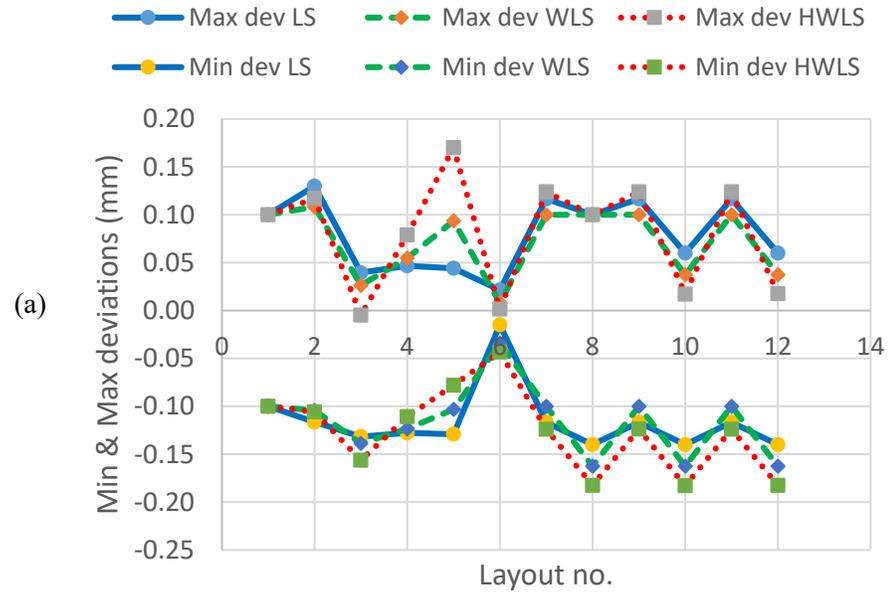


Figure 41. (a) Minimum and maximum point deviations, (b) range of deviations related to different point layouts (origin at the center)

5.1.2. Quarter circle profile

Before starting the discussion about the statistical analysis of fitting random points to this profile, two simple cases are compared together. In the first case, 16 points with equal distances between their projections (including feature start and end points) and 0.2 mm deviation are considered around the quarter circle profile (Figure 42.a). For the second case, some points are removed from the initial point set (Figure 42.b). These two point sets are fitted by means of the three introduced algorithms, and then acquired fitting parameters and point deviations are compared.

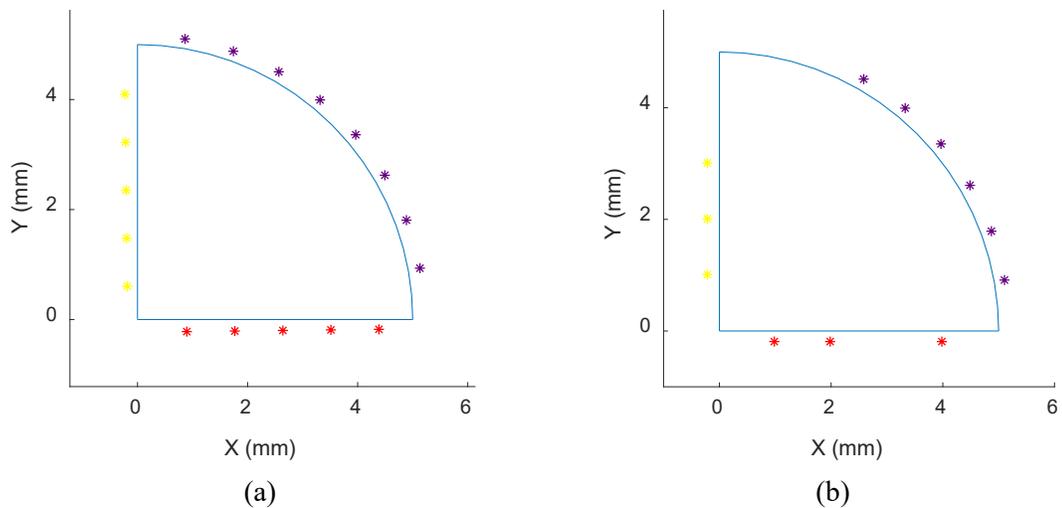


Figure 42. Two test samples with constant point deviations; (a) uniform point distribution; (b) non-uniform point distribution

It is expected that all optimum parameters in the first case are close to zero, and they have various values in the second sample. However, it is more reasonable to have equal optimum point deviations in the second sample as well as the first one. The computed optimum value of fitting parameters are presented in Table 1 and bar graphs of point deviations are shown in Figure 43. Various colors are used for points belonging to separate features to distinguish point deviations of each feature derived from fitting algorithms. These results declare that the minimum sensitivity to the uniformity of the point distribution belongs to the WLS algorithm, and the output of this algorithm has the minimum range of deviations. Surprisingly, the LS algorithm shows the highest

variation in fitting variables as well as the range of point deviations. It is comprehensive from point deviation graphs that the LS fitting method is biased to the feature with the highest number of points, and the result of the HWLS algorithm is nearly opposite of that. It seems that the weighting in the HWLS algorithm overwhelms the point deviation. Meanwhile, the weighting in the WLS algorithm is moderate, and this algorithm shows the most reasonable result for this case study.

Table 1. Optimum fitting parameters associated to uniform and non-uniform point distribution

Sample no.	Fitting algorithm	Theta (degree)	dx (mm)	dy (mm)
Sample 1 (uniform point distribution)	LS	-0.0230082e-08	-0.022352	-0.022353
	WLS	1.237376e-09	-0.000978	-0.000977
	HWLS	1.325025e-08	0.020513	0.020513
Sample 2 (non-uniform point distribution)	LS	-1.969307	-0.108625	0.088000
	WLS	-0.010660	-0.016893	0.010275
	HWLS	1.138457	0.060697	-0.062728

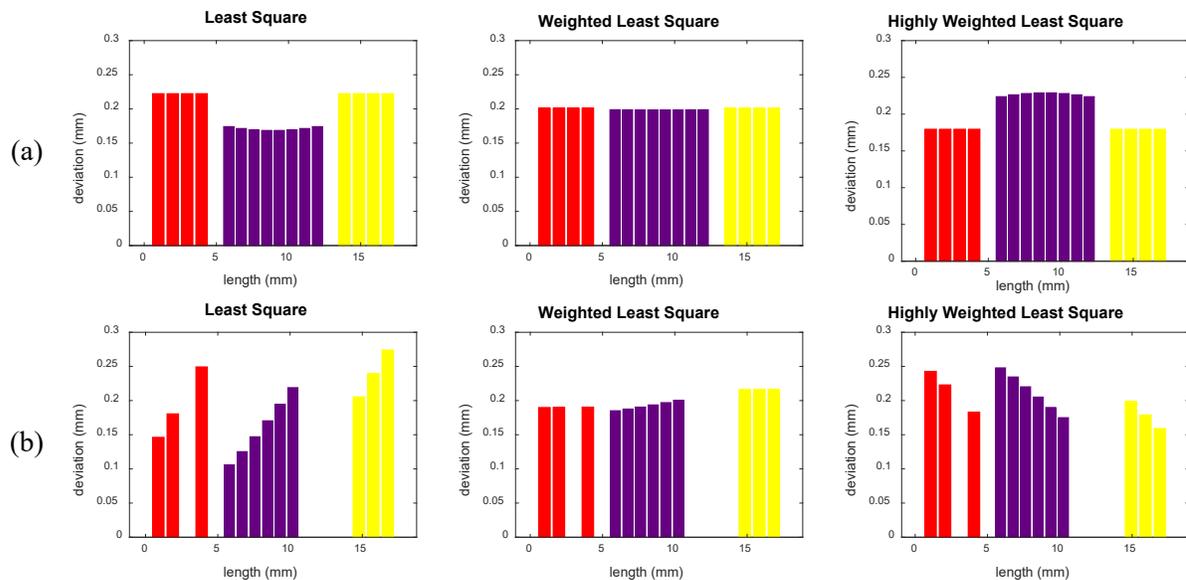


Figure 43. Graphs of point deviations vs. profile length associated to different fitting algorithms; (a) uniform and (b) non-uniform point distribution

To provide a general analysis, as mentioned in chapter 3, 18 random points with the standard deviation of 0.115 mm are simulated around the nominal quarter circle profile, and then these data sets are matched to the profile by using three introduced fitting algorithms. This procedure is repeated 100 times for the uniform and non-uniform point distributions, and the three

optimum fitting variables (θ , dx , and dy) obtained by each fitting algorithm are analyzed as well as the range of point deviations. Figure 44 and Figure 45 encompass the relative graph of fitting parameters. In these graphs, the horizontal axis is related to parameters obtained by the LS fitting, while the vertical axis shows values of optimum variables calculated by WLS and HWLS algorithms. It is obvious that in graphs of Figure 44 points are close to the line with the unit slope. It can be interpreted as when points have a uniform distribution, outputs of WLS and HWLS algorithms are close to outputs of the LS algorithm. It happens because weights of point deviations have lower variation in an organized structure of points. Oppositely, graphs of Figure 45 show a dispersion of points from the line with the slope of one, which means variables obtained by WLS and HWLS fitting algorithms are more different than parameters derived from the LS algorithm. In addition to that result, considering the range in which data vary clarifies that variations of WLS data are immensely close to variations of LS data sets, but variation ranges of parameters originated in the HWLS algorithm are abundantly wider than variation ranges of LS data sets.

For further perception, variables related to the LS algorithm are considered as reference and the difference of other obtained variables is calculated. Differences between WLS parameters and associated LS parameters (difference 1) are computed as well as differences between HWLS variables and reference LS variables (difference 2). Histograms of these differences are available in Figure 46 and Figure 47. Similarly, histograms prove that WLS parameters are closer to LS parameters in comparison to HWLS variables, and both differences are higher in the non-uniform point distribution condition.

For analyzing the range of point deviations (maximum deviation minus minimum deviation) the maximum and minimum deviations of every matching are saved, but histograms of ranges related to LS and WLS algorithms are close to each other, so it is decided to generate the CDF (Cumulative Distribution Function) of deviation ranges, and put CDFs of various algorithms

in one plot. The difference in the performance of these three algorithms will be more distinct in CDF plots. These CDF graphs are available in Figure 48.

From CDF graphs shown in Figure 48, it can be understood that the performance of three fitting algorithms is almost similar on uniform point samples; nonetheless, as it is expected, these algorithms ensue different results if point samples have a non-uniform structure. Surprisingly, the highest cumulative probability of the range of point deviations belongs to the LS algorithm, and it may be interpreted as the most desirable fitting, but it is in contrast with what was presented in the first part of this section. Therefore, more examinations are required for making a conclusion. Furthermore, it is important to state here that both of uniformly and normally random distribution of deviations are tested in simulations, but results of statistical analyses did not verify a clear difference between these two distribution models.

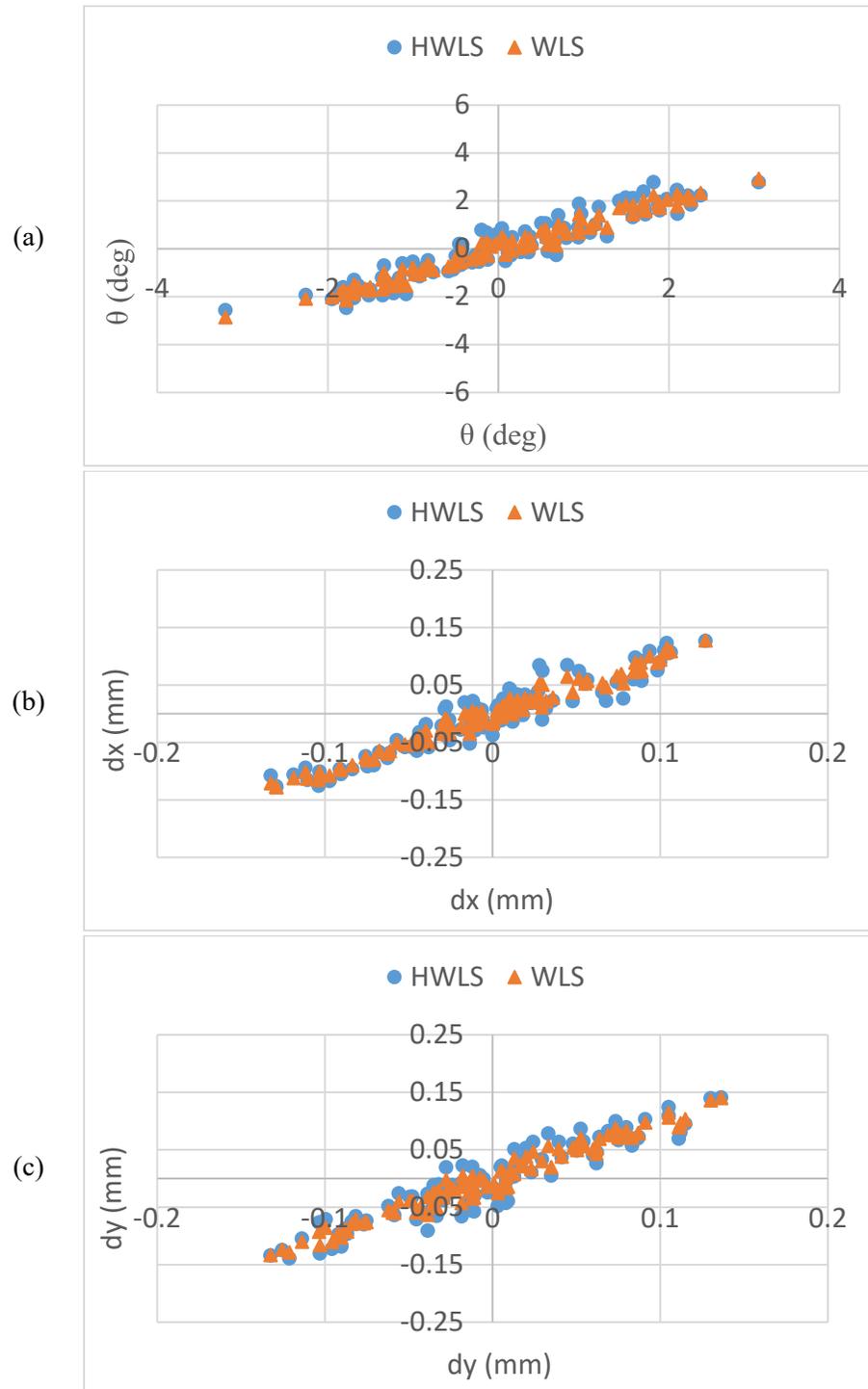


Figure 44. Relative graph of fitting parameters (a) rotation angle (b) translation along x (c) translation along y belonging to data with uniform structure

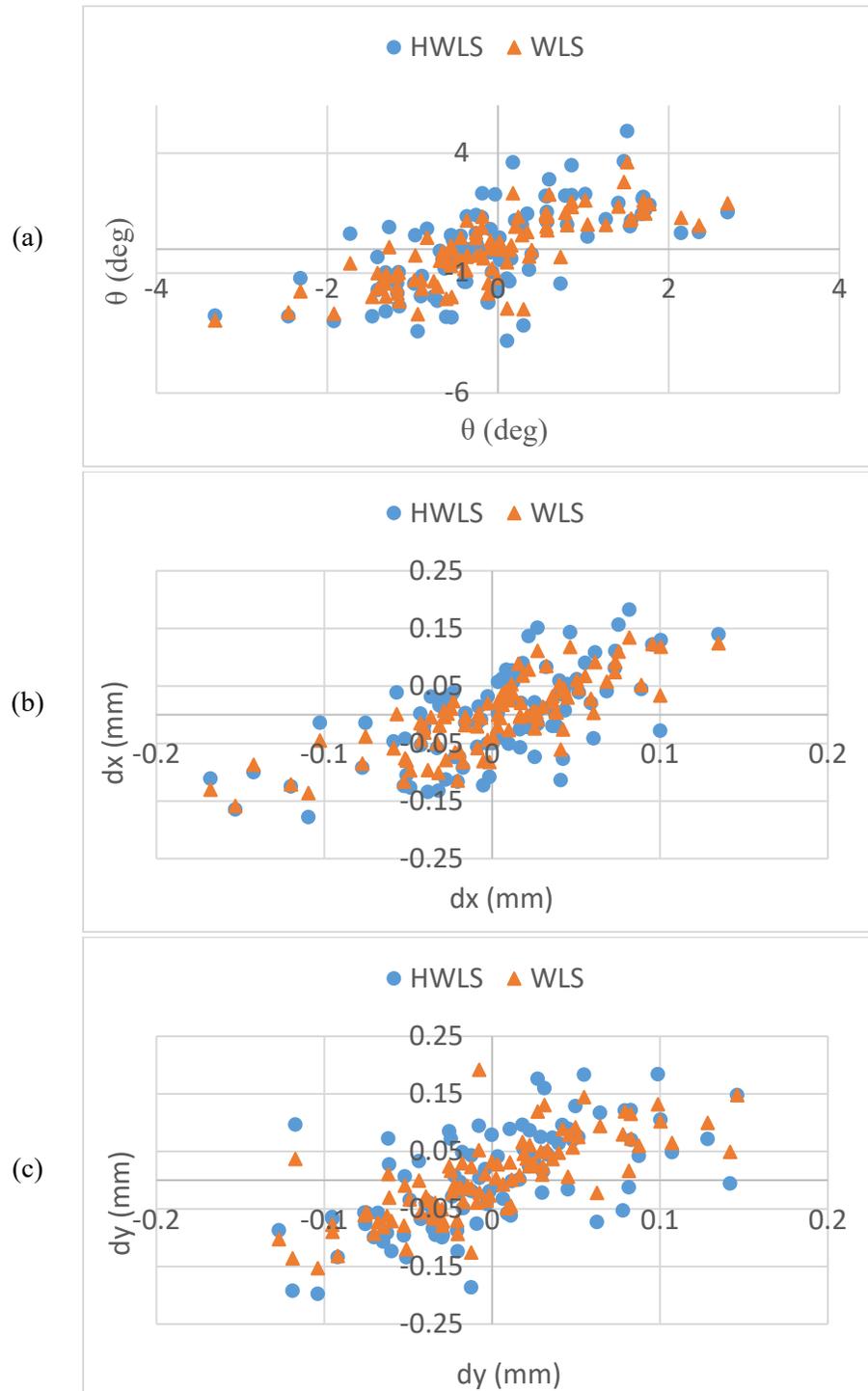


Figure 45. Relative graph of fitting parameters (a) rotation angle (b) translation along x (c) translation along y belonging to data with non-uniform structure

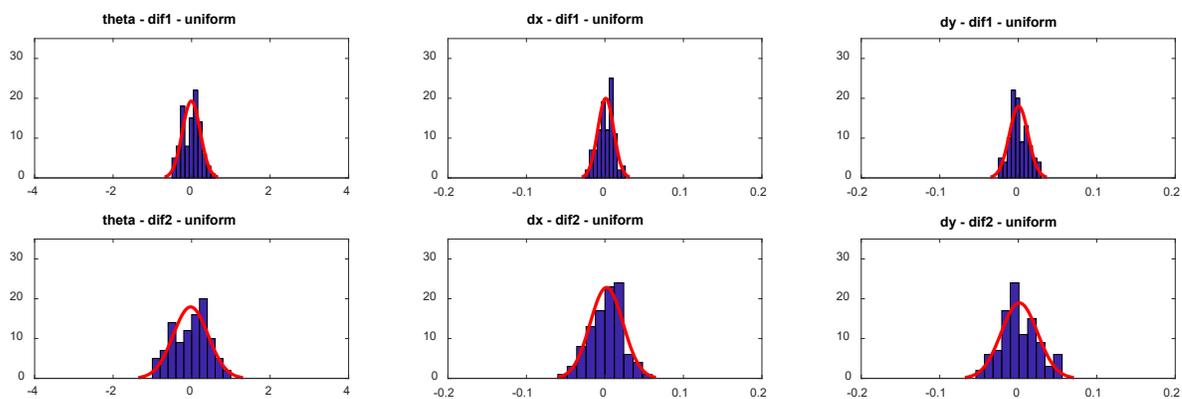


Figure 46. Histograms of differences in fitting parameters related to the uniform point distribution (vertical axes show the percentage of probability)

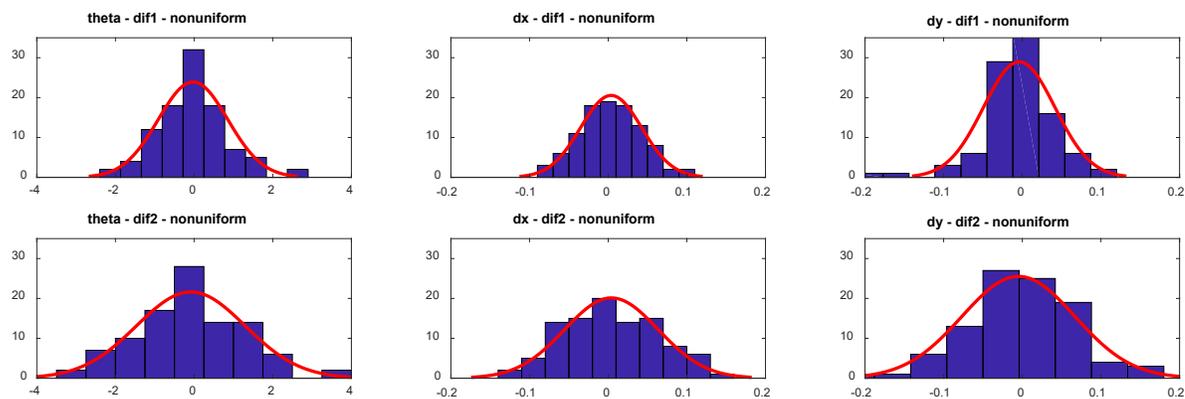


Figure 47. Histograms of differences in fitting parameters related to the non-uniform point distribution (vertical axes show the percentage of probability)

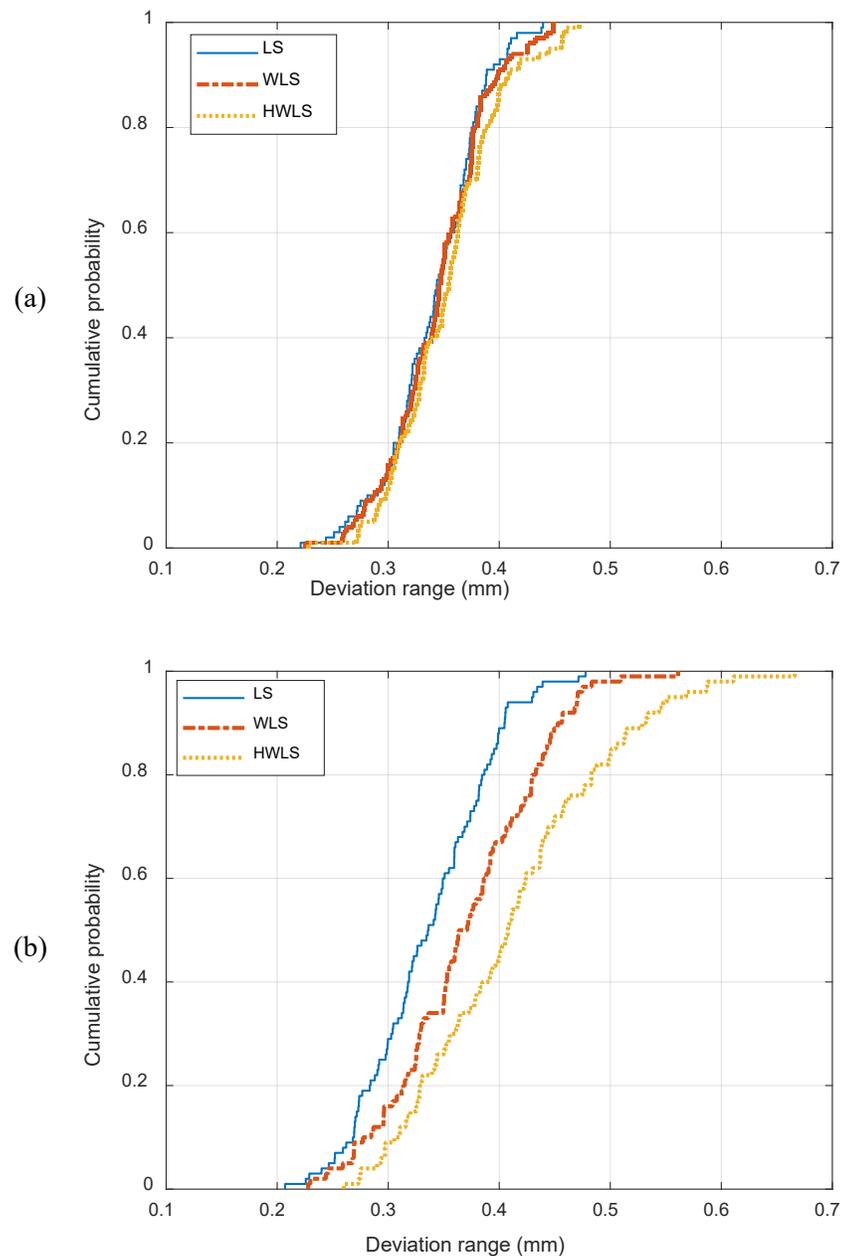


Figure 48. CDF plots of point deviation ranges obtained from various fitting algorithms (a) uniform point distribution; (b) non-uniform point distribution

5.1.3. Saddle profile

In the same procedure, multiple 3D point samples with random deviations from the nominal saddle surface are generated, and then they are matched to the profile surface by means of LS, WLS, and HWLS fitting algorithms. Since no datum constraints this matching process, so there are 6 variables associated to degrees of freedom, that need to be optimized. The horizontal axis in

graphs shown in Figure 49 and Figure 50 presents variables optimized by the LS algorithm, while values of optimum variables provided by WLS and HWLS algorithms are shown in the vertical axis. Similar to results of the quarter circle profile, in uniform point distribution, both types of data are close to the line with unit slope, but data points related to the HWLS algorithm are farther than WLS points. The same relation exists between three fitting algorithms on non-uniform point samples. However, there is a wider dispersion of data points in graphs related to the non-organized point distributions. Additionally, the relation between the range of variables seems interesting. In point samples with uniform structure, the variation range of fitting variables originated in WLS and HWLS algorithms is close to the range of LS data sets, while these variation ranges are larger on point clouds with unorganized arrangement.

To provide a more clear mathematical description, LS data sets are assumed as reference and the difference of WLS and HWLS data with LS data is calculated. The prior difference has number 1 (difference 1) in graphs and the latter one is shown with number 2 (difference 2). Histograms of differences in optimized data are presented in Figure 51 and Figure 52. Again, it is verified that results of HWLS algorithm have higher difference than parameters obtained by the WLS algorithm. Furthermore, it is comprehensive that both defined differences are smaller when the point cloud has a uniform structure.

Similar to the quarter circle profile, maximum and minimum point deviations were used to generate the CDF plot of deviation ranges. Resulted graphs are similar to graphs shown in Figure 48, so for further investigation another set of simulations is performed with a more narrow range of point deviations (point deviations within ± 0.05 mm). The CDF plot associated to the new analysis (Figure 53) does not declare any noticeable different result.

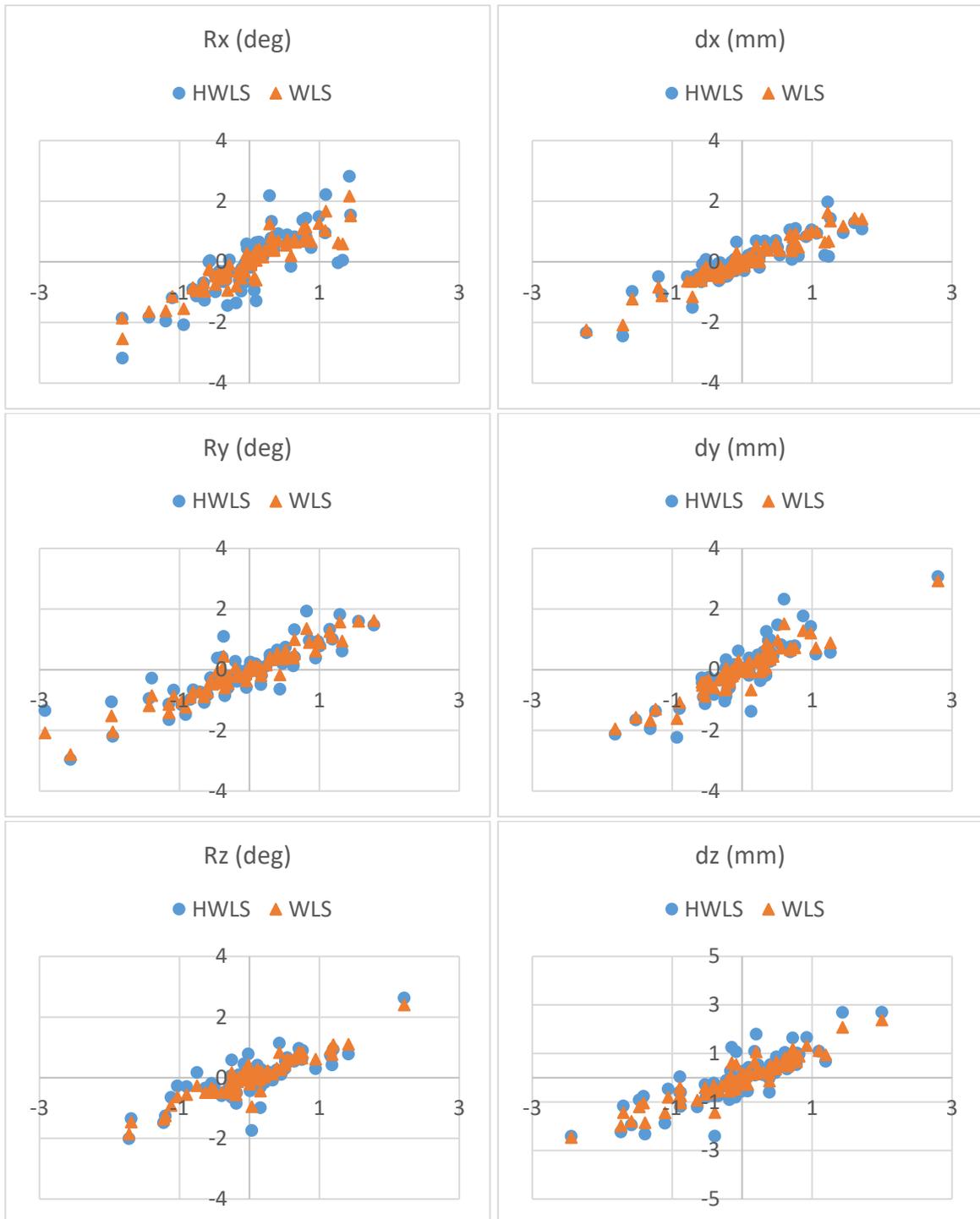


Figure 49. Relative graphs of optimum fitting parameters obtained by applying various algorithms to uniform point samples

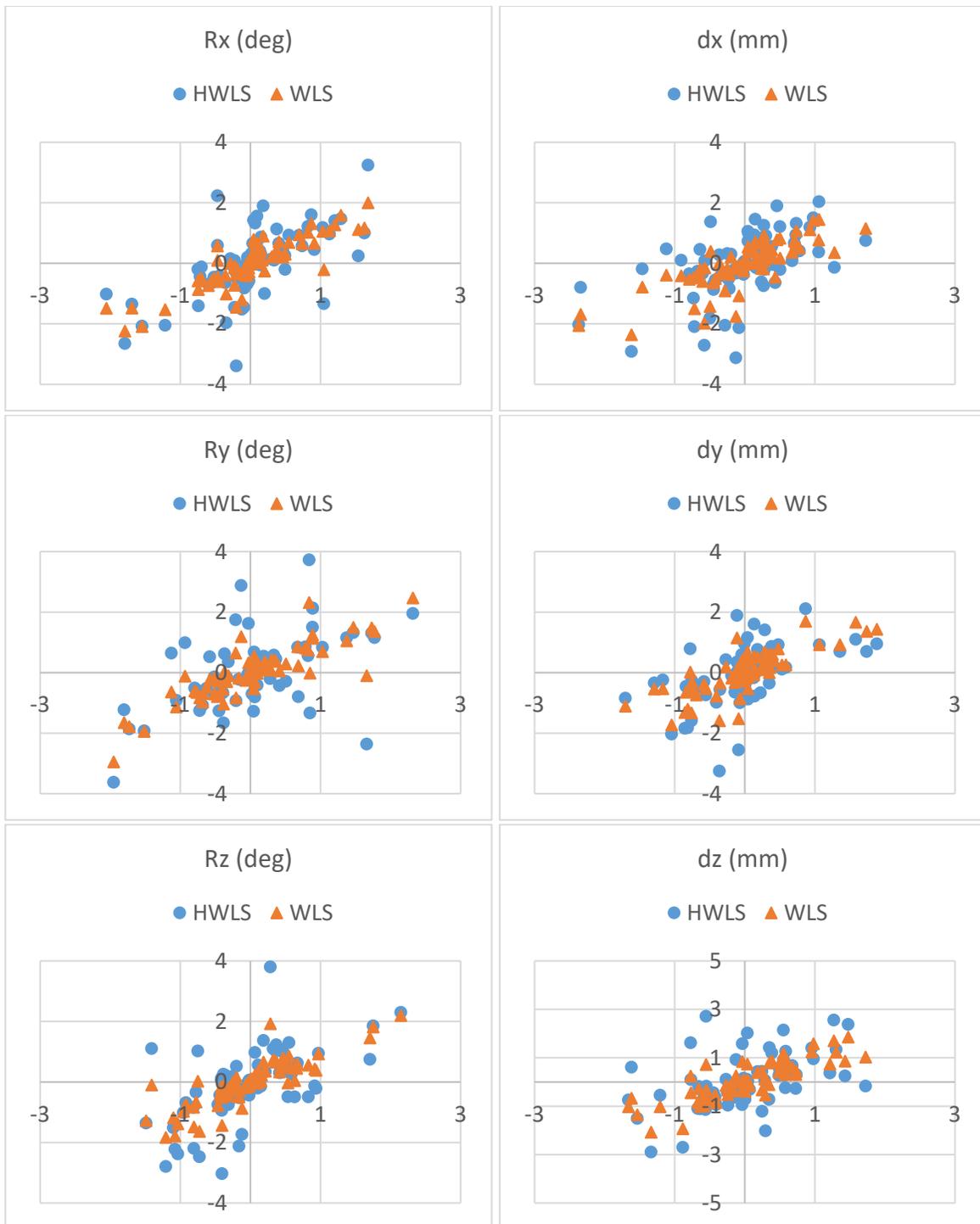


Figure 50. Relative graphs of optimum fitting parameters obtained by applying various algorithms to non-uniform point samples

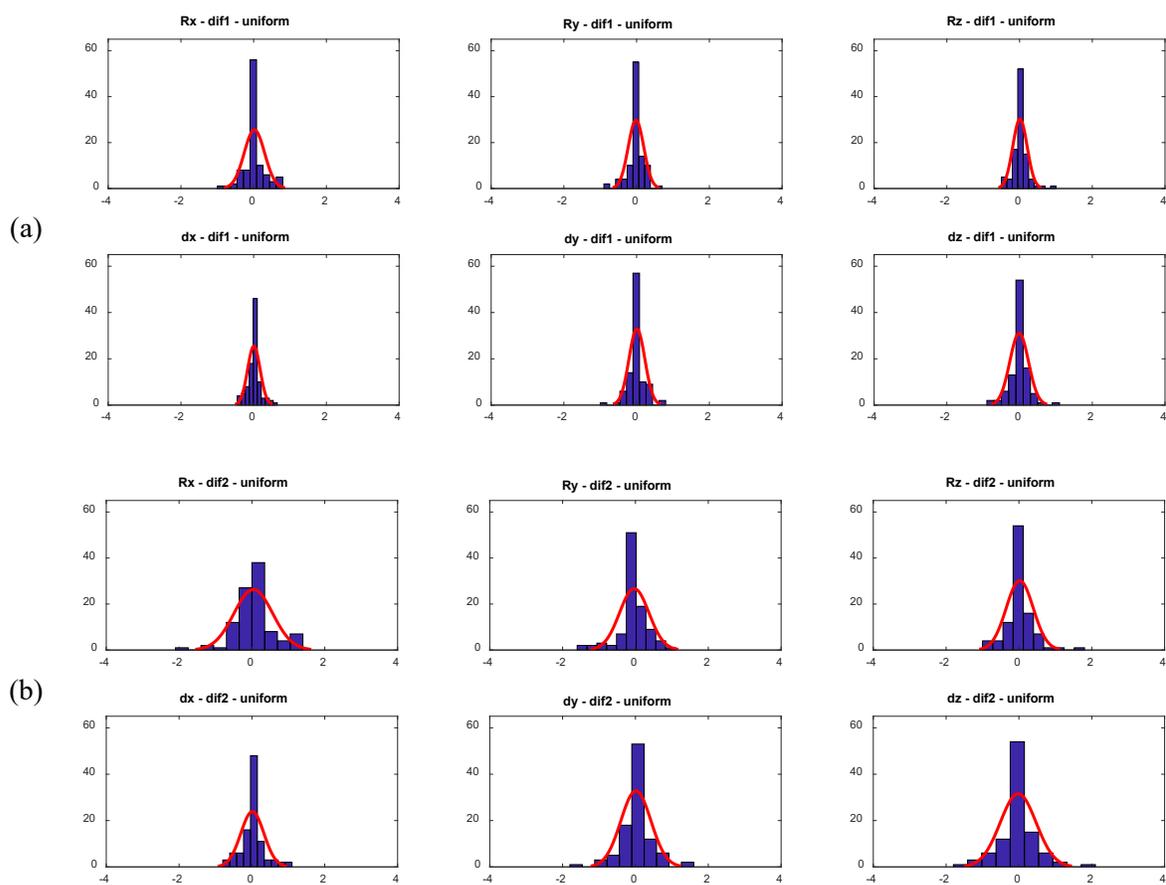


Figure 51. Histograms of differences in fitting parameters related to the uniform point distribution over the saddle profile; (a) difference between WLS and LS; (b) difference between HWLS and LS (vertical axes show the percentage of probability)

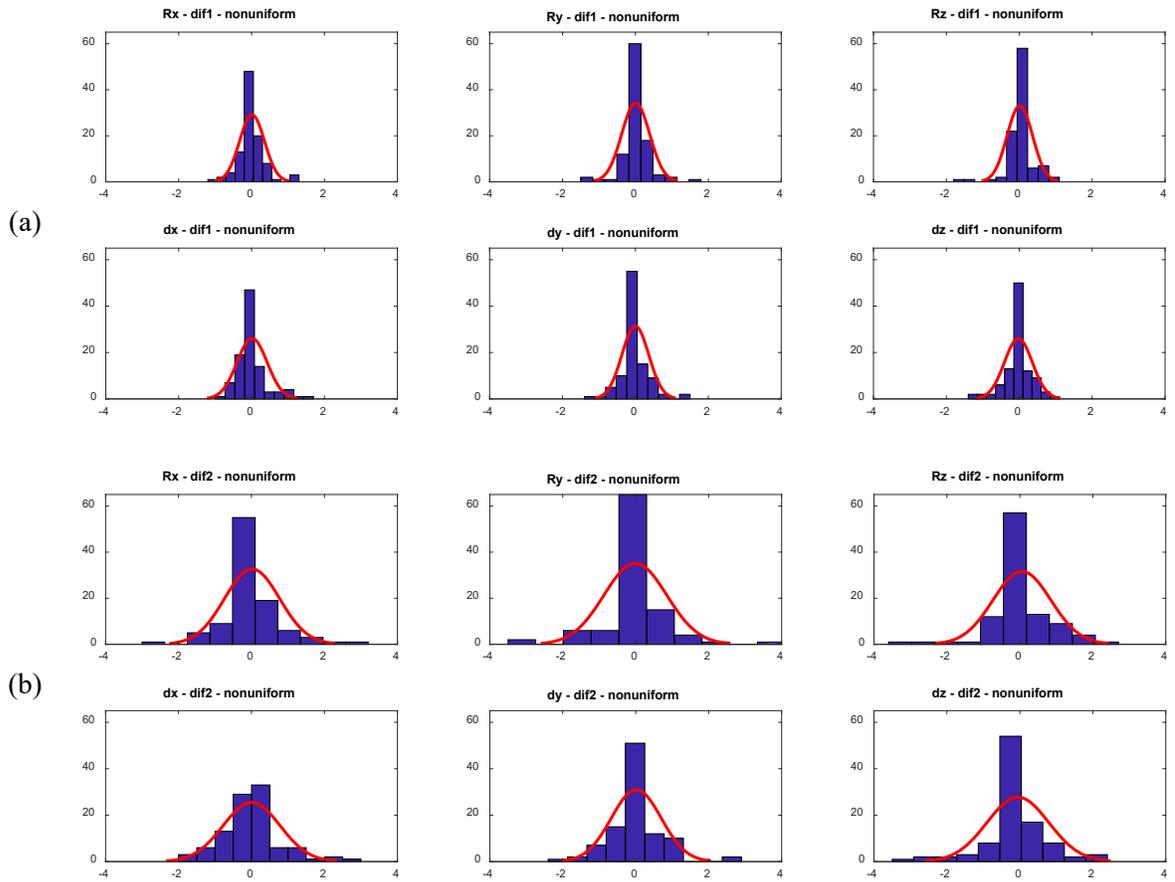


Figure 52. Histograms of differences in fitting parameters related to the non-uniform point distribution over the saddle profile (a) difference between WLS and LS; (b) difference between HWLS and LS (vertical axes show the percentage of probability)

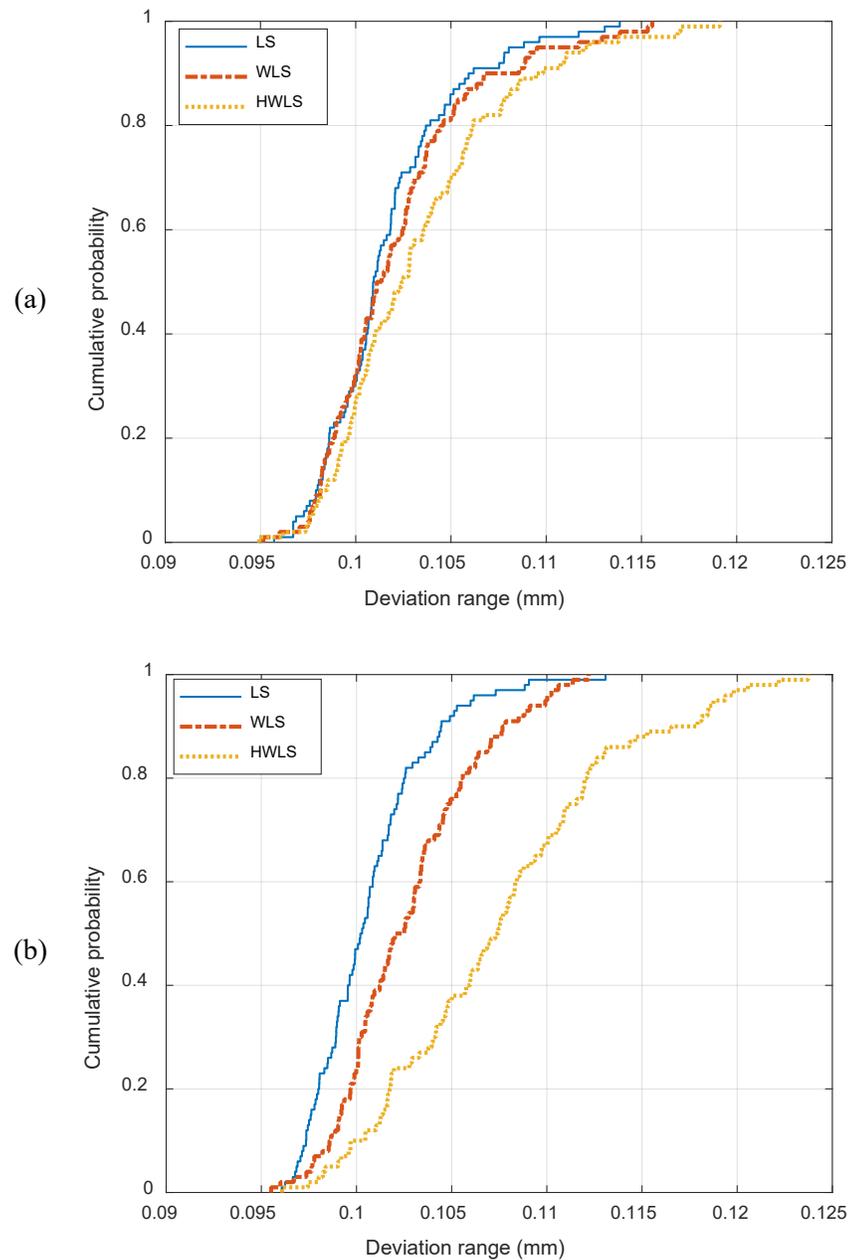


Figure 53. CDF plots of point deviation ranges obtained from various fitting algorithms (a) uniform point distribution; (b) non-uniform point distribution

5.2. Results of experimental measurements

5.2.1. Measurement and analysis of 2D profiles

Part profiles described in chapter 3 are scanned with various tactile and optical methods, and then the measured data sets are matched to their associated nominal 2D profiles by means of LS, WLS, and HWLS fitting algorithms. To compare these matched data sets with each other, they are quantified by the method described in chapter 4, and then MLDs originated in WLS and HWLS fitting algorithms are compared with MLDs obtained by the LS algorithm. As an example, the difference between two quantified data sets, related to the measurement of the slot profile (profile 2) with the optical method, that are shown in Figure 54 is presented in Figure 55.

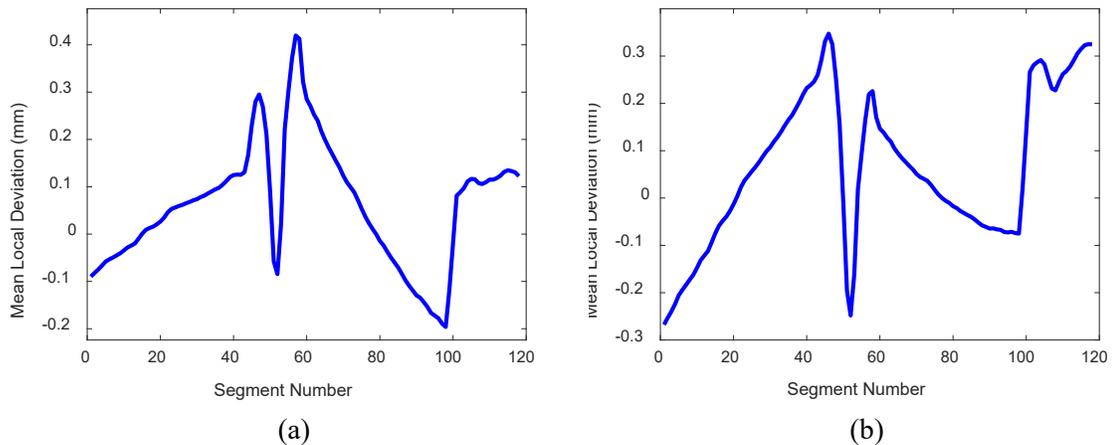


Figure 54. MLD graphs of the optically measured slot profile (profile 2) fitted by (a) LS algorithm; (b) WLS algorithm

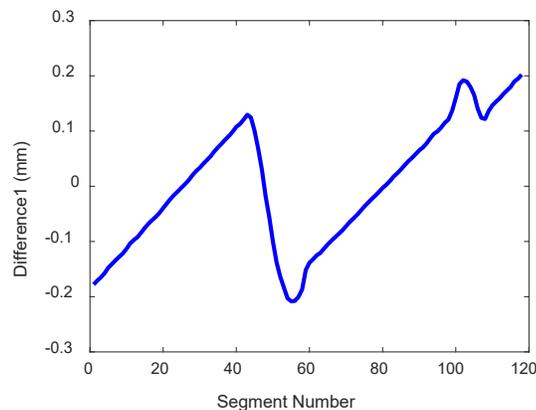


Figure 55. Difference between MLDs originated in WLS and LS fitting algorithms

It is clear that if points are distributed equally around the profile length then three algorithms would result in the same local deviation graph and their differences would be zero (because all points would have the same weight). In the experimental evaluation of repeatability, measurements are repeated eight to ten times for every sample part and with every instrument to realize how much the difference range (the maximum difference minus the minimum one) is in MLDs. Results declare that the most common range of differences that has the most occurrence on data from different instruments is about ± 200 micrometers. Furthermore, the range of difference 2 (difference between MLDs obtained by LS and HWLS) is generally higher than the range of difference 1 (difference between MLDs obtained by LS and WLS). It should be stated that the initial analysis with a default value for the target stop limit ($TolX= 10^{-4}$) shows the minimum possible range of difference in tactile data can be as low as ± 2 micrometers, while the minimum obtained range of difference for optical data is about ± 40 micrometers. However, decrease of the target stop limit to 10^{-12} makes the range related to tactile data close to the range obtained from optical data, while because of the extensive computation load the accessible computer cannot process optical data with this tiny target stop limit. Minimum ranges obtained from tactile and optical data during the initial analysis are shown in Figure 56.

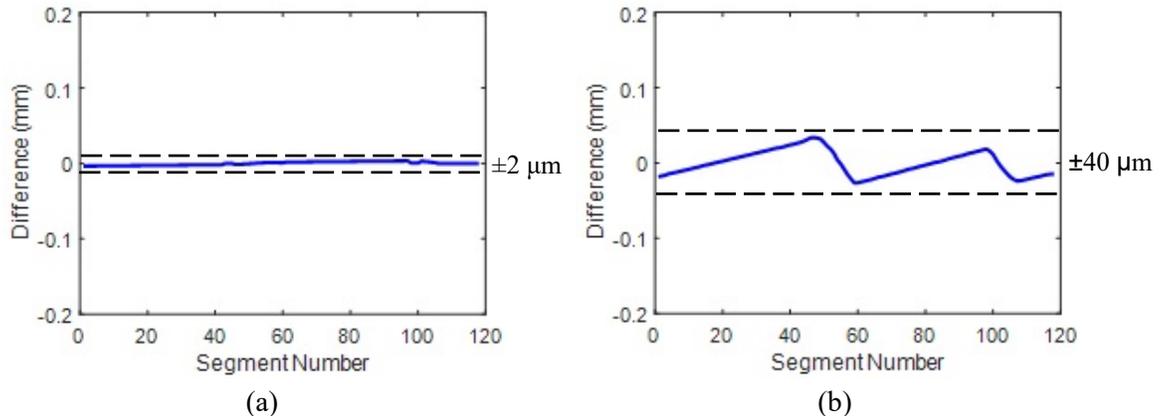


Figure 56. The lowest range of the difference in MLDs obtained from (a) tactile and (b) optical sensor on the Romer arm instrument

5.2.2. Measurement and analysis of 3D profiles

The wavy profile is measured with various instruments mentioned in chapter 3. Due to the high number of measured points in optical scans (100,000 to 350,000 points) the accessible computer cannot process all points, so a subset of measured data is selected randomly, and the selected subset is matched to the nominal profile by means of LS, WLS, and HWLS algorithms. For example, a set of optically measured data related to the scan of the wavy profile with the Romer arm has 130,068 points, and about 1000 randomly selected points are fitted by the three algorithms. Figure 57 presents normal deviations of selected points at the end of fitting procedures. It can be understood that point deviations are nearly constant at profile segments, and their coordinates in deviation graphs differ within a small range (less than 10 micrometers) between the fitting algorithms. This constant local deviation that is noticed even between various measurement methods is the main motivation for using MLDs instead of individual point deviations.

The quantification process is performed on individual point deviations obtained from different fitting algorithms. MLD graphs associated to data shown in Figure 57 are presented in Figure 58. It is obvious that the quantification procedure can decrease the size of spikes and drops in measured data; the range of MLD data is lower than the range of point deviations. There are three values for every profile segment. Comparing MLD data obtained from WLS and HWLS algorithms with MLD data originated in the LS algorithm may be helpful for understanding the influence of the non-uniformity of point clouds. For this purpose, the quantified data set related to the LS algorithm is considered as reference, and differences of other two data sets with this data set are calculated and shown in Figure 59. The front view of presented graphs declare that there is more dispersion in data of difference2, which is equal to higher difference values between MLDs derived from LS and HWLS algorithms. This result has congruence with results obtained from theoretical simulations.

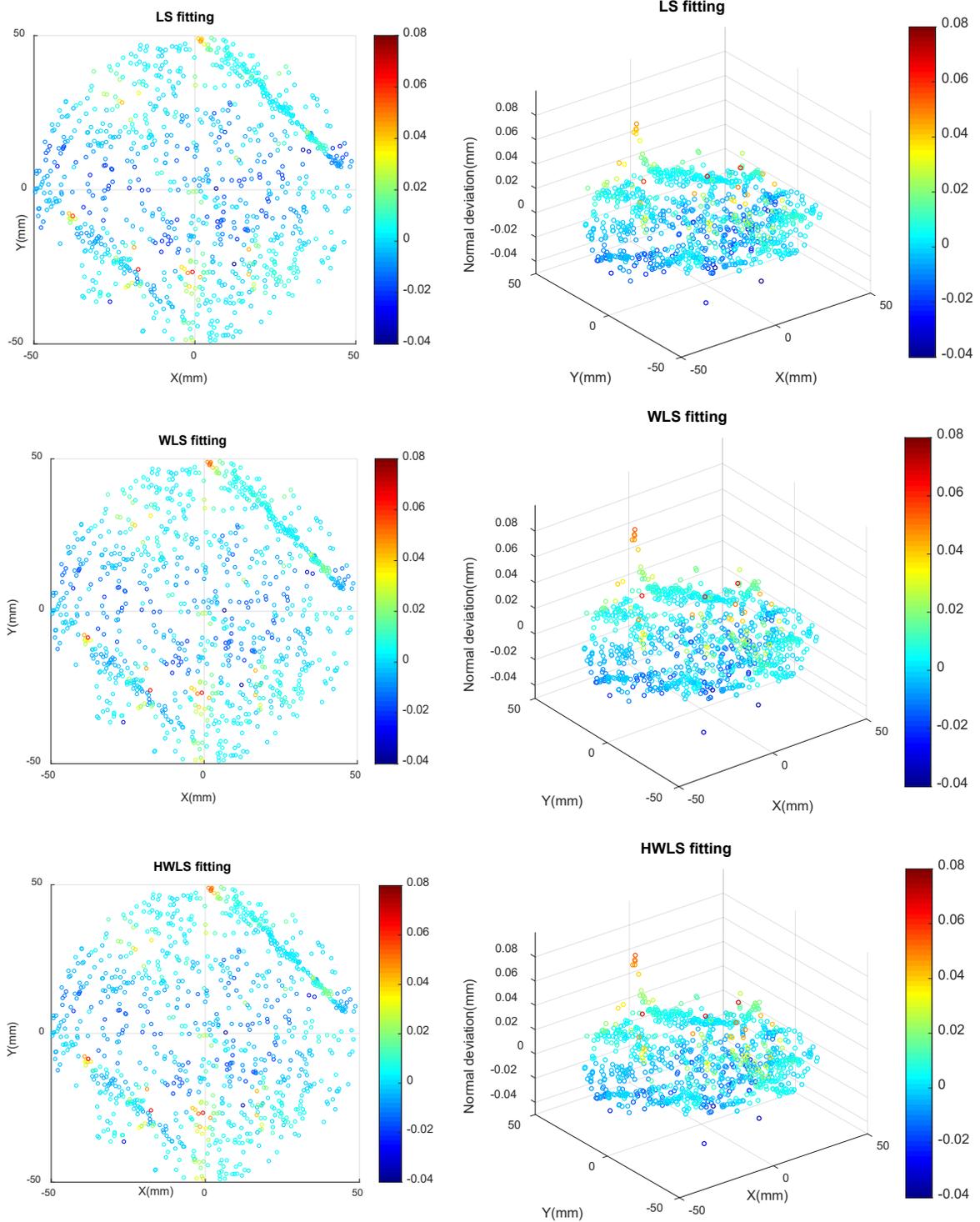


Figure 57. Individual point normal deviations from the wavy profile originated in measurement with the optical sensor on the Romer arm; (left column: top view; right column: isometric view)

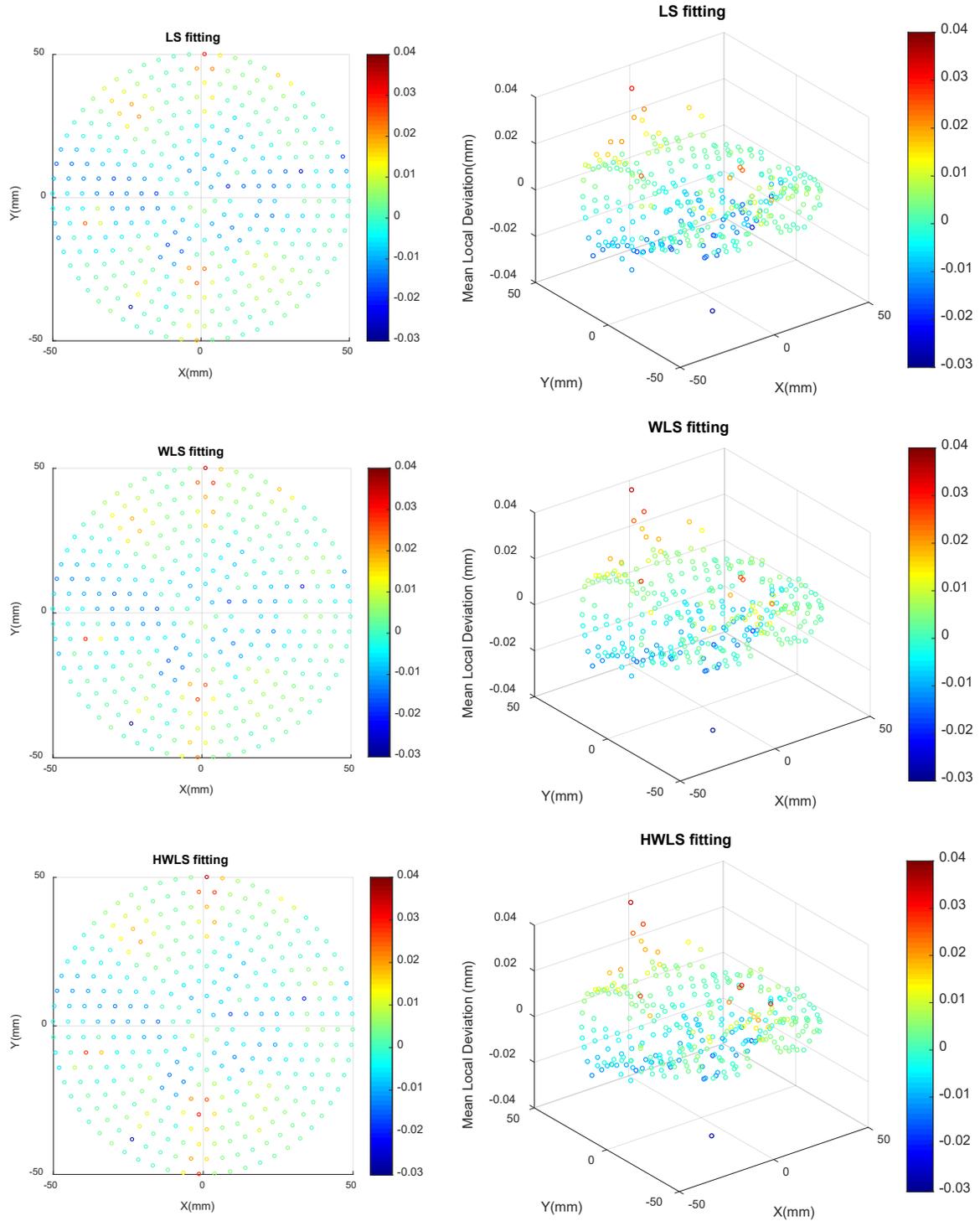


Figure 58. MLDs on segments of the wavy profile originated in measurement with the optical sensor on the Romer arm; (left column: top view; right column: isometric view)

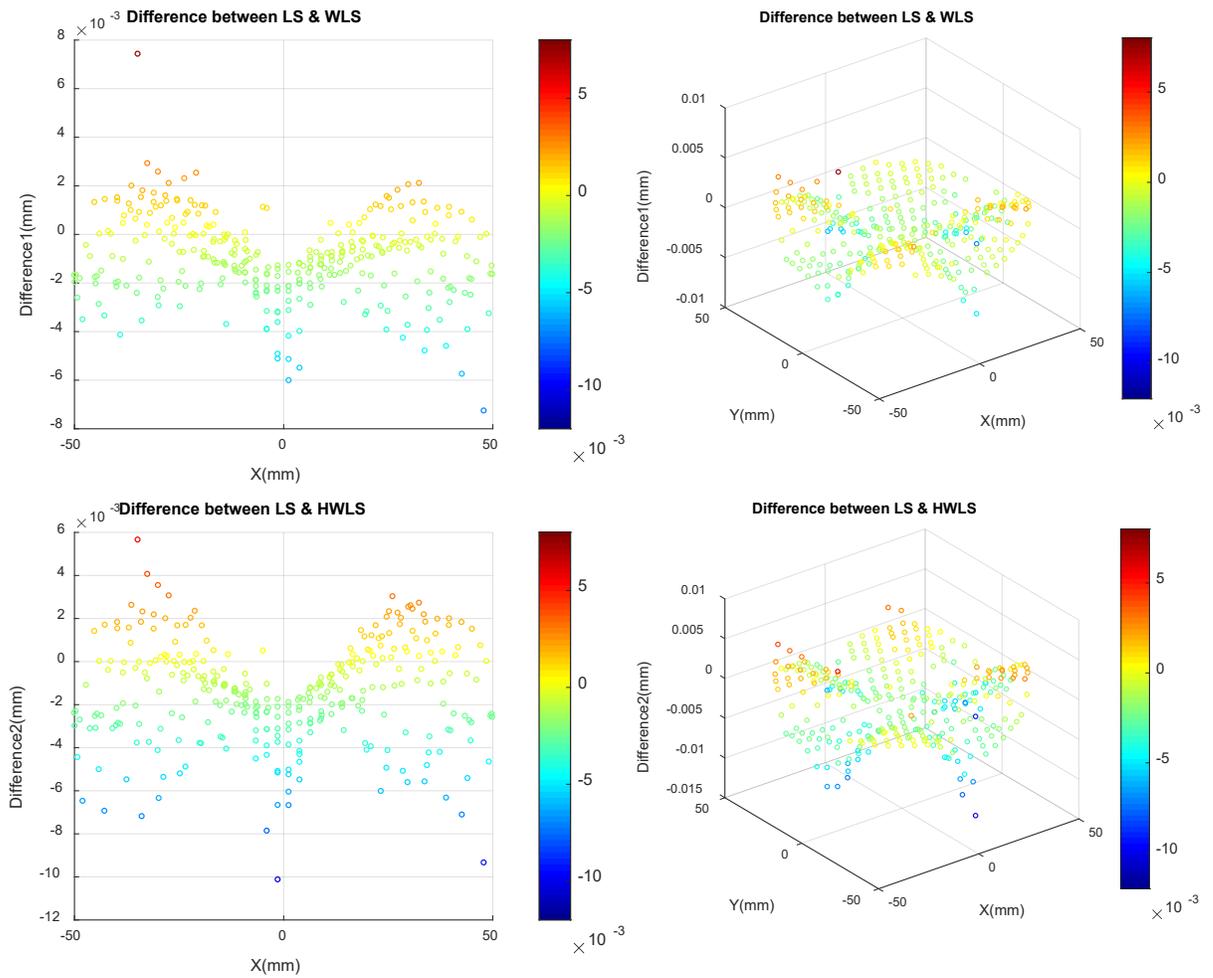


Figure 59. Differences of MLD data originated in measurement with the optical sensor on the Romer arm (left column: front view; right column: isometric view)

For further investigation, the same analysis is executed on data measured with the tactile method. The number of measured points is 7,697, while there are 328 segments on the profile. Therefore, averagely there are about 23 points in every profile segment. Since data sets obtained from three introduced fitting algorithms are extensively close to each other, and there is a tiny difference between them, here only one of those graphs is shown. Point deviations and MLDs of profile segments derived from the WLS algorithm are presented in Figure 60. Comparing these two types of data to the same type of data from the optical measurement shows the range of tactile data is lower than optical data, although the general shape of deviation plots is similar. Once more, MLD data obtained from LS fitting is defined as the reference, and the difference of MLDs derived from WLS and HWLS fitting algorithms is calculated. The related graphs of difference 1 and difference 2 are shown in Figure 61. The analogy of these graphs with graphs shown in Figure 59 declare that the differences obtained from tactile data are lower, and it is because of the uniform structure of tactile data. The range of differences obtained from the optical scan is nearly 5 times larger than the range of differences related to the tactile measurement. Furthermore, as it is expected, the difference between MLDs originated in LS and WLS algorithms is lower than associated data derived from LS and HWLS algorithms. However, due to the tiny variation of deviation weights that is a result of uniform point distribution, MLD data of tactile measurement resulted from three algorithms are significantly close.

To understand the effect of spraying, the wavy part is scanned with the blue light scanner once without any coating and the other time its top surface is treated with the Miconazole nitrate spray before the scan. The number of measured points in the first condition is 166,441, while the second scan generates 226,272 points. After the denoising procedure is applied to these two data sets (by using *pcdenoise* function) their point numbers became respectively 141,534 and 192,618. About 10% of each data set is randomly selected and matched to the nominal wavy profile. Besides, due to the high number of points the target stop limit in the optimization procedure has the default

value of 10^{-4} . Point deviations obtained from the LS fitting algorithm are shown in Figure 62. The introduced quantification method provides this opportunity to compare these two measured data sets and to evaluate the coating layer's effect. The difference between two measured data sets is presented in Figure 63. It is comprehensive that the sprayed coating can cause a maximum variation of ± 6 micrometers in the MLD of profile segments.

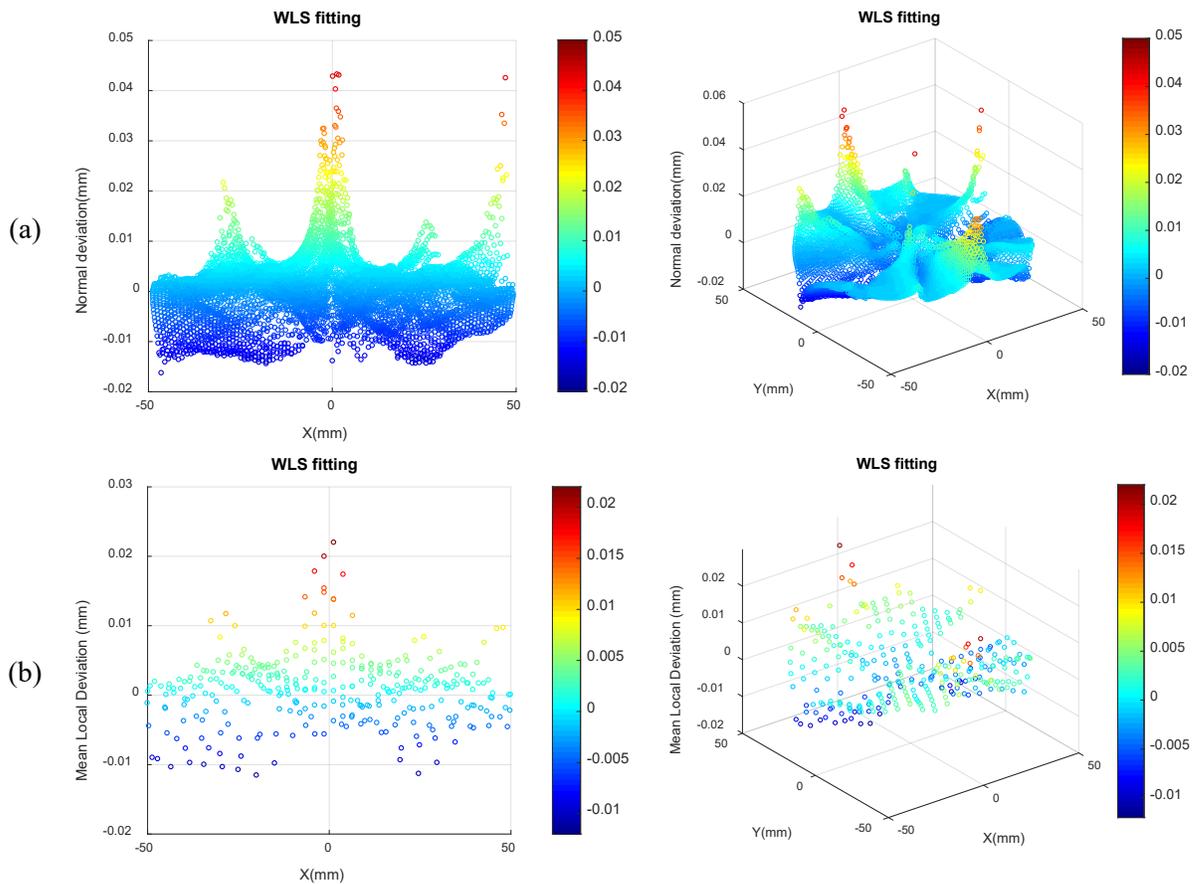


Figure 60. (a) individual point deviations and (b) MLDs obtained from tactile measurement of the wavy profile (left column: front view; right column: isometric view)

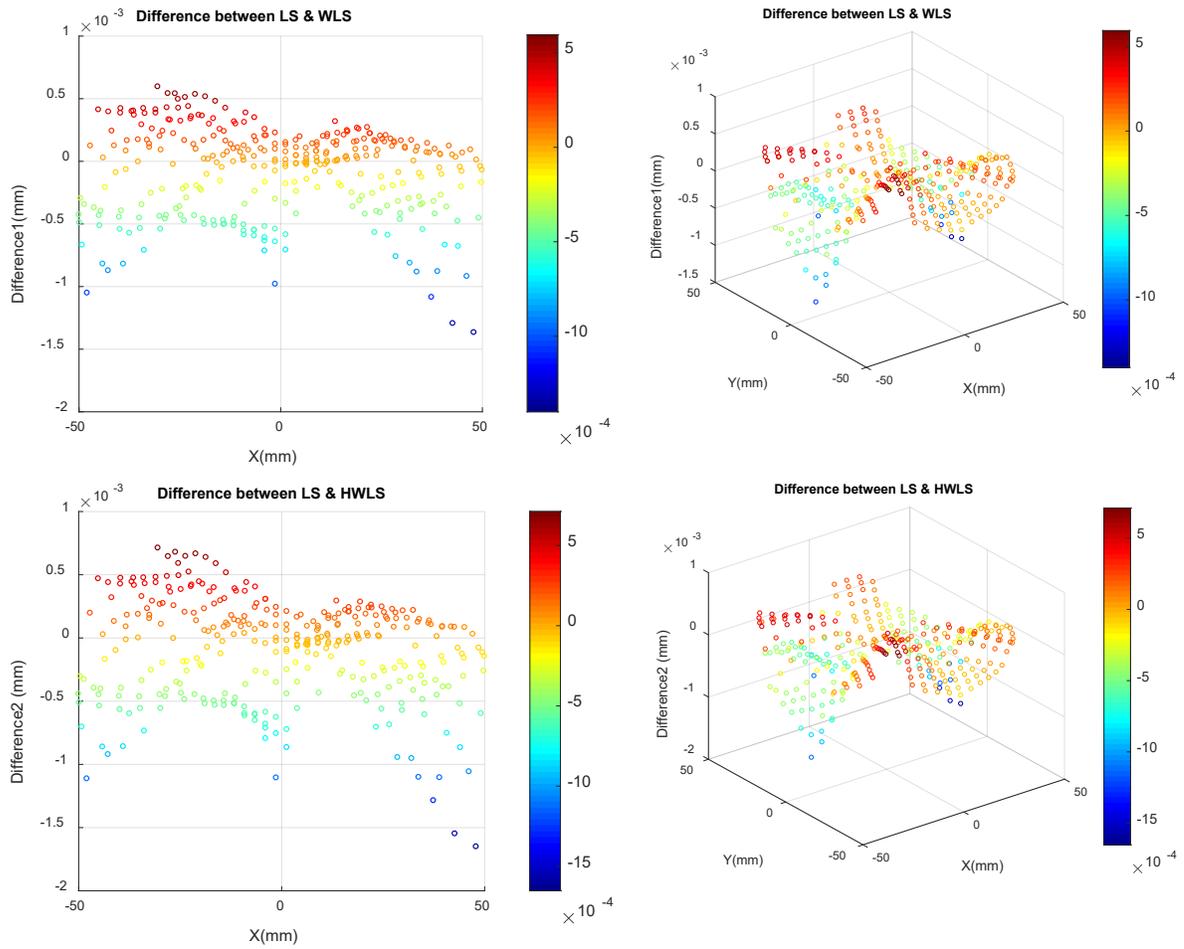


Figure 61. Differences of MLD data originated in measurement with the tactile CMM (left column: front view; right column: isometric view)

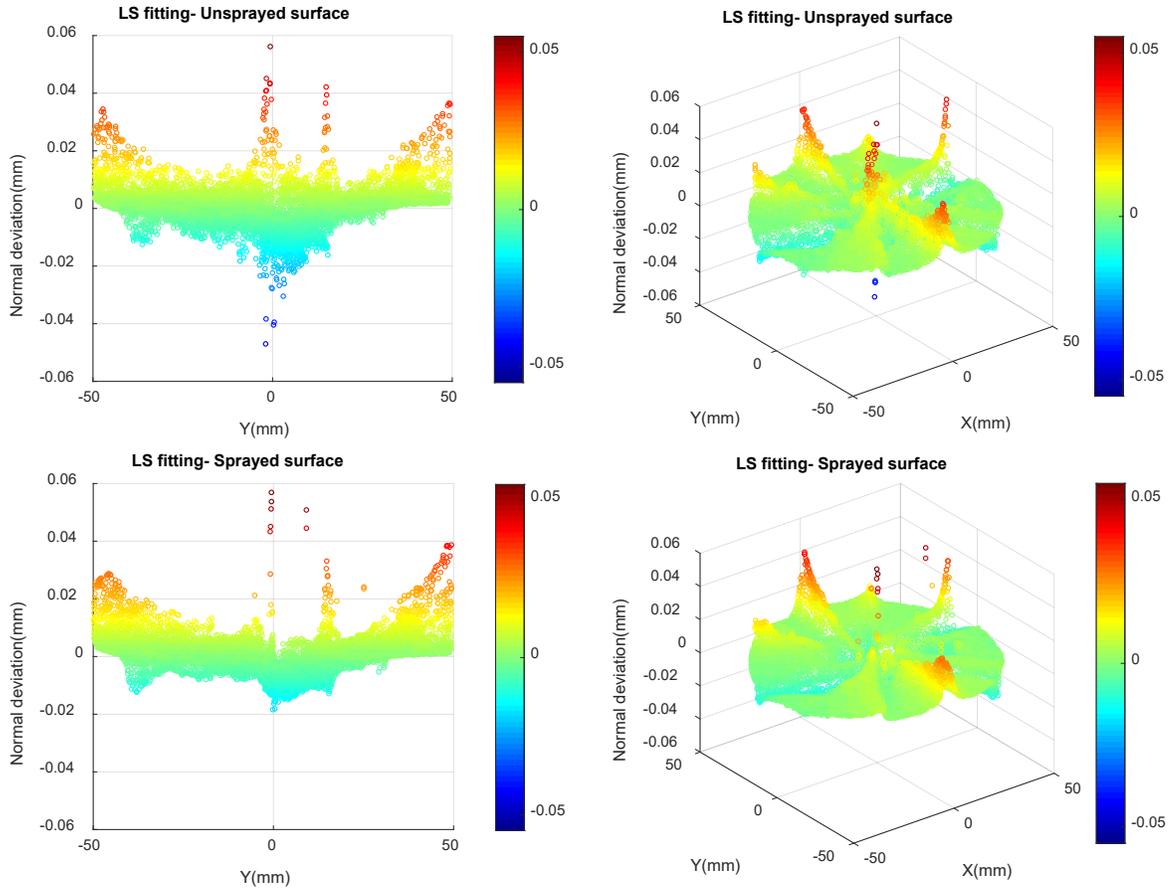


Figure 62. Individual point deviations resulted from scan of coated and uncoated surface (left column: side view; right column: isometric view)

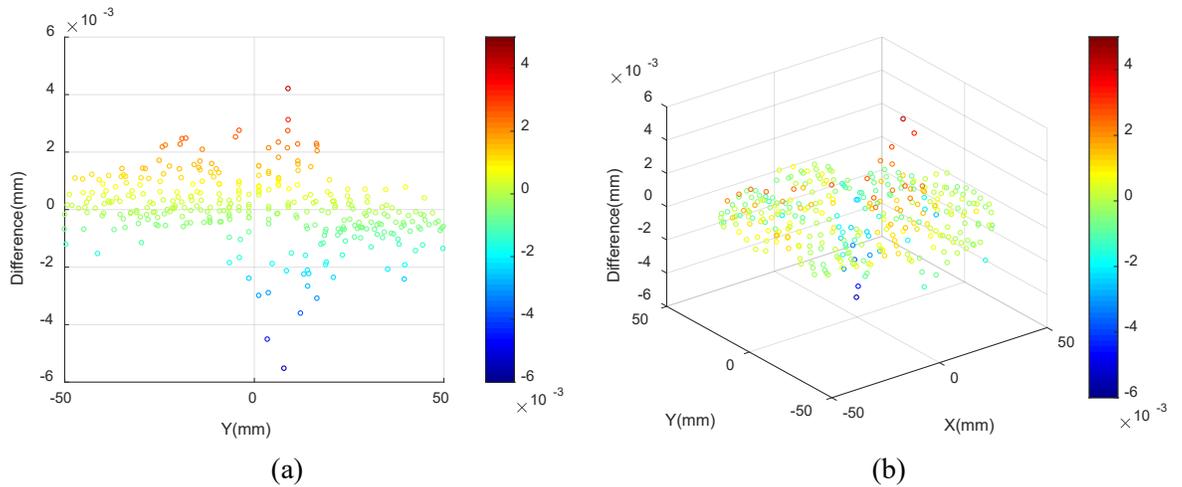


Figure 63. Difference between MLDs obtained from measurement of the sprayed wavy surface and the surface with no coating (a) side view; (b) isometric view

The saddle profile is scanned with the Romer arm and the Atos blue light scanner separately. Initial results of fitted data derived from the Atos instrument (Figure 64) show that reference stickers cause outside deviations, so those stickers are removed, and that group of scans is repeated with reference stickers on the optical table around the part.

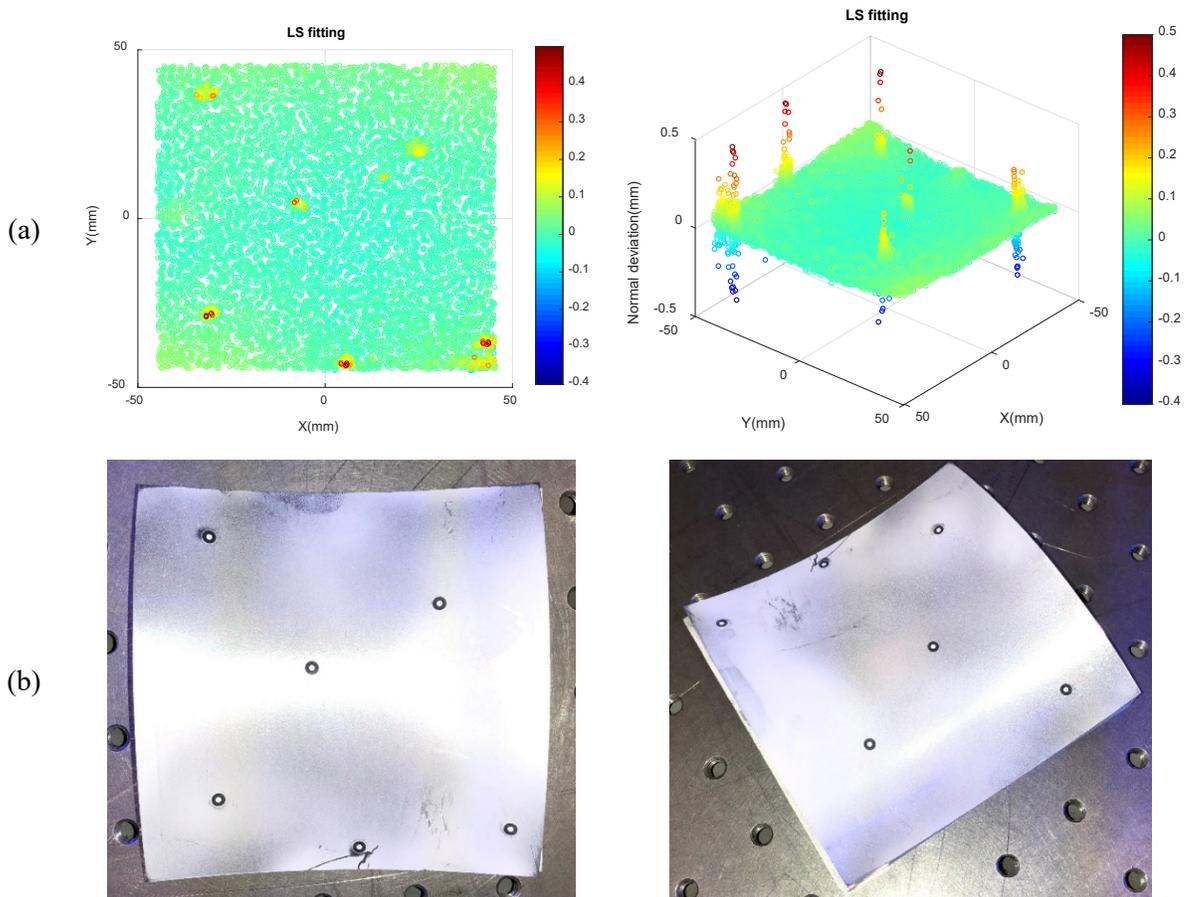


Figure 64. High deviation from the profile at the location of reference stickers
 (a) point deviations after fitting by LS algorithm; (b) photo of the scanned part (left column: top view; right column: 3D view)

The number of measured points, after that edge zones are cropped out, in scan with the Atos camera varies between 62,000 and 65,000, while the point density in scan with the Romer arm is significantly higher, and it depends on the light exposure time. Results of multiple scans show that the number of points, for this size of surface, is nearly from 160,000 to 350,000. From measured points, in all analyses, about 3,000 points are randomly selected, and the selected subset

is fitted to the nominal profile by using introduced LS, WLS, and HWLS algorithms. The saddle part is scanned 5 times with the Atos camera, and 5 subsets are selected from every measured data set. Therefore, 25 various values for each fitting parameter related to every algorithm will be obtained. Besides, by the projection of a grid of points onto the polygonised surface that was described in chapter 3, 5 samples with uniform structure can be created. Fitting these uniform samples using every algorithm will produce 5 values for each fitting parameter. Relative graphs of fitting variables are shown in Figure 65 and Figure 66. In these graphs, the parameter obtained from LS algorithm is shown on the horizontal axis, and fitting parameters resulted from WLS and HWLS algorithms are presented on the vertical axis. Similar to the simulated data, measured data show that fitting parameters derived from the WLS algorithm, in both uniform and non-uniform point distribution, are closer to the parameters originated in the LS algorithm. For better understanding, the defined differences in the fitting parameters are calculated, and the histogram of each variable is generated (Figure 67). Since the number of structured samples is not high enough, it is not reasonable to create the histogram of fitting variables for this type of point distribution. From the analogy between optimum fitting parameters obtained from simulations and experimental measurements, it is comprehensive that the range of Rz and dz is noticeably lower with experimental data. The main reason of this difference in the range of obtained fitting parameters should be in the dissimilarity between the assumed standard deviation and the standard deviation of real measured data. The distribution of point deviations and the effect of fitting algorithms will be discussed in detail in the next section of this chapter.



Figure 65. Relative graphs of optimum fitting parameters obtained by applying various algorithms to randomly selected samples with non-uniform structure

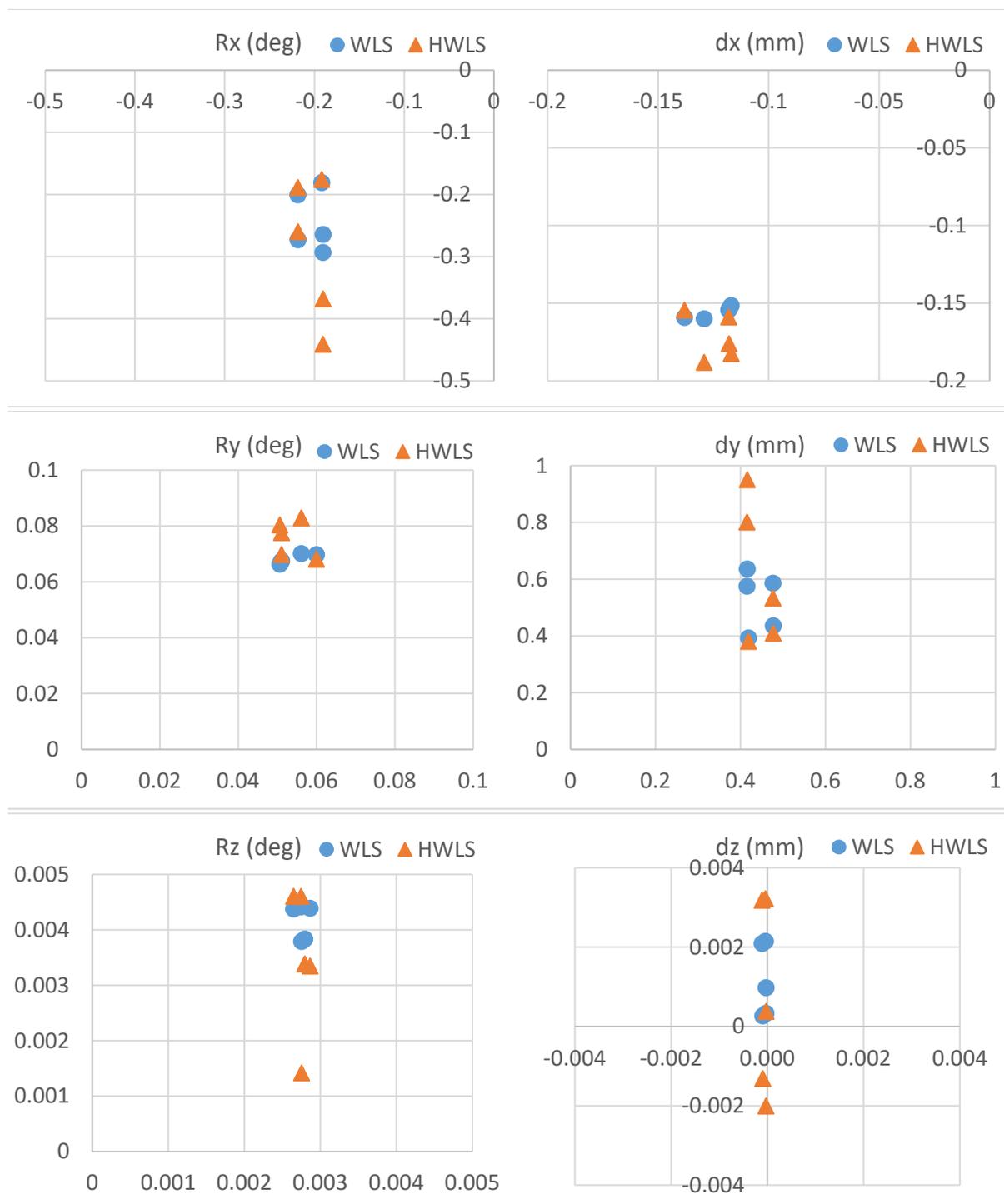


Figure 66. Relative graphs of optimum fitting parameters obtained by applying various algorithms to samples with uniform structure

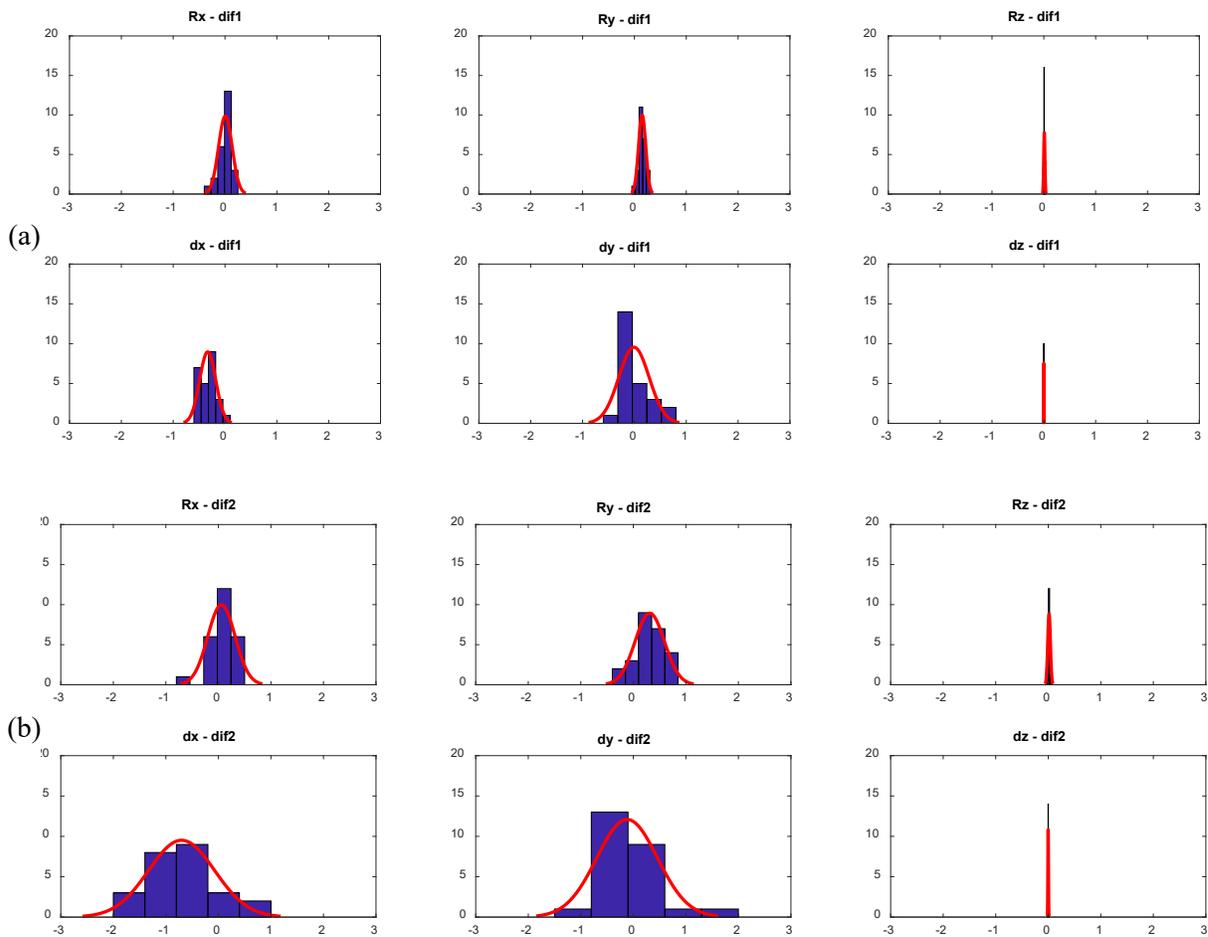


Figure 67. Histograms of differences in fitting parameters related to randomly selected samples with non-uniform point distribution (a) difference between WLS and LS; (b) difference between HWLS and LS

5.2.2.1. Statistical analysis of point deviations

It was described that about 3,000 points are randomly selected from scanned data, and these selected points are matched to the nominal saddle profile using LS, WLS, and HWLS fitting algorithms. After completion of fitting processes, the normal deviation of every point is available, so it is possible to generate the histogram of point deviations. Figure 68 presents two deviation graphs originated in measurement with GOM Atos camera and Romer arm. The normalized histogram of these two sets of point deviations are presented in Figure 69.

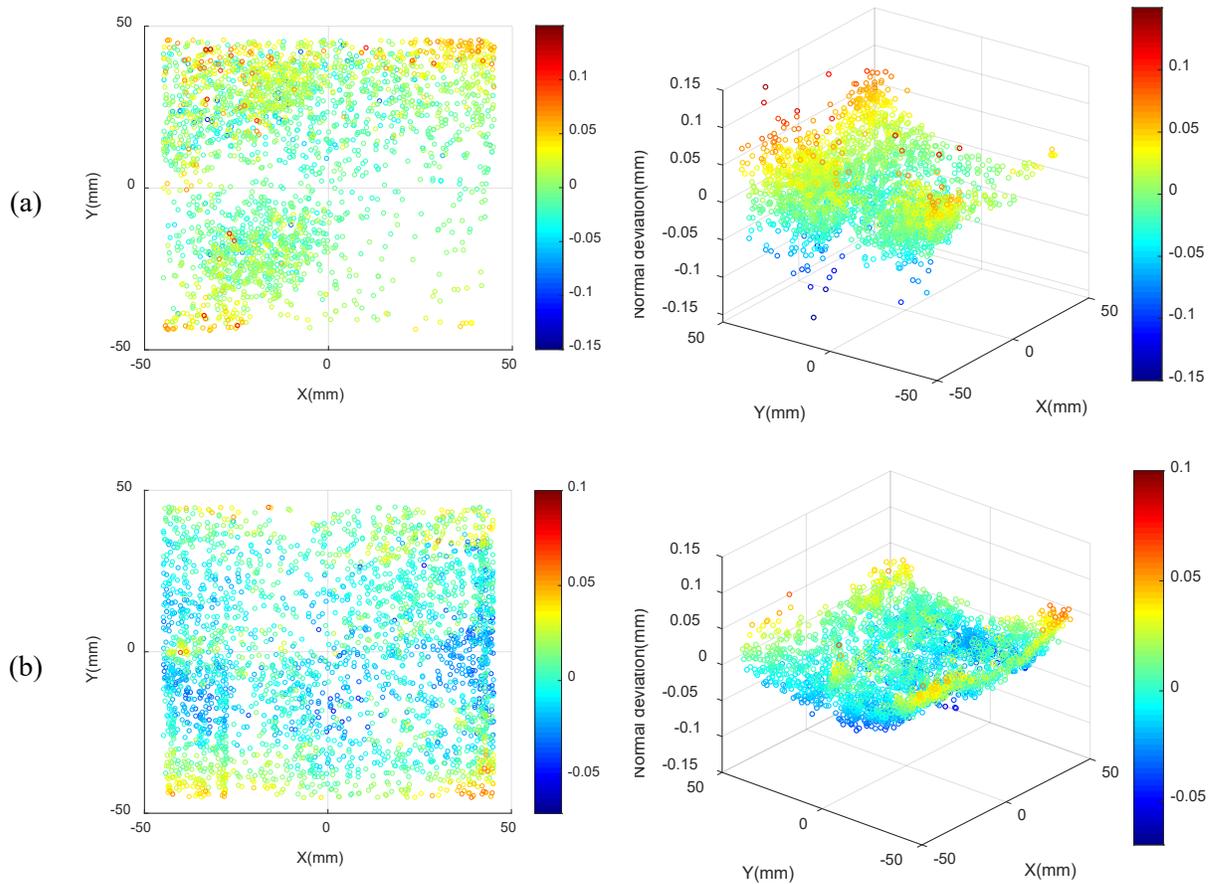


Figure 68. Individual point deviations from the saddle profile obtained from the WLS fitting algorithm; (a) measured with the GOM camera; (b) measured with the Romer arm

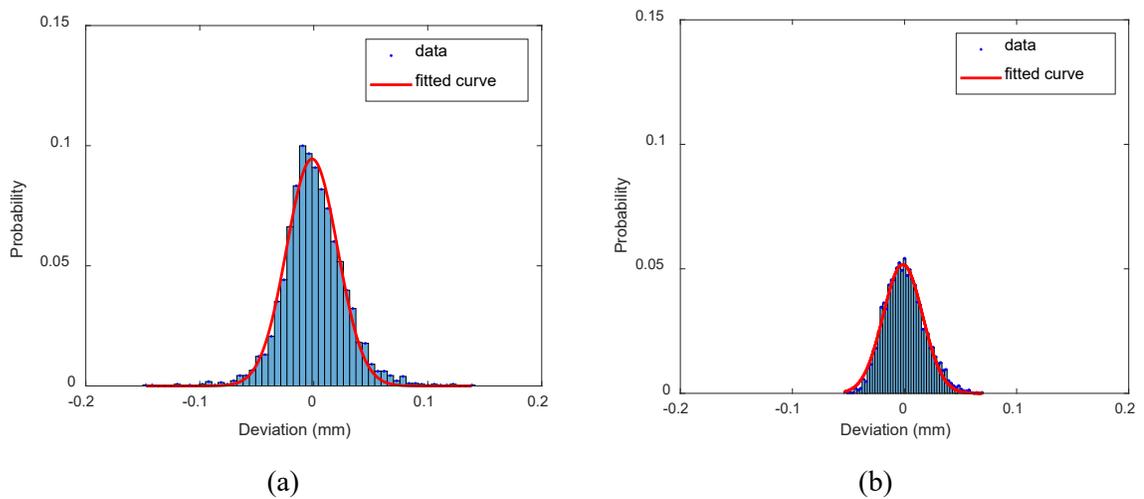


Figure 69. Histogram of deviations shown in previous figure; (a) measured with the GOM camera (b) measured with the Romer arm

Graphs of Figure 68 show a parabolic shape for point deviations, although the data originated in scanning with the GOM camera seem more scattered than the second deviation data set. This higher amount of the data dispersion is more visible in histograms. It is comprehensive that although points measured with the GOM Atos camera have wider range of deviations, but the probability of deviations close to zero is higher with this measuring instrument. Additionally, the standard deviation of two data sets can be estimated from the histograms. Therefore, the standard deviation of data obtained from the Romer arm and the GOM camera are respectively about 15 and 25 micrometers. These standard deviations are significantly smaller than the assumed value in simulations, that is why fitting parameters obtained from real samples are dispersed within ranges that are narrower than the range of simulated parameters. If the saddle part is measured with the tactile CMM to obtain data with a lower uncertainty, it will be possible to estimate the quality of performed optical scans.

For further analysis, the histogram and the CDF plot of point deviations derived from various fitting algorithms are generated and compared together, but they seem extremely identical, and no specific result is obtained from this comparison. Additionally, even histograms and CDF plots belonging to multiple scans with the same instrument, have similar appearance. Two CDF plots originated in separate scans of the saddle part with the Romer arm are shown in Figure 70. To provide a better comparison, the curve of normal distribution is shown with the red dashed line. Two curves do not coincide at the 0.5 cumulative probability because the normal curve is centered at zero, while the median of point deviations is a small positive number. Selected samples with uniform structure are processed as well as randomly selected samples. The histogram and the CDF plot of point deviations associated to a uniform sample are presented in Figure 71. It is visible that the range of deviations is lower with uniform sampling that is mainly due to the small size of samples (361 points). Nonetheless, the Gaussian curve fitted to the histogram data is nearly similar

to the curve derived from non-uniform sampling. Besides, it is important to notice that the latter histogram is not symmetric, and the dispersion of deviations is higher on the positive side.

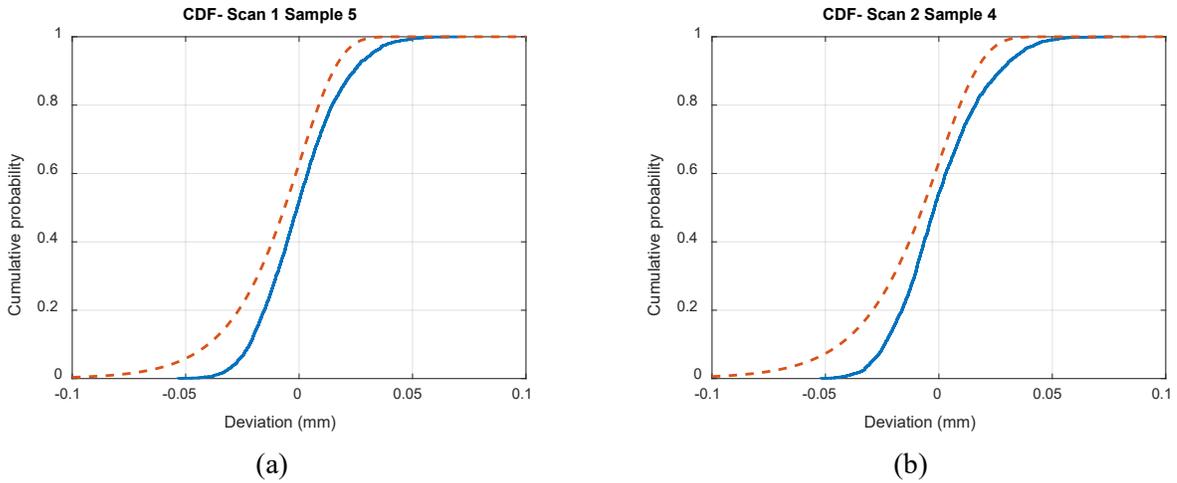


Figure 70. The CDF plot of point deviations derived from two separate scans with the Romer arm

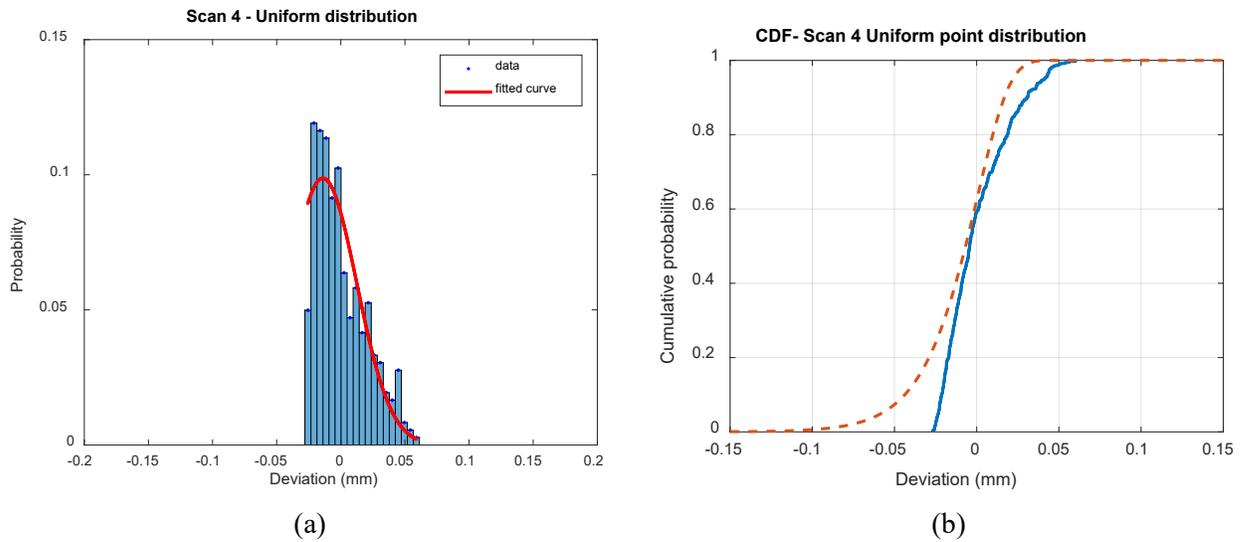


Figure 71. Distribution of point deviations obtained from fitting a sample with uniform structure; (a) histogram; (b) CDF plot

To understand the difference between separate deviation data sets, the two-sample Kolmogorov-Smirnov test (*kstest2* in Matlab) is used. In this method, the similarity of two selected samples is tested by comparing their CDF values. The default significance level is 5%, and it can

be set to other values. If the output of the test is one, it means the hypothesis of similarity is rejected (samples are not similar), and if the output is zero, the hypothesis is not rejected. Sets of deviation data obtained from three fitting algorithms are tested by the *kstest2* Matlab function. Results of analyzing non-uniform samples with 5% significance level declare that fitting algorithms produce dissimilar deviations, although deviations obtained from different samples selected from one measured data set are similar. The Evaluation of samples with uniform point density is performed with the mentioned test as well. As it is expected, point deviations resulted from three fitting algorithms are similar with uniform samples. However, it should be noticed that the Kolmogorov-Smirnov test evaluates the similarity of two deviation sets, and it does not consider the location of matched points on the profile.

CHAPTER 6: CONCLUSIONS AND FUTURE WORKS

In the first chapter, different intentions of part profile measurement were described, and the basic role of knowledge about the profile measurement uncertainty was discussed. It was mentioned that the quantification of profile measurement data is the first step for the assessment of the profile measurement uncertainty, and there has been no standard method for this required quantification so far.

In the second chapter, different profile measurement data processing methods were described with a focus on various data fitting methods. Recent research works focused on the analysis of profile measurement data were reviewed, and drawbacks and strong points of various fitting methods were discussed. It was stated that since the Minimax method is susceptible to noisy data, it is not used with measurement data obtained with optical methods. The Least Squares (LS) fitting method has a better performance on optically measured data, although the location of measured points is neglected in this method. Due to this characteristic, the non-uniform point density in profile scan data causes inconsistency in the output of the fitting procedure. Based on the LS method, two new fitting algorithms, Weighted Least Squares (WLS) and Highly Weighted Least Squares (HWLS), were developed to overcome the mentioned issue of the LS fitting method. In these algorithms, the weight of every point deviation is defined based on the distance of that point from neighbor points. Therefore, higher point density is equal to lower point deviation weights and vice versa.

For testing the performance of proposed fitting algorithms, separate data sets with uniform and non-uniform point distribution were simulated over 2D and 3D profiles. Simulated data points were matched to considered profiles by means of three fitting algorithms, and optimized fitting parameters resulted from various point layouts were analyzed as well as peak to valley (PV) distances (the range of individual point deviations). Results show that the optimum condition of

the target function is independent from the origin's position, and final point deviations are constant for every fitting algorithm, although obtained fitting parameters depend on the origin's position. As it was expected, three fitting algorithms had the same performance on uniform data, while their outputs were totally different with unorganized data sets. In some cases, the LS algorithm yielded poor results that are far from desired values, although the statistical analysis of simulated PV data is in contrast with it. Therefore, it is comprehensive that another criterion in addition to the PV distance is required for the performance evaluation of fitting algorithms. Researching new criteria was out of the scope of this work, so it is suggested as a future work. Another suggestion for the future research is to analyze the influence of non-random point deviations on the performance of fitting algorithms. For instance, periodic orthogonal functions can be used for the generation of the noise in data points.

In the next stage of this work, workpieces with irregular 2D profile shapes and freeform 3D shapes were selected and measured with various tactile and optical methods. After initial data preparation process, measured points were fitted to associated nominal profiles using three introduced fitting algorithms. Comparing point deviation graphs derived from various measurement methods declared that deviations are locally constant, and it originated the main idea of using a convolution averaging filter to produce the Mean Local Deviation (MLD) of profile segments for quantification of profile measurement data. In this method, the whole profile is divided into segments, and the deviation of points belonging to each segment are averaged. Obtained MLDs can be used for the analogy between profile data sets as well as the quantification of the profile measurement error. Point deviations obtained by applying three fitting algorithms to experimentally measured data were quantified by means of the proposed method, and the difference between them was calculated. Results show that the difference between WLS and LS algorithms is lower than the difference between HWLS and LS algorithms. Similarly, fitting parameters obtained from the WLS algorithm were closer to values derived from the LS algorithm. Furthermore, from

initial analysis, it seems that differences between quantified deviation data sets obtained from various fitting algorithms are lower with uniform point clouds. However, due to the lack of enough computation memory, it was not possible to analyze large data sets with a tiny stop limit for the target function (TolX) in the Matlab. This part of work can be executed in a future research work.

In addition to comparing quantified deviation data sets, the two-sample Kolmogorov-Smirnov test was used for the analogy of individual point deviation data. Results of analyses with the default value (5%) for the significance level show that deviation data sets produced by different fitting algorithms are dissimilar if the point distribution is non-uniform. However, three fitting algorithms yield similar point deviations with uniform samples, and this similarity exists even between multiple measurements. Analysis of the significance level for results obtained from non-uniform samples, and the influence of the proposed quantification method on data similarity are suggested for future works.

REFERENCES

- [1] Bradley, C, Vickers, G W, and Milroy, M. “Reverse Engineering of Quadric Surfaces Employing Three-Dimensional Laser Scanning.” *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture* 208.1 (1994): 21–28. Web.
- [2] Bradley, Colin H, and Vickers, Geoffrey W. “Free-Form Surface Reconstruction for Machine Vision Rapid Prototyping.” *Optical Engineering* 32.9 (1993): 2191–2200. Web.
- [3] Akca, Devrim, and Grün, Armin. *Least Squares 3D Surface Matching*. ETH Zurich, 2006. Web.
- [4] Zhao, Xiuyang et al. “IGA-Based Point Cloud Fitting Using B-Spline Surfaces for Reverse Engineering.” *Information Sciences* 245 (2013): 276–289. Web.
- [5] ASME standard Y14.5, section 8, *Tolerances of Profile*, 2009.
- [6] ISO standard, ISO 1660, *Geometrical product specifications (GPS) - Geometrical tolerancing- Profile tolerancing*, 2012.
- [7] Brown, Matthew S. *Automated Conversion of 3D Point Clouds to Solid Geometrical Computer Models*. Drexel University, 2018.
- [8] Hocken, Robert J., and Pereira, Paulo H. *Coordinate Measuring Machines and Systems*, Second Edition. 2nd ed. CRC Press, 2016. Print.
- [9] Van Gestel, Nick et al. “A Performance Evaluation Test for Laser Line Scanners on CMMs.” *Optics and Lasers in Engineering* 47.3 (2009): 336–342. Web.
- [10] Martínez, S. et al. “Methodology for Comparison of Laser Digitizing Versus Contact Systems in Dimensional Control.” *Optics and Lasers in Engineering* 48.12 (2010): 1238–1246. Web.

- [11] Bešić, Igor et al. "Accuracy Improvement of Laser Line Scanning for Feature Measurements on CMM." *Optics and Lasers in Engineering* 49.11 (2011): 1274–1280. Web.
- [12] Sładek, Jerzy et al. "The Hybrid Contact–optical Coordinate Measuring System." *Measurement* 44.3 (2011): 503–510. Web.
- [13] Valdez, Mario Orlando. *Task-Specific Uncertainty for Industrial Measurements*. 2015. Ann Arbor, MI.
- [14] <https://www.tristar.com/eight-wastes-product-development/>
- [15] www.gdandtbasics.com/profile-of-a-surface/
- [16] Drieschner, Rudolf. "Chebyshev Approximation to Data by Geometric Elements." *Numerical Algorithms* 5.10 (1993): 509–522. Web.
- [17] Jamiolahmadi, Saeed, and Barari, Ahmad. "Study of Detailed Deviation Zone Considering Coordinate Metrology Uncertainty." *Measurement* 126 (2018): 433–457. Web.
- [18] Edgeworth, Robert, and Wilhelm, Robert G. "Adaptive Sampling for Coordinate Metrology." *Precision Engineering* 23.3 (1999): 144–154. Web.
- [19] Summerhays, K.D et al. "Optimizing Discrete Point Sample Patterns and Measurement Data Analysis on Internal Cylindrical Surfaces with Systematic Form Deviations." *Precision Engineering* 26.1 (2002): 105–121. Web.
- [20] Shakarji, Craig. "Least-Squares Fitting Algorithms of the NIST Algorithm Testing System." *National Institute of Standards and Technology, Journal of Research* 103.6 (1998): 633–641. Web.
- [21] Dall&Apos, and Osso, Aldo. "A Least Squares Method in Back Fitting the Data Base of a Simulation Model." *Advances in Engineering Software* 33.11 (2002): 743–748. Web.

- [22] Shakarji, C.M., and Clement, A. "Reference Algorithms for Chebyshev and One-Sided Data Fitting for Coordinate Metrology." *CIRP Annals - Manufacturing Technology* 53.1 (2004): 439–442. Web.
- [23] Jiang, Qimi et al. "A Roundness Evaluation Algorithm with Reduced Fitting Uncertainty of CMM Measurement Data." *Journal of Manufacturing Systems* 25.3 (2006): 184–195. Web.
- [24] Zhang, Xiangchao et al. "Chebyshev Fitting of Complex Surfaces for Precision Metrology." *Measurement* 46.9 (2013): 3720–3724. Web.
- [25] Vemulapalli, Prabath. Reconciling the differences between tolerance specification and measurement methods. Arizona State University, 2014.
- [26] Smítka, Václav, and Štroner, Martin. "3D Scanner Point Cloud Denoising by Near Points Surface Fitting." Vol. 8791. SPIE, 2013. 87910C–87910C–15. Web.
- [27] Mohan, Prashant et al. "Development of a Library of Feature Fitting Algorithms for CMMs." *International Journal of Precision Engineering and Manufacturing* 16.10 (2015): 2101–2113. Web.
- [28] Zhang, Huaiqing et al. "Measurement Data Fitting Based on Moving Least Squares Method." *Mathematical Problems in Engineering* 2015 (2015): 10. Web.
- [29] Omar AL-Shebeeb, Tahseen F. Abbas, and Abdulwahab N. Abbas. Surface Fitting and Representation By Using 2D Least Squares Method in CAD Applications. Unpublished, 2015. Web.
- [30] Moroni, Giovanni, Syam, Wahyudin P., and Petró, Stefano. "Comparison of Chaos Optimization Functions for Performance Improvement of Fitting of Non-Linear Geometries." *Measurement* 86.C (2016): 79–92. Web.

- [31] Hutzschenreuter, Daniel, Härtig, Frank, and Schmidt, Markus. “An SQP Method for Chebyshev and Hole-Pattern Fitting with Geometrical Elements.” *Journal of Sensors and Sensor Systems* 7.1 (2018): n. pag. Web.
- [32] Arezki, Yassir et al. “A Novel Hybrid Trust Region Minimax Fitting Algorithm for Accurate Dimensional Metrology of Aspherical Shapes.” *Measurement* 127 (2018): 134–140. Web.
- [33] Zaman, Faisal, Wong, Ya, and Ng, Boon. “Density-Based Denoising of Point Cloud.” *arXiv.org* (2016): n. pag. Web.
- [34] Wagner, Marc et al. “C 2-continuous surface reconstruction with piecewise polynomial patches.” (2003).
- [35] Várady, Tamás, Martin, Ralph R, and Cox, Jordan. “Reverse Engineering of Geometric Models—an Introduction.” *Computer-Aided Design* 29.4 (1997): 255–268. Web.
- [36] Tzung-Sz Shen, Jianbing Huang, and Chia-Hsiang Menq. “Multiple-Sensor Integration for Rapid and High-Precision Coordinate Metrology.” *IEEE/ASME Transactions on Mechatronics* 5.2 (2000): 110–121. Web.
- [37] Colosimo, Bianca Maria, Pacella, Massimo, and Senin, Nicola. “Multisensor Data Fusion via Gaussian Process Models for Dimensional and Geometric Verification.” *Precision Engineering* 40.C (2015): 199–213. Web.
- [38] Rak, Michal Bartosz, Wozniak, Adam, and Mayer, J.R.R. “The Use of Low Density High Accuracy (LDHA) Data for Correction of High Density Low Accuracy (HDLA) Point Cloud.” *Optics and Lasers in Engineering* 81 (2016): 140–150. Web.

- [39] Chen, Y, and Medioni, G. "Object Modeling by Registration of Multiple Range Images." Proceedings. 1991 IEEE International Conference on Robotics and Automation. Vol. 3. IEEE Comput. Soc. Press, 1991. 2724–2729 vol.3. Web.
- [40] Besl, Paul J, and Mckay, Neil D. "Method for Registration of 3-D Shapes." Proceedings of SPIE. Vol. 1611. N.p., 1992. 586–606. Web.
- [41] Turk, Greg and Marc Levoy. "Zippered polygon meshes from range images." *SIGGRAPH* (1994).
- [42] Akca, D. and Gruen, A., 2006. Recent Advances in Least Squares 3d Surface Matching, ETH Zurich.
- [43] Milroy, Michael James. "Automation of laser-scanner-based reverse engineering." (1998): 2637-2637.
- [44] Senin, N., Colosimo, B.M., and Pacella, M. "Point Set Augmentation through Fitting for Enhanced ICP Registration of Point Clouds in Multisensor Coordinate Metrology." *Robotics and Computer Integrated Manufacturing* 29.1 (2013): 39–52. Web.
- [45] Mitra, Niloy J. "Algorithms for comparing and analyzing three-dimensional geometry." (2006).
- [46] Cheng, Peng. Automatic complete polygon surface reconstruction from overlapped 3d point clouds. Diss. University of South Carolina, 2001.
- [47] Lukács, G., A. D. Marshall, and R. R. Martin. "Geometric least-squares fitting of spheres, cylinders, cones and tori." *RECCAD deliverable documents* 2 (1997): 1-20.
- [48] Mavriplis, Dimitri J. "An advancing front Delaunay triangulation algorithm designed for robustness." *Journal of Computational Physics* 117.1 (1995): 90-101.

- [49] Bernardini, F et al. "The Ball-Pivoting Algorithm for Surface Reconstruction." IEEE Transactions on Visualization and Computer Graphics 5.4 (1999): 349–359. Web.
- [50] Zhang, Xuechang, et al. "Design of an open tolerance and complex surfaces inspection system based on the point cloud data." 2009 IEEE International Conference on Intelligent Computing and Intelligent Systems. Vol. 4. IEEE, 2009.
- [51] Inoue, Hiroshi. "A Least-squares Smooth Fitting for Irregularly Spaced Data: Finite-element Approach Using the Cubic B-Spline Basis." GEOPHYSICS 51.11 (1986): 2051–2066. Web.
- [52] Arun, K. Somani, Thomas S. Huang, and Steven D. Blostein. "Least-squares fitting of two 3-D point sets." IEEE Transactions on pattern analysis and machine intelligence 5 (1987): 698-700.
- [53] Levin, David. "The approximation power of moving least-squares." Mathematics of Computation of the American Mathematical Society 67.224 (1998): 1517-1531.
- [54] Amenta, N., and Bern, M. "Surface Reconstruction by Voronoi Filtering." Discrete & Computational Geometry 22.4 (1999): 481–504. Web.
- [55] Ohtake, Yutaka et al. "Mesh Smoothing by Adaptive and Anisotropic Gaussian Filter Applied to Mesh Normals." *VMV* (2002).
- [56] A. Belyaev, Y. Ohtake, "A Comparison of Mesh Smoothing Methods", *Proc. Israel-Korea Bi-Nat'l Conf. Geometric Modeling and Computer Graphics*, pp. 83-87, 2003.
- [57] S. Fleishman, D. Cohen-Or, C.T. Silva, Robust moving least-squares fitting with sharp features, in: SIGGRAPH '05: ACM SIGGRAPH 2005 Papers, 2005.
- [58] Vieira, M. "Processing three-dimensional laser scanner data for automotive styling design." (2005): 1674-1674.

- [59] Huang, Jianbing. Geometric feature extraction and model reconstruction from unorganized points for reverse engineering of mechanical objects with arbitrary topology. Diss. The Ohio State University, 2001.
- [60] <http://www.ifp.uni-stuttgart.de/publications/phowo13/300Ioannides.pdf>
- [61] <https://reference.wolfram.com/language/TriangleLink/tutorial/UsingTriangleLink.html>
- [62] Javidpour, F, and E P Morse. “Quantification of Part Profile Scanning Error.” American Society for Precision Engineering, *Proceedings of the 32nd Annual Meeting*, Nov. 2017.

APPENDIX B: A COLLECTION OF MATLAB CODES

Data simulation and fitting to the quarter circle profile

Uniform point distribution

Main program

```

%% Clearing workspace and command window
clear all; clc; close all

%% Definition of variables
global R % Probing direction
global points % Captured points
global features % list and characteristics of nominal features
global NOF % Number of features
global NOP % Number of points for each feature
global CP % Points on the surface
txy0 = [0,0,0]; % Initial translation and rotation of points [Rot(d)
TrX TrY]

%% Creation of profile matrix by using the text of profile data file
[features, NOF] = make_nominal('Quarter_circle_nominal.txt');
m=1; % number of repetition
for k= 1:m
%% Creation of point coordinates (constant distance)
n2= 10; % number of points on the arc
dis= 10*pi/(4*(n2-1)); % distance between points
n1= floor(5/dis); % number of points on the lines
n3= n1;
x_pts1= (1:n1) '*dis;
y_pts1= 0.4*rand(n1,1)-0.2;
y_pts1= -0.2* ones(n1,1);

theta= (10:10:80);
x_pts2= ((0.4*rand(n2-2,1)+4.8).*cosd(theta)');
y_pts2= ((0.4*rand(n2-2,1)+4.8).*sind(theta)');

x_pts3= 0.4*rand(n3,1)-0.2;
y_pts3= (5-dis:-dis:1)';
points= [x_pts1, y_pts1; x_pts2, y_pts2; x_pts3, y_pts3];

%% Distribution of points to the features
[CP,NOP]=dist_points(points, features); % output consists of groups of
points and the number of points in each group

%% fitting the simulated points to the nominal profile
[SS0,TP,resids0,lenghts0]= Least_Sqr(txy0); % TP= Transferred points
options = optimset('Display', 'iter', 'TolX', 1e-12);
TXY_LS(k,1:3) = fminsearch(@Least_Sqr, txy0, options);
[LS,FP_LS,dev_LS,len_LS]= Least_Sqr(TXY_LS(k,1:3)); % FP= Fitted points

Min_dev(k,1)= min(min(dev_LS));

```

```

Max_dev(k,1)= max(max(dev_LS));

[HWSS0,TP,resids0,lenghts0]= Highly_Weighted_Least_Sqr(txy0); % TP=
Transferred points
options = optimset('Display', 'iter','TolX', 1e-12);
TXY_HWLS(k,1:3) = fminsearch(@Highly_Weighted_Least_Sqr, txy0,
options);

[HWLS,FP_HWLS,dev_HWLS,len_HWLS]=
Highly_Weighted_Least_Sqr(TXY_HWLS(k,1:3)); % FP= Fitted points

Min_dev(k,3)= min(min(dev_HWLS));
Max_dev(k,3)= max(max(dev_HWLS));

[WSS0,TP,resids0,lenghts0]= Weighted_Least_Sqr(txy0); % TP= Transferred
points
options = optimset('Display', 'iter','TolX', 1e-12);
TXY_WLS(k,1:3) = fminsearch(@Weighted_Least_Sqr, txy0, options);
[WLS,FP_WLS,dev_WLS,len_WLS]= Weighted_Least_Sqr(TXY_WLS(k,1:3)); % FP=
Fitted points

Min_dev(k,2)= min(min(dev_WLS));
Max_dev(k,2)= max(max(dev_WLS));

%% CDFs of individual point deviations
no_devs = max(NOP)*NOF;
pd = reshape(dev_LS,no_devs,1);
pd_LS = nonzeros(pd);
p1 = cdfplot(pd_LS);
set(p1,'LineWidth', 1.5, 'Color','b');
hold on

pd = reshape(dev_WLS,no_devs,1);
pd_WLS = nonzeros(pd);
p2 = cdfplot(pd_WLS);
set(p2,'LineWidth', 1.5, 'Color','g');

pd = reshape(dev_HWLS,no_devs,1);
pd_HWLS = nonzeros(pd);
p3 = cdfplot(pd_HWLS);
set(p3,'LineWidth', 1.5, 'Color','r');

clear pd pd_LS pd_WLS pd_HWLS p1 p2 p3
end

hold off
%% plotting the nominal profile and fitted points

plot_profile(features,FP_LS);
t1= title('Fitted points (LS)');
set(t1,'FontSize',22);
plot_profile(features,FP_HWLS);
t2= title('Fitted points (HWLS)');
set(t2,'FontSize',22);
plot_profile(features,FP_WLS);

```

```

t3= title('Fitted points (WLS)');
set(t3,'FontSize',22);
%% generating the bar graph of point deviations
[Point_lengths_LS, Point_deviations_LS]= bar_graph(features,len_LS,
dev_LS);
t4= title('Least Square');
set(t4,'FontSize',22);
xlim([min(Point_lengths_LS)-2, max(Point_lengths_LS)+2]);

[Point_lengths_HWLS, Point_deviations_HWLS]=
bar_graph(features,len_HWLS, dev_HWLS);
t5= title('Highly Weighted Least Square');
set(t5,'FontSize',22);
xlim([min(Point_lengths_HWLS)-2, max(Point_lengths_HWLS)+2]);

[Point_lengths_WLS, Point_deviations_WLS]= bar_graph(features,len_WLS,
dev_WLS);
t6= title('Weighted Least Square');
set(t6,'FontSize',22);
xlim([min(Point_lengths_WLS)-2, max(Point_lengths_WLS)+2]);

%% generation of histograms related to the range of point deviations
range_dev= Max_dev - Min_dev;

figure;
histfit(range_dev(:,1));
xlim([0.1 0.8]);
ylim([0 30]);
title('Range of point deviations- LS');

figure;
histfit(range_dev(:,2));
xlim([0.1 0.8]);
ylim([0 30]);
title('Range of point deviations- WLS');

figure;
histfit(range_dev(:,3));
xlim([0.1 0.8]);
ylim([0 30]);
title('Range of point deviations- HWLS');

%% generation of CDF plots of deviation ranges
figure;
cdfplot(range_dev(:,1));
hold on
cdfplot(range_dev(:,2));
cdfplot(range_dev(:,3));
xlim([0.2 0.5]);
hold off

```

Functions

```
function [components, n] = make_nominal(filename)
```

```
global debug
```

```
global R
```

```
file_id = fopen(filename, 'r');
t = fgetl(file_id);
components = zeros(1,7);
n = 0;
while true
    t = fgetl(file_id);
    if(debug),fprintf('%s\n',t);end;
    d = sscanf(t,'%s %f %f %f %f %f %f');
    if d(1) == 'E', break, end;
    if d(1) == '%', n=n-1; end;
    if d(1) == 'R', R=d(2); n=n-1; end;
    n = n+1;
    if d(1) == 'L'
        components(n,1:5) = d;
    end
    if d(1) == 'C'
        components(n,1:7) = d;
    end
end
fclose(file_id);
%
if(debug),disp('done!');end;

return
```

Quarter_circle_nominal.txt

```
% Sample quarter circle
%
% L=Line      X1 Y1 X2 Y2
% C=Circle   Xc Yc R A1 A2 Dir (+1 is CCW, -1 is CW)
% R=Rotation direction (Direction of probe movement around the profile)
% (+1 is CCW, -1 is CW)
% E= end of file
%
%
R  1
L  0  0  5  0
C  0  0  5  0  90  1
L  0  5  0  0
E
```

```
function [SP,t]= dist_points(pts,fters)
% outputs are the distributed points and the number of points for each
feature
global NOF
n=length(pts); % total number of points
Pgroups=zeros(n,2,NOF); % point groups
Distances=ones(n,NOF)*20; % distances of points to the features
for i=1:NOF
```

```

switch ftrs(i,1)
    case 'L'
        stpt = ftrs(i,2:3);
        endpt =ftrs(i,4:5);
        line_vec = endpt-stpt;
        for j=1:n
            delta = pts(j,1:2)-stpt;
            proj = dot(delta, line_vec)/norm(line_vec);
            dist = (norm(cross([delta 0], [line_vec
0])))/norm(line_vec); % vertical distance of point to the line
            if proj >= 0 && proj < norm(line_vec)
                if dist <= 0.3
                    Pgroups(j,1:2,i)= pts(j,1:2);
                    Distances(j,i)= dist;
                end
            end
        end
    case 'C'
        Xc=ftrs(i,2); Yc=ftrs(i,3); R=ftrs(i,4);
        A1= degtorad(ftrs(i,5)); A2=degtorad(ftrs(i,6)); DIR =
ftrs(i,7);
        x1=Xc+R*cos(A1); y1=Yc+R*sin(A1);
        x2=Xc+R*cos(A2); y2=Yc+R*sin(A2);
        center= [Xc,Yc];
        stpt= [x1,y1];
        endpt=[x2,y2];
        if (A2 < A1) && (DIR==1)
            A2=A2+2*pi;
        end
        if (A2 > A1) && (DIR==-1)
            A1=A1+2*pi;
        end
        for j= 1:n
            delta = pts(j,1:2)- center;
            [theta,d]= cart2pol(delta(1),delta(2));
            if A1==2*pi && theta*DIR > 0
                theta = theta+2*pi;
            end
            if d >= R-0.3 && d <= R+0.3
                if (theta-A1)*DIR > 0 && (theta-A2)*DIR < 0
                    Pgroups(j,1:2,i)= pts(j,1:2);
                    Distances(j,i)= abs(d-R);
                end
            end
        end
end

end
SP= zeros(n,2,NOF); % separated points
for i= 1:NOF
    s= 0; % number of points for each feature
    for j=1:n
        m = min(Distances(j,:)); % minimum distance of point to the
features
        if m == Distances(j,i)

```

```

        s= s+1;
        SP(s,1:2,i)= Pgroups(j,1:2,i);
    end
end
t(i)= s;
end

function [ss,tran_pts,dev,len]= Least_Sqr(txy)

global R
global NOF
global NOP
global CP
global features

rot = [cosd(txy(1)), sind(txy(1)); -sind(txy(1)), cosd(txy(1))];
ss=0; % sum-square of deviations
for i=1:NOF
    n=0;
    for j= 1:length(CP)
        if CP(j,1:2,i)~= [0,0]
            n=n+1; % n= number of points
            p(n,1:2)=CP(j,1:2,i);
        end
    end
    if n > 1
        NOP(i)= n;
        tr = ones(n,1)*[txy(2), txy(3)];
        tran_pts(1:n,1:2,i) = p(1:n,1:2) *rot+ tr;
        switch features(i,1)
            case 'L'
                sp= features(i,2:3); % start point
                ep= features(i,4:5); % end point
                lv= ep-sp; % line vector
                [la,l]= cart2pol(lv(1), lv(2));
                for j= 1:n
                    pv= tran_pts(j,1:2,i)-sp; % point vector
                    [pa,dis]= cart2pol(pv(1), pv(2)); % point angle and
distance
                    if pa <0 && la >0
                        pa= pa+2*pi;
                    end
                    len(j,i)= dot(pv,lv)/norm(lv);
                    pv_3d= [pv 0];
                    lv_3d= [lv 0];
                    if pa > la
                        dev(j,i)= -R*norm(cross(pv_3d,lv_3d))/norm(lv);
                    else
                        dev(j,i)= R*norm(cross(pv_3d,lv_3d))/norm(lv);
                    end
                end
            case 'C'
                center= features(i,2:3); % arc center
                r= features(i,4); % arc radius

```

```

        sa= degtorad(features(i,5)); % start angle
        dir= features(i,7);
        for j= 1:n
            pv= tran_pts(j,1:2,i)-center; % point vector
            [pa,dis]= cart2pol(pv(1),pv(2)); % point angle and
distance
            angle= dir*(pa-sa);
            len(j,i)= angle*r;
            dev(j,i)= R*dir*(norm(pv)-r);
        end
        otherwise
            disp('the feature is not recognized!');
        end
        ss=ss+ sumsqr(dev(1:n,i));
    else
        ss= 1000;
        disp('n is not big enough');
    end
end
end

```

```
function [ss,tran_pts,dev,len]= Weighted_Least_Sqr(txy)
```

```

global R
global NOF
global NOP
global CP
global features

```

```

rot = [cosd(txy(1)), sind(txy(1)); -sind(txy(1)), cosd(txy(1))];
ss=0; % sum-square of deviations
for i=1:NOF
    n=0;
    for j= 1:length(CP)
        if CP(j,1:2,i)~= [0,0]
            n=n+1; % n= number of points
            p(n,1:2)=CP(j,1:2,i);
        end
    end
    end
    if n > 1
        NOP(i)= n;
        tr = ones(n,1)*[txy(2), txy(3)];
        tran_pts(1:n,1:2,i) = p(1:n,1:2) *rot+ tr;
        switch features(i,1)
            case 'L'
                sp= features(i,2:3); % start point
                ep= features(i,4:5); % end point
                lv= ep-sp; % line vector
                [la,l]= cart2pol(lv(1), lv(2));
                for j= 1:n
                    pv= tran_pts(j,1:2,i)-sp; % point vector
                    [pa,dis]= cart2pol(pv(1), pv(2)); % point angle and
distance
                    if pa <0 && la >0
                        pa= pa+2*pi;
                    end
                end
            end
        end
    end
end

```

```

end
len(j,i)= dot(pv,lv)/norm(lv);
pv_3d= [pv 0];
lv_3d= [lv 0];
if pa > la
    dev(j,i)= -R*norm(cross(pv_3d,lv_3d))/norm(lv);
else
    dev(j,i)= R*norm(cross(pv_3d,lv_3d))/norm(lv);
end
end
case 'C'
center= features(i,2:3); % arc center
r= features(i,4); % arc radius
A1= degtorad(features(i,5)); % start angle
A2= degtorad(features(i,6)); % end angle
dir= features(i,7);
if (A2 < A1) && (dir==1)
    A2=A2+2*pi;
end
if (A2 > A1) && (dir==-1)
    A1=A1+2*pi;
end
arc_ang= (A2-A1)*dir; % arc angle
l= arc_ang*r;
for j= 1:n
    pv= tran_pts(j,1:2,i)-center; % point vector
    [pa,dis]= cart2pol(pv(1),pv(2)); % point angle and
distance
    angle= dir*(pa-A1);
    len(j,i)= angle*r;
    dev(j,i)= R*dir*(norm(pv)-r);
end
otherwise
    disp('the feature is not recognized!');
end
mid_point(1,i)= 0;
for j=2:n
    mid_point(j,i)= (len(j,i)+len(j-1,i))/2;
end
mid_point(n+1,i)= l;
for j= 1:n
    weight(j,i)= mid_point(j+1,i)- mid_point(j,i);
end
w_dev(1:n,i)= abs(weight(1:n,i).*(dev(1:n,i)).^2); % weighted
deviation
ss=ss+ sum(w_dev(1:n,i));
else
    ss= 1000;
    disp('n is not big enough');
end
end
end

function [ss,tran_pts,dev,len]= Highly_Weighted_Least_Sqr(txy)

global R

```

```

global NOF
global NOP
global CP
global features

rot = [cosd(txy(1)), sind(txy(1)); -sind(txy(1)), cosd(txy(1))];
ss=0; % sum-square of deviations
for i=1:NOF
    n=0;
    for j= 1:length(CP)
        if CP(j,1:2,i)~= [0,0]
            n=n+1; % n= number of points
            p(n,1:2)=CP(j,1:2,i);
        end
    end
    end
    if n > 1
        NOP(i)= n;
        tr = ones(n,1)*[txy(2), txy(3)];
        tran_pts(1:n,1:2,i) = p(1:n,1:2) *rot+ tr;
        switch features(i,1)
            case 'L'
                sp= features(i,2:3); % start point
                ep= features(i,4:5); % end point
                lv= ep-sp; % line vector
                [la,1]= cart2pol(lv(1), lv(2));
                for j= 1:n
                    pv= tran_pts(j,1:2,i)-sp; % point vector
                    [pa,dis]= cart2pol(pv(1), pv(2)); % point angle and
distance

                    if pa <0 && la >0
                        pa= pa+2*pi;
                    end
                    len(j,i)= dot(pv,lv)/norm(lv);
                    pv_3d= [pv 0];
                    lv_3d= [lv 0];
                    if pa > la
                        dev(j,i)= -R*norm(cross(pv_3d,lv_3d))/norm(lv);
                    else
                        dev(j,i)= R*norm(cross(pv_3d,lv_3d))/norm(lv);
                    end
                end
            case 'C'
                center= features(i,2:3); % arc center
                r= features(i,4); % arc radius
                A1= degtorad(features(i,5)); % start angle
                A2= degtorad(features(i,6)); % end angle
                dir= features(i,7);
                if (A2 < A1) && (dir==1)
                    A2=A2+2*pi;
                end
                if (A2 > A1) && (dir==-1)
                    A1=A1+2*pi;
                end
                arc_ang= (A2-A1)*dir; % arc angle
                l= arc_ang*r;
                for j= 1:n
                    pv= tran_pts(j,1:2,i)-center; % point vector

```

```

                                [pa,dis]= cart2pol(pv(1),pv(2)); % point angle and
distance
                                angle= dir*(pa-A1);
                                len(j,i)= angle*r;
                                dev(j,i)= R*dir*(norm(pv)-r);
                                end
                                otherwise
                                    disp('the feature is not recognized!');
                                end
                                mid_point(1,i)= 0;
                                for j=2:n
                                    mid_point(j,i)= (len(j,i)+len(j-1,i))/2;
                                end
                                mid_point(n+1,i)= 1;
                                for j= 1:n
                                    weight(j,i)= mid_point(j+1,i)- mid_point(j,i);
                                end
                                w_dev(1:n,i)= weight(1:n,i).* dev(1:n,i); % weighted deviation
                                ss=ss+ sumsqr(w_dev(1:n,i));
                                else
                                    ss= 1000;
                                    disp('n is not big enough');
                                end
                                end
                                end
end

```

```
function plot_profile(ftrs,pts)
```

```
global NOF
```

```

n= length(pts);
figure
for i=1:NOF
    switch ftrs(i,1)
        case 'L'
            line(ftrs(i,[2,4]),ftrs(i,[3,5]));
        case 'C'
            Xc = ftrs(i,2); Yc = ftrs(i,3);
            R = ftrs(i,4);
            A1 = ftrs(i,5); A2 = ftrs(i,6);
            DIR = ftrs(i,7); % DIR = 1 is CCW, DIR = -1 is CW
            if (A2 < A1) && (DIR==1)
                A2=A2+360;
            end
            if (A2 > A1) && (DIR==-1)
                A1=A1+360;
            end
            arc_ang= (A2-A1)*DIR; % arc angle
            seg_num = ceil(arc_ang/5)+1;
            a_list = linspace(A1,A2,seg_num);
            x_list = Xc + R*cosd(a_list);
            y_list = Yc + R*sind(a_list);
            line(x_list,y_list);
        otherwise
            disp('feature is not recognized!');
    end
end

```

```

end
hold on
% color codes
color= [1 0 0; 0.4 0 0.5; 1 1 0; 1 0.5 0.5;... % red, purple,
yellow, pink
        0 1 0; 0 0 0; 0 0 1; 0.2 0.5 0.3;... % light green,
black, dark blue, dark green
        1 0.5 0; 0.1 0.6 1; 0.8 0.8 0.8; 0.5 0.2 0]; % orange,
light blue, gray, brown
for j= 1:n
    if pts(j,1,i)~=0 || pts(j,2,i)~=0
        plt= plot(pts(j,1,i),pts(j,2,i), '*');
        set(plt, 'Color',color(i,:));
    end
end
end
axis equal
a= min(min(pts(:,1,:)))-1; b= max(max(pts(:,1,:)))+1; % X axis limits
c= min(min(pts(:,2,:)))-1; d= max(max(pts(:,2,:)))+1; % Y axis limits
axis([a,b,c,d]);
t1= xlabel('X (mm)');
t2= ylabel('Y (mm)');
%t3= title('Fitted points');
set(t1, 'FontSize',18);
set(t2, 'FontSize',18);
%set(t3, 'FontSize',22);
set(gca, 'FontSize', 16);
hold off

```

```

function [pt_lens, pt_devs]= bar_graph(ftrs, lns, devs)
global NOP
m= size(lns,1);
n= size(lns,2); % number of features
l=0; % initial length
s=0;
color= [1 0 0; 0.4 0 0.5; 1 1 0; 1 0.5 0.5;... % red, purple, yellow,
pink
        0 1 0; 0 0 0; 0 0 1; 0.2 0.5 0.3;... % light green, black,
dark blue, dark green
        1 0.5 0; 0.1 0.6 1; 0.8 0.8 0.8; 0.5 0.2 0]; % orange, light
blue, gray, brown
figure
for i=1:n
    for j= 1:m
        if lns(j,i)~=0
            lns(j,i)= lns(j,i)+1;
            pt_lens(s+1)= lns(j,i);
            pt_devs(s+1)= devs(j,i);
            s= s+1;
        end
    end
    bg= bar(lns(1:NOP(i),i),devs(1:NOP(i),i));
    set(bg, 'EdgeColor',color(i,:), 'FaceColor',color(i,:));
    hold on
    switch ftrs(i,1)

```

```

    case 'L'
        sp= ftrs(i,2:3);
        ep= ftrs(i,4:5);
        lv= ep- sp;
        l= 1+ norm(lv);
    case 'C'
        R = ftrs(i,4); A1 = ftrs(i,5); A2 = ftrs(i,6);
        DIR = ftrs(i,7); % DIR = 1 is CCW, DIR = -1 is CW
        if (A2 < A1) && (DIR==1)
            A2=A2+360;
        end
        if (A2 > A1) && (DIR==-1)
            A1=A1+360;
        end
        arc_ang= (A2-A1)*DIR; % arc angle
        l= 1+ degtorad(arc_ang)*R;
    otherwise
        disp('no L or C');
end

end

t1= xlabel('length (mm)');
t2= ylabel('deviation (mm)');
set(t1,'FontSize',18);
set(t2,'FontSize',18);
set(gca, 'FontSize', 16);
hold off

```

Data simulation and fitting to the saddle profile

Non-uniform point distribution

Main program

```

%% Clearing workspace and command window
clear all; clc; close all

%% Definition of variables
global N % Number of points
global MP % Measured points
RT0 = [0 0 0; 0 0 0]; % Initial translation and rotation of points
[Rotx(d) Roty(d) Rotz(d); TrX TrY TrZ]

%% generation of saddle surface and measured points
[X,Y] = meshgrid(-50:5:50,-50:5:50);
Z = (X.^2 - Y.^2) / 250;
n1= 19; % number of points in X direction
n2= 19; % number of points in Y direction
x_pts = 2*49*rand(n1, n2, 1)-49;
y_pts = 2*49*rand(n1, n2, 1)-49;
m= 100; % number of repetition
for k= 1:m
    dev= 0.1 * rand(n1,n2,1)-0.05;
    z_pts= (x_pts.^2 - y_pts.^2)/250 + dev;
    N= n1*n2;
end

```

```

x_vector= reshape(x_pts,[N, 1]);
y_vector= reshape(y_pts,[N, 1]);
z_vector= reshape(z_pts,[N, 1]);
MP = [x_vector, y_vector, z_vector];

%% Fitting the measured points to the desired surface (minimum delta Z)
[SS0, TP, DZ]= Min_DZ(RT0); % outputs: sum of squares of DZs,
Transferred points, DZs and XYs
options = optimset('Display', 'iter', 'TolX', 1e-9);
RT_Min_DZ(1:2,1:3,k) = fminsearch(@Min_DZ, RT0, options);
[SS_MZ, TP_MZ , DZ_MZ]= Min_DZ(RT_Min_DZ(1:2,1:3,k));

%% Fitting the measured points to the desired surface (Least Squares
method)
[SS1, TP1, Dist1]= Min_Dist(RT_Min_DZ(1:2,1:3,k)); % outputs: sum of
squares of normal distances, Transferred points & distances
options = optimset('Display', 'iter', 'TolX', 1e-12);
RT_Min_Dist(1:2,1:3,k) = fminsearch(@Min_Dist, RT_Min_DZ(1:2,1:3,k),
options); % optimum rotation and translation amounts
[SS_MD, TP_MD , Dist_MD]= Min_Dist(RT_Min_Dist(1:2,1:3,k));

range_dev_LS(k)= max(Dist_MD)- min(Dist_MD);

%% Fitting the measured points to the desired surface (Weighted Least
Squares method)
[SS2, TP2, W_Dist2,Dist2]= Min_Weighted_Dist(RT_Min_Dist(1:2,1:3,k)); %
outputs: sum of weighted normal distances, Transferred points, weight
of distances
options = optimset('Display', 'iter', 'TolX', 1e-12);
RT_Min_W_Dist(1:2,1:3,k) = fminsearch(@Min_Weighted_Dist,
RT_Min_Dist(1:2,1:3,k), options);
[SS_MWD, TP_MWD , W_Dist_MWD , Dist_MWD]=
Min_Weighted_Dist(RT_Min_W_Dist(1:2,1:3,k));

range_dev_WLS(k)= max(Dist_MWD)- min(Dist_MWD);

%% Fitting the measured points to the desired surface (Highly Weighted
Least Squares method)
%[SS3, TP3, W_Dist3,Dist3]=
Min_Highly_Weighted_Dist(RT_Min_W_Dist(1:2,1:3,k)); % outputs: sum of
squares of weighted normal distances, Transferred points & weight of
distances
options = optimset('Display', 'iter', 'TolX', 1e-12, 'TolFun', 1e-2);
RT_Min_HW_Dist(1:2,1:3,k) = fminsearch(@Min_Highly_Weighted_Dist,
RT_Min_W_Dist(1:2,1:3,k), options);
[SS_MHWD, TP_MHWD , W_Dist_MHWD , Dist_MHWD]=
Min_Highly_Weighted_Dist(RT_Min_HW_Dist(1:2,1:3,k));

range_dev_HWLS(k)= max(Dist_MHWD)- min(Dist_MHWD);

end
%%
LS= reshape(RT_Min_Dist,[m, 6]);
WLS= reshape(RT_Min_W_Dist,[m, 6]);

```

```

HWLS= reshape(RT_Min_HW_Dist,[m, 6]);

%% Point residuals
figure;
scatter3(TP_MD(:,1),TP_MD(:,2),Dist_MD,10,Dist_MD);
colormap(jet);
colorbar;
xlabel('X(mm)');
ylabel('Y(mm)');
zlabel('Normal deviation(mm)');
title('LS');

figure;
scatter3(TP_MWD(:,1),TP_MWD(:,2),Dist_MWD,10,Dist_MWD);
colormap(jet);
colorbar;
xlabel('X(mm)');
ylabel('Y(mm)');
zlabel('Normal deviation(mm)');
title('WLS');

figure;
scatter3(TP_MHWD(:,1),TP_MHWD(:,2),Dist_MHWD,10,Dist_MHWD);
colormap(jet);
colorbar;
xlabel('X(mm)');
ylabel('Y(mm)');
zlabel('Normal deviation(mm)');
title('HWLS');

%% Deviation filtering and calculation of mean deviations
[MLD_LS, fil_XY_LS]= Area_filter_square(Dist_MD, TP_MD);
[MLD_WLS, fil_XY_WLS]= Area_filter_square(Dist_MWD, TP_MWD);
[MLD_HWLS, fil_XY_HWLS]= Area_filter_square(Dist_MHWD, TP_MHWD);

figure;
scatter3(fil_XY_LS(:,1),fil_XY_LS(:,2),MLD_LS,10,MLD_LS);
colormap(jet);
colorbar;
xlabel('X(mm)');
ylabel('Y(mm)');
zlabel('Mean Local Deviation(mm)');
title(' LS fitting');

figure;
scatter3(fil_XY_WLS(:,1),fil_XY_WLS(:,2),MLD_WLS,10,MLD_WLS);
colormap(jet);
colorbar;
xlabel('X(mm)');
ylabel('Y(mm)');
zlabel('Mean Local Deviation(mm)');
title(' WLS fitting');

figure;
scatter3(fil_XY_HWLS(:,1),fil_XY_HWLS(:,2),MLD_HWLS,10,MLD_HWLS);

```

```

colormap(jet);
colorbar;
xlabel('X(mm) ');
ylabel('Y(mm) ');
zlabel('Mean Local Deviation(mm) ');
title(' HWLS fitting');

%% calculation of differences from LS data
Difference1= MLD_LS - MLD_WLS;
figure;
scatter3(fil_XY_WLS(:,1),fil_XY_WLS(:,2),Difference1,10,Difference1);
colormap(jet);
colorbar;
xlabel('X(mm) ');
ylabel('Y(mm) ');
zlabel('Difference1(mm) ');
title('Difference between LS & WLS');

Difference2= MLD_LS - MLD_HWLS;
figure;
scatter3(fil_XY_HWLS(:,1),fil_XY_HWLS(:,2),Difference2,10,Difference2);
colormap(jet);
colorbar;
xlabel('X(mm) ');
ylabel('Y(mm) ');
zlabel('Difference2(mm) ');
title('Difference between LS & HWLS');

%% generation of CDF (Cumulative Distribution Function)plots of
deviation ranges
figure;
cdfplot(range_dev_LS);
hold on
cdfplot(range_dev_WLS);
cdfplot(range_dev_HWLS);

```

Functions

```

function [SS, tpts, DZs, XYs]= Min_DZ(RT)
global N
global CP

alpha= RT(1,1);    % rotation around X in degrees
beta= RT(1,2);    % rotation around Y in degrees
theta= RT(1,3);   % rotation around Z in degrees
Rx= [1 0 0; 0 cosd(alpha) -sind(alpha); 0 sind(alpha) cosd(alpha)];
Ry= [cosd(beta) 0 sind(beta); 0 1 0; -sind(beta) 0 cosd(beta)];
Rz= [cosd(theta) -sind(theta) 0; sind(theta) cosd(theta) 0; 0 0 1];
Rot= Rx*Ry*Rz;

dx= RT(2,1);
dy= RT(2,2);
dz= RT(2,3);

for i= 1:N

```

```

    tpts(i,1:3)= Rot*CP(i,1:3)+'+[dx dy dz]';
    XYs(i,1:2)= tpts(i,1:2);
    [t,r]= cart2pol(XYs(i,1), XYs(i,2));
    Z(i)= 1/75000*r^3*(cos(t)+sin(5*t));
    DZs(i)= tpts(i,3)-Z(i);
end
SS= sumsqr(DZs);
end

function [SS, tpts, Dist]= Min_Dist(RT)
global N
global MP

alpha= RT(1,1);    % rotation around X in degrees
beta= RT(1,2);    % rotation around Y in degrees
theta= RT(1,3);   % rotation around Z in degrees
Rx= [1 0 0; 0 cosd(alpha) -sind(alpha); 0 sind(alpha) cosd(alpha)];
Ry= [cosd(beta) 0 sind(beta); 0 1 0; -sind(beta) 0 cosd(beta)];
Rz= [cosd(theta) -sind(theta) 0; sind(theta) cosd(theta) 0; 0 0 1];
Rot= Rx*Ry*Rz;

dx= RT(2,1);
dy= RT(2,2);
dz= RT(2,3);

for i= 1:N
    tpts(i,1:3)= Rot*MP(i,1:3)+'+[dx dy dz]';
    X(i)= tpts(i,1);
    Y(i)= tpts(i,2);
    Z(i)= (X(i)^2 - Y(i)^2)/250;
    DZ= [0,0,tpts(i,3)-Z(i)];
    norm_vec= [X(i)/125, Y(i)/125,-1];
    n= norm_vec/norm(norm_vec);
    if n(3)<0
        n= -n;
    end
    Dist(i)= dot(DZ,n);
end
SS= sumsqr(Dist);
end

function [SS, tpts, W, norm_dist]= Min_Weighted_Dist(RT)
global N
global MP

alpha= RT(1,1);    % rotation around X in degrees
beta= RT(1,2);    % rotation around Y in degrees
theta= RT(1,3);   % rotation around Z in degrees
Rx= [1 0 0; 0 cosd(alpha) -sind(alpha); 0 sind(alpha) cosd(alpha)];
Ry= [cosd(beta) 0 sind(beta); 0 1 0; -sind(beta) 0 cosd(beta)];
Rz= [cosd(theta) -sind(theta) 0; sind(theta) cosd(theta) 0; 0 0 1];
Rot= Rx*Ry*Rz;

dx= RT(2,1);
dy= RT(2,2);
dz= RT(2,3);

```

```

for i= 1:N
    tpts(i,1:3)= Rot*MP(i,1:3)+'[dx dy dz]';
    X(i)= tpts(i,1);
    Y(i)= tpts(i,2);
    Z(i)= (X(i)^2 - Y(i)^2)/250;
    DZ= [0,0,tpts(i,3)-Z(i)];
    norm_vec= [X(i)/125, Y(i)/125,-1];
    n= norm_vec/norm(norm_vec);
    if n(3)<0
        n= -n;
    end
    norm_dist(i)= dot(DZ,n);
end
Tri= delaunayTriangulation(MP(:,1),MP(:,2));
m= size(Tri,1);
W= zeros(1,N);
for j=1:m
    S=0;
    r1= Tri(j,1);
    r2= Tri(j,2);
    r3= Tri(j,3);
    P1= MP(r1,:);
    P2= MP(r2,:);
    P3= MP(r3,:);
    V1= P2-P1;
    V2= P3-P1;
    S= 0.5*norm(cross(V1,V2));
    W(r1)= W(r1)+S/3;
    W(r2)= W(r2)+S/3;
    W(r3)= W(r3)+S/3;
    clear r1 r2 r3 P1 P2 P3 V1 V2
end

SS= sum(W.*norm_dist.^2);
end

function [SS, tpts, W, norm_dist]= Min_Highly_Weighted_Dist(RT)
global N
global MP

alpha= RT(1,1);    % rotation around X in degrees
beta= RT(1,2);    % rotation around Y in degrees
theta= RT(1,3);   % rotation around Z in degrees
Rx= [1 0 0; 0 cosd(alpha) -sind(alpha); 0 sind(alpha) cosd(alpha)];
Ry= [cosd(beta) 0 sind(beta); 0 1 0; -sind(beta) 0 cosd(beta)];
Rz= [cosd(theta) -sind(theta) 0; sind(theta) cosd(theta) 0; 0 0 1];
Rot= Rx*Ry*Rz;

dx= RT(2,1);
dy= RT(2,2);
dz= RT(2,3);

for i= 1:N
    tpts(i,1:3)= Rot*MP(i,1:3)+'[dx dy dz]';
    X(i)= tpts(i,1);

```

```

    Y(i)= tpts(i,2);
    Z(i)= (X(i)^2 - Y(i)^2)/250;
    DZ= [0,0,tpts(i,3)-Z(i)];
    norm_vec= [X(i)/125, Y(i)/125,-1];
    n= norm_vec/norm(norm_vec);
    if n(3)<0
        n= -n;
    end
    norm_dist(i)= dot(DZ,n);
end
Tri= delaunayTriangulation(MP(:,1),MP(:,2));
m= size(Tri,1);
W= zeros(1,N);
for j=1:m
    S=0;
    r1= Tri(j,1);
    r2= Tri(j,2);
    r3= Tri(j,3);
    P1= MP(r1,:);
    P2= MP(r2,:);
    P3= MP(r3,:);
    V1= P2-P1;
    V2= P3-P1;
    S= 0.5*norm(cross(V1,V2));
    W(r1)= W(r1)+S/3;
    W(r2)= W(r2)+S/3;
    W(r3)= W(r3)+S/3;
    clear r1 r2 r3 P1 P2 P3 V1 V2
end

W_Dist= W.*norm_dist;
SS= sumsqr(W_Dist);
end

function [dev, fil_XY]= Area_filter_square(Dist, XY)
global N

seg_dim= 12; % dimension of segments
step_len= 4; % step length (along x & y)
x_min= -50; x_max= 50; % x limits of projected surface
y_min= -50; y_max= 50; % y limits of projected surface

seg_no= 0; % segment number

for y_up = (y_min + seg_dim): step_len : y_max % upper y of every
segment
    y_low= y_up - seg_dim; % lower y of every segment
    y_center= (y_low + y_up)/2; % y of center point
    for x_up = (x_min + seg_dim): step_len : x_max % upper x of every
segment
        seg_no= seg_no + 1;
        x_low= x_up - seg_dim; % lower x of every segment
        x_center= (x_up + x_low)/2; % x of center point
        S=0; % sum of deviations inside the segment
        m=0; % number of points inside the segment
        for i= 1:N

```

```

        if (XY(i,1) >= x_low) && (XY(i,1)< x_up) && (XY(i,2) >=
y_low) && (XY(i,2)<y_up)
            S= S + Dist(i);
            m= m+1;
        end
    end
    if S==0
        dev(seg_no,1)=0;
    else
        dev(seg_no,1)= S/m; % average of deviations related to
points in the segment
    end
    fil_XY(seg_no,1:2)= [x_center, y_center]; % x & y coordinates
of center of segment
    end
end
end

```

Analysis of data originated in the tactile measurement of the wavy profile

Main program

```

%% Clearing workspace and command window
clear all; clc; close all

%% Definition of variables
global D % Offset distance (probe diameter)
global points % Captured points
global N % Number of points
global CP % Contact points
RT0 = [0,0,0; 0,0,-10]; % Initial translation and rotation of points
[Rotx(d) Roty(d) Rotz(d); TrX TrY TrZ]

%% Importing the coordinates of points from text file
text = 'Tactile- CMM.txt';
[points]=import_data(text);
N= length(points);
clearvars text;

%% Finding the coordinates of contact points
D= 3;
for i= 1:N
    for j= 1:N
        dist(j)= norm(points(i,1:3)-points(j,1:3));
    end
    sort_dist= sort(dist);
    short_dis= sort_dist(2:4);
    row_num= [];
    for j= 1:3
        row_num= [ row_num, find(dist==short_dis(j))];
    end
    pt1= points(row_num(1),1:3);
    pt2= points(row_num(2),1:3);
    pt3= points(row_num(3),1:3);
    norm_vec(1:3)= cross((pt2-pt1),(pt3-pt1));
    if norm_vec(3)> 0
        norm_vec= -norm_vec;
    end
end

```

```

    end
    CP(i,1:3)= points(i,1:3)+D/(norm(norm_vec))*norm_vec;
    clear dist
end

%% Fitting the measured points to the desired surface (minimum delta Z)
[SS0, TP, DZ, XY]= Min_DZ(RT0); % outputs: sum of squares of DZs,
Transferred points, DZs and XYs
options = optimset('Display', 'iter');
RT_Min_DZ = fminsearch(@Min_DZ, RT0, options);

[SS_MZ, TP_MZ , DZ_MZ , XY_MZ]= Min_DZ(RT_Min_DZ);

%% Fitting the measured points to the desired surface (LS)
[SS1, TP1, Dist1, XY1]= Least_Sqr(RT_Min_DZ); % outputs: sum of squares
of normal distances, Transferred points, distances and XYs
options = optimset('Display', 'iter', 'TolX', 1e-12);
RT_LS = fminsearch(@Least_Sqr, RT_Min_DZ, options);

[SS_LS, TP_LS , Dist_LS , XY_LS]= Least_Sqr(RT_LS);

%% Fitting the measured points to the desired surface (WLS)
[SS2, TP2, WDist2,Dist2, XY2]= Weighted_Least_Sqr(RT_LS); % outputs:
sum of squares of normal distances, Transferred points, weighted
distances and XYs
options = optimset('Display', 'iter', 'TolX', 1e-12);
RT_WLS = fminsearch(@Weighted_Least_Sqr, RT_LS, options);

[SS_WLS, TP_WLS , W_Dist_WLS , Dist_WLS, XY_WLS]=
Weighted_Least_Sqr(RT_WLS);

%% Fitting the measured points to the desired surface (HWLS)
[SS3, TP3, WDist3,Dist3, XY3]= Highly_Weighted_Least_Sqr(RT_LS); %
outputs: sum of squares of normal distances, Transferred points,
weighted distances and XYs
options = optimset('Display', 'iter', 'TolX', 1e-12);
RT_HWLS = fminsearch(@Highly_Weighted_Least_Sqr, RT_LS, options);

[SS_HWLS, TP_HWLS , W_Dist_HWLS , Dist_HWLS, XY_HWLS]=
Weighted_Least_Sqr(RT_HWLS);

%% Point residuals
figure;
scatter3(TP_MZ(:,1),TP_MZ(:,2),DZ_MZ,10,DZ_MZ);
colormap(jet);
colorbar;
xlabel('X(mm) ');
ylabel('Y(mm) ');
zlabel('Delta Z(mm) ');
title('Minimum deltaZ fitting');
figure;
scatter3(TP_LS(:,1),TP_LS(:,2),Dist_LS,10,Dist_LS);
colormap(jet);
colorbar;
xlabel('X(mm) ');

```

```

ylabel('Y(mm) ');
xlabel('Normal deviation(mm) ');
title('LS fitting');
figure;
scatter3(TP_WLS(:,1),TP_WLS(:,2),Dist_WLS,10,Dist_WLS);
colormap(jet);
colorbar;
xlabel('X(mm) ');
ylabel('Y(mm) ');
xlabel('Normal deviation(mm) ');
title('WLS fitting');
figure;
scatter3(TP_HWLS(:,1),TP_HWLS(:,2),Dist_HWLS,10,Dist_HWLS);
colormap(jet);
colorbar;
xlabel('X(mm) ');
ylabel('Y(mm) ');
xlabel('Normal deviation(mm) ');
title('HWLS fitting');
%% Deviation filtering and calculation of mean deviations
[dev_LS, fil_XY_LS]= Area_filter(Dist_LS, XY_LS);
[dev_WLS, fil_XY_WLS]= Area_filter(Dist_WLS, XY_WLS);
[dev_HWLS, fil_XY_HWLS]= Area_filter(Dist_HWLS, XY_HWLS);

figure;
scatter3(fil_XY_LS(:,1),fil_XY_LS(:,2),dev_LS,10,dev_LS);
colormap(jet);
colorbar;
xlabel('X(mm) ');
ylabel('Y(mm) ');
xlabel('Mean Local Deviation(mm) ');
title(' LS fitting');

figure;
scatter3(fil_XY_WLS(:,1),fil_XY_WLS(:,2),dev_WLS,10,dev_WLS);
colormap(jet);
colorbar;
xlabel('X(mm) ');
ylabel('Y(mm) ');
xlabel('Mean Local Deviation(mm) ');
title('WLS fitting');

figure;
scatter3(fil_XY_HWLS(:,1),fil_XY_HWLS(:,2),dev_HWLS,10,dev_HWLS);
colormap(jet);
colorbar;
xlabel('X(mm) ');
ylabel('Y(mm) ');
xlabel('Mean Local Deviation(mm) ');
title('HWLS fitting');

%% calculation of differences from LS data
Differencel= dev_LS - dev_WLS;
figure;
scatter3(fil_XY_WLS(:,1),fil_XY_WLS(:,2),Differencel,10,Differencel);
colormap(jet);

```

```

colorbar;
xlabel('X (mm) ');
ylabel('Y (mm) ');
zlabel('Difference1 (mm) ');
title('Difference between LS & WLS');

Difference2= dev_LS - dev_HWLS;
figure;
scatter3(fil_XY_HWLS(:,1),fil_XY_HWLS(:,2),Difference2,10,Difference2);
colormap(jet);
colorbar;
xlabel('X (mm) ');
ylabel('Y (mm) ');
zlabel('Difference2 (mm) ');
title('Difference between LS & HWLS');

```

Functions

```

function [SS, tpts, DZs, XYs]= Min_DZ(RT)
global N
global CP

alpha= RT(1,1);    % rotation around X in degrees
beta= RT(1,2);    % rotation around Y in degrees
theta= RT(1,3);   % rotation around Z in degrees
Rx= [1 0 0; 0 cosd(alpha) -sind(alpha); 0 sind(alpha) cosd(alpha)];
Ry= [cosd(beta) 0 sind(beta); 0 1 0; -sind(beta) 0 cosd(beta)];
Rz= [cosd(theta) -sind(theta) 0; sind(theta) cosd(theta) 0; 0 0 1];
Rot= Rx*Ry*Rz;

dx= RT(2,1);
dy= RT(2,2);
dz= RT(2,3);

for i= 1:N
    tpts(i,1:3)= Rot*CP(i,1:3)+'+[dx dy dz]';
    XYs(i,1:2)= tpts(i,1:2);
    [t,r]= cart2pol(XYs(i,1), XYs(i,2));
    Z(i)= 1/75000*r^3*(cos(t)+sin(5*t));
    DZs(i)= tpts(i,3)-Z(i);
end
SS= sumsqr(DZs);
end

function [SS, tpts, Dist, XYs]= Least_Sqr(RT)
global N
global CP

alpha= RT(1,1);    % rotation around X in degrees
beta= RT(1,2);    % rotation around Y in degrees
theta= RT(1,3);   % rotation around Z in degrees
Rx= [1 0 0; 0 cosd(alpha) -sind(alpha); 0 sind(alpha) cosd(alpha)];
Ry= [cosd(beta) 0 sind(beta); 0 1 0; -sind(beta) 0 cosd(beta)];
Rz= [cosd(theta) -sind(theta) 0; sind(theta) cosd(theta) 0; 0 0 1];
Rot= Rx*Ry*Rz;

```

```

dx= RT(2,1);
dy= RT(2,2);
dz= RT(2,3);

for i= 1:N
    tpts(i,1:3)= Rot*CP(i,1:3)+'+[dx dy dz]';
    XYs(i,1:2)= tpts(i,1:2);
    [t,r]= cart2pol(XYs(i,1), XYs(i,2));
    Z= 1/75000*r^3*(cos(t)+sin(5*t));
    DZ= [0,0,tpts(i,3)-Z];
    norm_vec= [3/75000*r^2*(cos(t)+sin(5*t)), 1/75000*r^3*(-
sin(t)+5*cos(5*t)), -1];
    n= norm_vec/norm(norm_vec);
    if n(3)<0
        n= -n;
    end
    Dist(i)= dot(DZ,n);
end
SS= sumsqr(Dist);
end

function [SS, tpts, W_Dist, norm_dist, XYs]= Weighted_Least_Sqr(RT)
global N
global CP

alpha= RT(1,1);    % rotation around X in degrees
beta= RT(1,2);    % rotation around Y in degrees
theta= RT(1,3);   % rotation around Z in degrees
Rx= [1 0 0; 0 cosd(alpha) -sind(alpha); 0 sind(alpha) cosd(alpha)];
Ry= [cosd(beta) 0 sind(beta); 0 1 0; -sind(beta) 0 cosd(beta)];
Rz= [cosd(theta) -sind(theta) 0; sind(theta) cosd(theta) 0; 0 0 1];
Rot= Rx*Ry*Rz;

dx= RT(2,1);
dy= RT(2,2);
dz= RT(2,3);

for i= 1:N
    tpts(i,1:3)= Rot*CP(i,1:3)+'+[dx dy dz]';
    XYs(i,1:2)= tpts(i,1:2);
    [t,r]= cart2pol(XYs(i,1), XYs(i,2));
    Z= 1/75000*r^3*(cos(t)+sin(5*t));
    DZ= [0,0,tpts(i,3)-Z];
    norm_vec= [3/75000*r^2*(cos(t)+sin(5*t)), 1/75000*r^3*(-
sin(t)+5*cos(5*t)), -1];
    n= norm_vec/norm(norm_vec);
    if n(3)<0
        n= -n;
    end
    norm_dist(i)= dot(DZ,n);
end
Tri= delaunayTriangulation(CP(:,1),CP(:,2));
m= size(Tri,1);
W= zeros(1,N);
for j=1:m

```

```

S=0;
r1= Tri(j,1);
r2= Tri(j,2);
r3= Tri(j,3);
P1= CP(r1,:);
P2= CP(r2,:);
P3= CP(r3,:);
V1= P2-P1;
V2= P3-P1;
S= 0.5*norm(cross(V1,V2));
W(r1)= W(r1)+S/3;
W(r2)= W(r2)+S/3;
W(r3)= W(r3)+S/3;
clear r1 r2 r3 P1 P2 P3 V1 V2
end

W_Dist= W.*(norm_dist.^2);
SS= sum(W_Dist);
end

function [SS, tpts, W_Dist, norm_dist, XYs]= Weighted_Least_Sqr(RT)
global N
global CP

alpha= RT(1,1); % rotation around X in degrees
beta= RT(1,2); % rotation around Y in degrees
theta= RT(1,3); % rotation around Z in degrees
Rx= [1 0 0; 0 cosd(alpha) -sind(alpha); 0 sind(alpha) cosd(alpha)];
Ry= [cosd(beta) 0 sind(beta); 0 1 0; -sind(beta) 0 cosd(beta)];
Rz= [cosd(theta) -sind(theta) 0; sind(theta) cosd(theta) 0; 0 0 1];
Rot= Rx*Ry*Rz;

dx= RT(2,1);
dy= RT(2,2);
dz= RT(2,3);

for i= 1:N
tpts(i,1:3)= Rot*CP(i,1:3)+'+[dx dy dz]';
XYs(i,1:2)= tpts(i,1:2);
[t,r]= cart2pol(XYs(i,1), XYs(i,2));
Z= 1/75000*r^3*(cos(t)+sin(5*t));
DZ= [0,0,tpts(i,3)-Z];
norm_vec= [3/75000*r^2*(cos(t)+sin(5*t)), 1/75000*r^3*(-
sin(t)+5*cos(5*t)), -1];
n= norm_vec/norm(norm_vec);
if n(3)<0
n= -n;
end
norm_dist(i)= dot(DZ,n);
end

Tri= delaunayTriangulation(CP(:,1),CP(:,2));
m= size(Tri,1);
W= zeros(1,N);
for j=1:m
S=0;
r1= Tri(j,1);

```

```

r2= Tri(j,2);
r3= Tri(j,3);
P1= CP(r1,:);
P2= CP(r2,:);
P3= CP(r3,:);
V1= P2-P1;
V2= P3-P1;
S= 0.5*norm(cross(V1,V2));
W(r1)= W(r1)+S/3;
W(r2)= W(r2)+S/3;
W(r3)= W(r3)+S/3;
clear r1 r2 r3 P1 P2 P3 V1 V2
end

W_Dist= W.*norm_dist;
SS= sumsqr(W_Dist);
end

function [dev, fil_XY]= Area_filter(Dist, XY)
global N
for k=1:N
    [t(k),r(k)]= cart2pol(XY(k,1), XY(k,2));
    if t(k)<0
        t(k)= t(k)+2*pi;
    end
end
v=0; % counter of middle points (sections)
for i=1:10
    Ri= 5*i-5; % inner radius
    Ro= 5*i+5; % outer radius
    Rm= 5*i; % middle radius
    alpha= 2*pi*25/(Ro^2-Ri^2); % angle of circular sector
    R= mod(2*pi,2/3*alpha); % calculation of quotient and remainder
    if R==0
        n= (2*pi)/(2/3*alpha); % # of steps in a whole round
    else
        n= fix((2*pi)/(2/3*alpha))+1;
    end
    for j= 1:n
        sa= 2/3*alpha*(j-1); % start angle
        ea= 2/3*alpha*(j+0.5); % end angle
        ma= 0.5*(sa+ea); % middle angle
        v= v+1;
        [fil_XY(v,1),fil_XY(v,2)]= pol2cart(ma,Rm);
        S=0; % sum of deviations inside the section
        m=0; % # of points inside the section
        for k= 1:N
            if (r(k)>=Ri) && (r(k)<=Ro)
                if (t(k)>=sa) && (t(k)<=ea)
                    S= S+ Dist(k);
                    m= m+1;
                end
            end
        end
        if S==0
            dev(v,1)=0
        end
    end
end

```

```

        else
            dev(v,1)= S/m;
        end
    end
end
end

```

Analysis of data measured with the Romer arm

Wavy profile

Main program

```

%% Clearing workspace and command window
clear all; clc; close all

%% Definition of variables
global N % Number of points
global CP % Contact points
RT0 = [0,0,0; 0,0,-11]; % Initial translation and rotation of points
[Rotx(d) Roty(d) Rotz(d); TrX TrY TrZ]

%% Importing the coordinates of points from text file
text = 'Scan1- cross.txt';
[points]=import_data(text);
pc_w_noise= pointCloud(points);
pc_denoised= pcdenoise(pc_w_noise); % removing noise from the data
pts_denoised= pc_denoised.Location; % extracting the point coordinates
from the denoised array

%% removing points that are on the edge area
nop= length(pts_denoised);
m = fix(nop/130); % number of randomly selected points
s=0; % counter of points which are out of desired region
for i= 1:m
    nor= 130*(i-1)+round(rand*130); % row number of the selected point
    if nor == 0
        nor= 1;
    end
    [t,r]= cart2pol(pts_denoised(nor,1), pts_denoised(nor,2));
    random_points(i,1:3)= pts_denoised(nor,1:3);
    if r > 49.7
        s= s+1;
        d(s)= i;
    end
end
end
CP= removerows(random_points, 'ind',d);
N= length(CP);
clearvars text points random_points t r i d s;

%% Fitting the measured points to the desired surface (minimum delta Z)

[SS0, TP, DZ, XY]= Min_DZ(RT0); % outputs: sum of squares of DZs,
Transferred points, DZs and XYs
options = optimset('Display', 'iter');
RT_Min_DZ = fminsearch(@Min_DZ, RT0, options);

```

```

[SS_MZ, TP_MZ , DZ_MZ , XY_MZ]= Min_DZ(RT_Min_DZ);

%% Fitting the measured points to the desired surface (LS)
[SS1, TP1, Dist1, XY1]= Least_Sqr(RT_Min_DZ); % outputs: sum of squares
of normal distances, Transferred points, distances and XYs
options = optimset('Display', 'iter', 'TolX', 1e-12);
RT_LS = fminsearch(@Least_Sqr, RT_Min_DZ, options);

[SS_LS, TP_LS , Dist_LS , XY_LS]= Least_Sqr(RT_LS);

%% Fitting the measured points to the desired surface (WLS)
[SS2, TP2, WDist2,Dist2, XY2]= Weighted_Least_Sqr(RT_LS); % outputs:
sum of squares of normal distances, Transferred points, weighted
distances and XYs
options = optimset('Display', 'iter', 'TolX', 1e-12);
RT_WLS = fminsearch(@Weighted_Least_Sqr, RT_LS, options);

[SS_WLS, TP_WLS , W_Dist_WLS , Dist_WLS, XY_WLS]=
Weighted_Least_Sqr(RT_WLS);

%% Fitting the measured points to the desired surface (HWLS)
[SS3, TP3, WDist3,Dist3, XY3]= Highly_Weighted_Least_Sqr(RT_LS); %
outputs: sum of squares of normal distances, Transferred points,
weighted distances and XYs
options = optimset('Display', 'iter', 'TolX', 1e-12);
RT_HWLS = fminsearch(@Highly_Weighted_Least_Sqr, RT_LS, options);

[SS_HWLS, TP_HWLS , W_Dist_HWLS , Dist_HWLS, XY_HWLS]=
Highly_Weighted_Least_Sqr(RT_HWLS);

%% Point residuals

figure;
scatter3(TP_MZ(:,1),TP_MZ(:,2),DZ_MZ,10,DZ_MZ);
colormap(jet);
colorbar;
xlabel('X(mm)');
ylabel('Y(mm)');
zlabel('Delta Z(mm)');
title('Minimum deltaZ fitting');
figure;
scatter3(TP_LS(:,1),TP_LS(:,2),Dist_LS,10,Dist_LS);
colormap(jet);
colorbar;
xlabel('X(mm)');
ylabel('Y(mm)');
zlabel('Normal deviation(mm)');
title('LS fitting');
figure;
scatter3(TP_WLS(:,1),TP_WLS(:,2),Dist_WLS,10,Dist_WLS);
colormap(jet);
colorbar;
xlabel('X(mm)');
ylabel('Y(mm)');

```

```

xlabel('Normal deviation(mm) ');
title('WLS fitting');
figure;
scatter3(TP_HWLS(:,1),TP_HWLS(:,2),Dist_HWLS,10,Dist_HWLS);
colormap(jet);
colorbar;
xlabel('X(mm) ');
ylabel('Y(mm) ');
xlabel('Normal deviation(mm) ');
title('HWLS fitting');
%% Deviation filtering and calculation of mean deviations
[dev_LS, fil_XY_LS]= Area_filter(Dist_LS, XY_LS);
[dev_WLS, fil_XY_WLS]= Area_filter(Dist_WLS, XY_WLS);
[dev_HWLS, fil_XY_HWLS]= Area_filter(Dist_HWLS, XY_HWLS);

figure;
scatter3(fil_XY_LS(:,1),fil_XY_LS(:,2),dev_LS,10,dev_LS);
colormap(jet);
colorbar;
xlabel('X(mm) ');
ylabel('Y(mm) ');
xlabel('Mean Local Deviation(mm) ');
title(' LS fitting');

figure;
scatter3(fil_XY_WLS(:,1),fil_XY_WLS(:,2),dev_WLS,10,dev_WLS);
colormap(jet);
colorbar;
xlabel('X(mm) ');
ylabel('Y(mm) ');
xlabel('Local mean normal deviation(mm) ');
title('WLS fitting');

figure;
scatter3(fil_XY_HWLS(:,1),fil_XY_HWLS(:,2),dev_HWLS,10,dev_HWLS);
colormap(jet);
colorbar;
xlabel('X(mm) ');
ylabel('Y(mm) ');
xlabel('Local mean normal deviation(mm) ');
title('HWLS fitting');

%% calculation of differences from LS data
Difference1= dev_LS - dev_WLS;
figure;
scatter3(fil_XY_WLS(:,1),fil_XY_WLS(:,2),Difference1,10,Difference1);
colormap(jet);
colorbar;
xlabel('X(mm) ');
ylabel('Y(mm) ');
xlabel('Difference1(mm) ');
title('Difference between LS & WLS');

Difference2= dev_LS - dev_HWLS;
figure;
scatter3(fil_XY_HWLS(:,1),fil_XY_HWLS(:,2),Difference2,10,Difference2);

```

```

colormap(jet);
colorbar;
xlabel('X(mm)');
ylabel('Y(mm)');
zlabel('Difference2(mm)');
title('Difference between LS & HWLS');

```

Functions

```

function [pts]= import_data (filename)

file_id = fopen(filename, 'r');
nums = fscanf(file_id, '%f %f %f,');
n= length(nums)/3;
for i=1:n
    pts(i,1:3)=[nums(3*i-2),nums(3*i-1),nums(3*i)];
end

fclose(file_id);

function [SS, tpts, DZs, XYs]= Min_DZ(RT)
global N
global CP

alpha= RT(1,1);    % rotation around X in degrees
beta= RT(1,2);    % rotation around Y in degrees
theta= RT(1,3);   % rotation around Z in degrees
Rx= [1 0 0; 0 cosd(alpha) -sind(alpha); 0 sind(alpha) cosd(alpha)];
Ry= [cosd(beta) 0 sind(beta); 0 1 0; -sind(beta) 0 cosd(beta)];
Rz= [cosd(theta) -sind(theta) 0; sind(theta) cosd(theta) 0; 0 0 1];
Rot= Rx*Ry*Rz;

dx= RT(2,1);
dy= RT(2,2);
dz= RT(2,3);

for i= 1:N
    tpts(i,1:3)= Rot*CP(i,1:3)+'[dx dy dz]';
    XYs(i,1:2)= tpts(i,1:2);
    [t,r]= cart2pol(XYs(i,1), XYs(i,2));
    Z(i)= 1/75000*r^3*(cos(t)+sin(5*t));
    DZs(i)= tpts(i,3)-Z(i);
end
SS= sumsqr(DZs);
end

function [SS, tpts, Dist, XYs]= Least_Sqr(RT)
global N
global CP

alpha= RT(1,1);    % rotation around X in degrees
beta= RT(1,2);    % rotation around Y in degrees
theta= RT(1,3);   % rotation around Z in degrees
Rx= [1 0 0; 0 cosd(alpha) -sind(alpha); 0 sind(alpha) cosd(alpha)];
Ry= [cosd(beta) 0 sind(beta); 0 1 0; -sind(beta) 0 cosd(beta)];

```

```

Rz= [cosd(theta) -sind(theta) 0; sind(theta) cosd(theta) 0; 0 0 1];
Rot= Rx*Ry*Rz;

dx= RT(2,1);
dy= RT(2,2);
dz= RT(2,3);

for i= 1:N
    tpts(i,1:3)= Rot*CP(i,1:3)+'[dx dy dz]';
    XYs(i,1:2)= tpts(i,1:2);
    [t,r]= cart2pol(XYs(i,1), XYs(i,2));
    Z= 1/75000*r^3*(cos(t)+sin(5*t));
    DZ= [0,0,tpts(i,3)-Z];
    norm_vec= [3/75000*r^2*(cos(t)+sin(5*t)), 1/75000*r^3*(-
sin(t)+5*cos(5*t)), -1];
    n= norm_vec/norm(norm_vec);
    if n(3)<0
        n= -n;
    end
    Dist(i)= dot(DZ,n);
end
SS= sumsqr(Dist);
end

function [SS, tpts, W_Dist, norm_dist, XYs]= Weighted_Least_Sqr(RT)
global N
global CP

alpha= RT(1,1);    % rotation around X in degrees
beta= RT(1,2);    % rotation around Y in degrees
theta= RT(1,3);    % rotation around Z in degrees
Rx= [1 0 0; 0 cosd(alpha) -sind(alpha); 0 sind(alpha) cosd(alpha)];
Ry= [cosd(beta) 0 sind(beta); 0 1 0; -sind(beta) 0 cosd(beta)];
Rz= [cosd(theta) -sind(theta) 0; sind(theta) cosd(theta) 0; 0 0 1];
Rot= Rx*Ry*Rz;

dx= RT(2,1);
dy= RT(2,2);
dz= RT(2,3);

for i= 1:N
    tpts(i,1:3)= Rot*CP(i,1:3)+'[dx dy dz]';
    XYs(i,1:2)= tpts(i,1:2);
    [t,r]= cart2pol(XYs(i,1), XYs(i,2));
    Z= 1/75000*r^3*(cos(t)+sin(5*t));
    DZ= [0,0,tpts(i,3)-Z];
    norm_vec= [3/75000*r^2*(cos(t)+sin(5*t)), 1/75000*r^3*(-
sin(t)+5*cos(5*t)), -1];
    n= norm_vec/norm(norm_vec);
    if n(3)<0
        n= -n;
    end
    norm_dist(i)= dot(DZ,n);
end
Tri= delaunayTriangulation(CP(:,1),CP(:,2));
m= size(Tri,1);

```

```

W= zeros(1,N);
for j=1:m
    S=0;
    r1= Tri(j,1);
    r2= Tri(j,2);
    r3= Tri(j,3);
    P1= CP(r1,:);
    P2= CP(r2,:);
    P3= CP(r3,:);
    V1= P2-P1;
    V2= P3-P1;
    S= 0.5*norm(cross(V1,V2));
    W(r1)= W(r1)+S/3;
    W(r2)= W(r2)+S/3;
    W(r3)= W(r3)+S/3;
    clear r1 r2 r3 P1 P2 P3 V1 V2
end

W_Dist= W.*(norm_dist.^2);
SS= sum(W_Dist);
end

function [SS, tpts, W_Dist, norm_dist, XYs]= Weighted_Least_Sqr(RT)
global N
global CP

alpha= RT(1,1);    % rotation around X in degrees
beta= RT(1,2);    % rotation around Y in degrees
theta= RT(1,3);   % rotation around Z in degrees
Rx= [1 0 0; 0 cosd(alpha) -sind(alpha); 0 sind(alpha) cosd(alpha)];
Ry= [cosd(beta) 0 sind(beta); 0 1 0; -sind(beta) 0 cosd(beta)];
Rz= [cosd(theta) -sind(theta) 0; sind(theta) cosd(theta) 0; 0 0 1];
Rot= Rx*Ry*Rz;

dx= RT(2,1);
dy= RT(2,2);
dz= RT(2,3);

for i= 1:N
    tpts(i,1:3)= Rot*CP(i,1:3)+'[dx dy dz]';
    XYs(i,1:2)= tpts(i,1:2);
    [t,r]= cart2pol(XYs(i,1), XYs(i,2));
    Z= 1/75000*r^3*(cos(t)+sin(5*t));
    DZ= [0,0,tpts(i,3)-Z];
    norm_vec= [3/75000*r^2*(cos(t)+sin(5*t)), 1/75000*r^3*(-
sin(t)+5*cos(5*t)),-1];
    n= norm_vec/norm(norm_vec);
    if n(3)<0
        n= -n;
    end
    norm_dist(i)= dot(DZ,n);
end

Tri= delaunayTriangulation(CP(:,1),CP(:,2));
m= size(Tri,1);
W= zeros(1,N);
for j=1:m

```

```

S=0;
r1= Tri(j,1);
r2= Tri(j,2);
r3= Tri(j,3);
P1= CP(r1,:);
P2= CP(r2,:);
P3= CP(r3,:);
V1= P2-P1;
V2= P3-P1;
S= 0.5*norm(cross(V1,V2));
W(r1)= W(r1)+S/3;
W(r2)= W(r2)+S/3;
W(r3)= W(r3)+S/3;
clear r1 r2 r3 P1 P2 P3 V1 V2
end

W_Dist= W.*norm_dist;
SS= sumsqr(W_Dist);
end

function [dev, fil_XY]= Area_filter(Dist, XY)
global N
for k=1:N
    [t(k),r(k)]= cart2pol(XY(k,1), XY(k,2));
    if t(k)<0
        t(k)= t(k)+2*pi;
    end
end
end
v=0; % counter of middle points (sections)
for i=1:10
    Ri= 5*i-5; % inner radius
    Ro= 5*i+5; % outer radius
    Rm= 5*i; % middle radius
    alpha= 2*pi*25/(Ro^2-Ri^2); % angle of circular sector
    R= mod(2*pi,2/3*alpha); % calculation of quotient and remainder
    if R==0
        n= (2*pi)/(2/3*alpha); % # of steps in a whole round
    else
        n= fix((2*pi)/(2/3*alpha))+1;
    end
    for j= 1:n
        sa= 2/3*alpha*(j-1); % start angle
        ea= 2/3*alpha*(j+0.5); % end angle
        ma= 0.5*(sa+ea); % middle angle
        v= v+1;
        [fil_XY(v,1),fil_XY(v,2)]= pol2cart(ma,Rm);
        S=0; % sum of deviations inside the section
        m=0; % # of points inside the section
        for k= 1:N
            if (r(k)>=Ri) && (r(k)<=Ro)
                if (t(k)>=sa) && (t(k)<=ea)
                    S= S+ Dist(k);
                    m= m+1;
                end
            end
        end
    end
end
end
end

```

```
    if S==0
        dev(v,1)=0;
    else
        dev(v,1)= S/m;
    end
end
end
```