A data fusion pipeline for registering point clouds with novel characteristics

Enabling the computer to recognise a pattern without training dataset

Zhongyi Michael Zhang

Supervised by:

Prof. Samanta Piano

Dr. Sofia Catalucci

Dr. Adam Thompson

Prof. Richard Leach

July 2024

Abstract

Data fusion is the technical process which can provide comprehensive information about an object by combining multiple datasets that are collected by different sensors. It has been employed for point cloud registration in the context of optical coordinate measurement, an important subject in metrology. Researchers in this field have proposed numerous methods to improve the performance of data fusion, which can be categorised into user-dependent methods, including Gaussian process (GP) and weighted least-squares (WLS) algorithms, and user-independent methods such as machine learning. Recent research has shown the convenience of deploying GP and WLS and the flexibility and autonomous functionality of machine learning solutions. However, the target scenarios have been focused on point clouds in similar sizes and point densities. This trend leaves room for further innovation in point cloud registration.

In this research project, a new algorithmic pipeline, which is capable of registering two point clouds with the following characteristics contained in a maximum working volume of 500 × 500 × 500 mm, is proposed: 1) the two point clouds are collected from an engineered object via two separate optical measurement systems, i.e. they are located in two uncorrelated coordinate frames; 2) the smaller point cloud shows the surface texture on a small area on the engineered object, which is represented by the larger point

cloud; 3) the point density of the smaller point cloud is > 10 times the point density of the larger point cloud. The challenge lies in the omission of training data: the variation of surface texture is infinite and the area on the engineered object cannot be rigorously determined by the user. As such, the thesis proposes a statistical method to register two point clouds with aforementioned characteristics, which can be summarised as the "geometrical similarity comparison". In the step of detecting the target area, the larger point cloud is subdivided into equally sized subsections (sub-clouds); the geometrical similarity between each sub-cloud and the smaller point cloud is measured via principal component analysis (PCA). The comparison based on PCA will result in the smaller point cloud being located in the target area formed by the selected sub-clouds. Afterwards, the space mutually occupied by the target area and the smaller point cloud is voxelised so that the spatial point distributions of both point clouds can be assessed. The orientation of the smaller point cloud which aligns it to the target area is determined as the correct orientation, and hence completes the whole registration process.

To test the performance of this algorithmic pipeline, three experimental cases were designed with a gradation of geometrical complexity: two cases include synthetic point clouds generated from CAD models and one case in which the point clouds were collected from a coin. The differences of point densities between the pair of point clouds in these cases are in the range of 10 to 10^2. The results indicate that, though manual double-check is needed as the geometries of the test object increases, the algorithmic pipeline is capable of detecting the location in the larger point cloud to register the smaller point cloud. When scanning the point clouds collected from the coin, the most geometrically complex engineered artefact in this research, the pipeline detected the top 0.73% sub-

clouds (186 out of 25,459) which potentially formed the target area in the larger point cloud to register the smaller point cloud.

The pipeline is adequate of detecting the most optimised orientation to register the smaller point cloud regardless of the geometrical complexity of both point clouds. With a 10° interval for 360° orientation attempts, the optimal orientation achieves a registration error ranging from 5% to 0.15%, based on the mean point-to-point distance relative to the smaller point cloud's dimensions.

The directions left for future work are increasing the accuracy of the detection of the target area and renovating the algorithm to reduce the computational cost. Additionally, big-picture topics questioning the empirical methodology of this research are discussed and concluded.

Acknowledgements

At the completion of this research project, I would like to dedicate this thesis to my whole family, including my parents, maternal grandparents, aunt, uncle and cousin. From a materialistic perspective, the expense of my study was several times the average living standard in Mainland China[1] whereas my family have never complaint about the tread-off between the cost and the gain. From a spiritual perspective, they are always taking a pride of my thirst for knowledge and always willing to support me in pursuit of more knowledge.

Then I will pay my gratitude to Prof. Richard Leach, who admitted me into this PhD programme and the research group. Before this happened, I had applied for 109 PhD positions and unfortunately failed all of them, as my undergraduate and postgraduate degrees were regarded as "obsolete" subjects by every research institute. As I was utterly disappointed with my educational background and intellectual aptitude, Prof. Leach offered me the chance to explore a field that piqued my interest. This thesis would have never been presented here if he had rejected my application like other professors did.

[1] The disposable income per capita in China was £ 4,360/year (CNY 39,218 or $ 5,511) in 2023, as shown by this link:
https://english.www.gov.cn/archive/statistics/202401/17/content_WS65a73d26c6d0868f4e8e32e0.html

After Prof. Leach announced his semi-retirement, Prof. Samanta Piano succeeded his position as my primary supervisor at the start of my third year. Although the history of my study under her supervision is not that long, I had spent a remarkably enjoyable 21 months by the time this thesis was finished. During these 21 months, Prof. Piano tolerated and sometimes even welcomed my radical and rebellious thoughts and attitudes towards my research project, life and science. She always gave me the freedom to delve into the direction that I genuinely wanted to go. I must admit that this thesis would not have proposed any new theory without her tolerance and inclusion.

My vice-supervisors, Dr. Sofia Catalucci and Dr. Adam Thompson, should not be slighted to any extent either. Since the beginning of my academic journey, they had been with me all the way to the completion of this thesis. During the past three and half years of study, they gave me the best advice and suggestions and made sure that my posters, papers and this thesis were of the highest quality. Furthermore, the cooperation and casual conversations with them enhanced my ability to debate various topics ranging from error assessment to public policymaking.

Last but not least, I would like to convey my gratitude to my colleagues Mr. Osman Ali, Dr. Waiel Elmadih, Dr. Francisco Ulises Hernandes Ledezma (and Mrs. Gema Ledezma), Dr. Helia Hooshmand (and Mr. Behzad Akhlaghi), Dr. Mohammad Isa, Dr. Xiangjun Kong, Mr. Ahmed Koça, Dr. Athanasios Pappas, Dr. Nelson Pech-May, Dr. Afaf Remani, Dr. Christopher Tompkins, Dr. Wen Wu (and Dr. Pengyu Zhang), Dr. Ruidong Rickon Xue, Dr. Tibebe Yalew and Mr. Mingda Harvey Yang. They celebrated my birthday twice, and we spent countless joyful weekends throughout the grey and gloomy days.

Contents

List of publications

Zhang Z M, Catalucci S, Thompson A, Leach R, Piano S 2023 Applications of data fusion in optical coordinate metrology: a review *J. Adv. Manuf. Technol.* **124** 1341–56

Zhang Z M, Catalucci S, Thompson A, Leach R, Piano S 2023 A new data fusion algorithm for point cloud registration *euspen's 23rd Internat. Conf. & Exhibit., Copenhagen*

List of figures

List of tables

List of parameters

W_i	The weight for calculating the similarity score for each level of voxels.
n_i	The number of matched pairs of voxels in each level of voxels.
s_x, s_y, s_z	The sizes of steps of subdividing the large-sparse point cloud into sub-clouds on *x*-, *y*- and *z*-directions.
N	The number of sub-clouds.
T_1	The number of bins in the histograms of the point-to-PC-plane distances.
$T_1^{'}$	The number of bins that are matched in the histograms of the point-to-PC-plane distances for the two point clouds.
$\{h_{1,i}\}_{i=0,1,2,\ldots,T_1}$	The heights of the bins in the histograms of the point-to-PC-plane distances.
$\{x_{1,i}\}_{i=0,1,2,\ldots,T_1}$	The locations (the midlines on the *x*-axis) of the bins in the histograms of the point-to-PC-plane distances.
$\delta_{large}, \delta_{small}$	The tolerance parameters for comparing the bin height ratios in the histograms of the point-to-PC-plane distances
$\varepsilon_{large}, \varepsilon_{small}$	The tolerance parameters for comparing the bin locations in the histograms of the point-to-PC-plane distances
T_2	The number of bins in the histograms of the local-norm-to-PC-plane angles.
$T_2^{'}$	The number of bins with the largest heights (top bins) in the histograms of the local-norm-to-PC-plane angles.
$\{x_{2,i}\}_{i=0,1,2,\ldots,T_2^{'}}$	The locations of the top bins in the histograms of the local-norm-to-PC-plane angles.
$\theta_{large}, \theta_{small}$	The tolerance parameters for comparing the locations of the top bins in the histograms of the local-norm-to-PC-plane angles.

N	The total number of unique points in the matched sub-clouds (target area in the large-sparse point cloud selected with PCA)
O	The number of orientations of the small-dense point cloud.
o	The index of the small-dense point cloud in a certain orientation.
n_x, n_y, n_z	The numbers of voxels on *x*-, *y*- and *z*-directions.

List of abbreviations

3D	3-dimensional
AI	Artificial intelligence
ANN	Artificial neural network
CAD	Computer aided design
CMM	Coordinate measuring machine
CSI	Coherence scanning interferometry
DCP	Deep Closest Point
FGP	Fused Gaussian process
GP	Gaussian process
HA	High-accuracy
ICP	Iterative closest point
LA	Low-accuracy
NN	Neural network
PC	Principal component
PCA	Principal component analysis

SVD	Singular value decomposition
SVM	Singular vector machine
WLS	Weighted least square

Chapter 1

Introduction

This research is aimed at proposing a novel data fusion pipeline which is a capable of registering point clouds with different sizes and point densities. Data fusion has been implemented in a huge diversity of technologies and industries, ranging from healthcare to defence [1–6]. Having been aware of the functionality and potential of data fusion process, I foresaw that it should be the right moment to introduce data fusion into the research of metrology. In this chapter, the concepts and terms that are necessary to proceed this thesis will be introduced and explained.

1.1. Backgrounds

Data fusion was formally defined for the first time in 1987 by the Joint Directors of Laboratories (JDL) in the USA (United States of America) [7]. In 1991, the JDL defined data fusion as a general technical term as follows [8]:

> *"A process dealing with the association, correlation, and combination of data and information from single and multiple sources to achieve refined position and identity estimates, and complete and timely assessments of situations and threats as well as their significance."*

Data fusion has been employed for research in various subjects, including metrology, a subject on the measurement of the external shape and surface texture of an engineered component [9–11]. The most widely acknowledged definition of data fusion specifically within the context of metrology was given by Weckenmann et al. [12] in 2009, stated as follows:

> *"Multi-sensor data fusion in dimensional metrology can be defined as the process of combining data from several information sources (sensors) into a common representational format in order that the metrological evaluation can benefit from all available sensor information and data. This means that measurement results can be determined, which could not – or only with worse accuracy – be determined solely on the basis of data from an individual source (sensor) only."*

From this definition, the advantages of using data fusion in metrological contexts are clear: fusion of data from a multi-sensor system can be used to acquire information that are not measurable using a single-sensor system [13], which forms the motivation of this research project.

In the scope of this particular research project, "data fusion" refers specifically to the registration of point clouds in the context of optical coordinate metrology, i.e. the registered point clouds are collected from an engineered object using instruments equipped with optical sensors. The term "registration" is defined by Catalucci et al. [14], based on the definition presented in ISO 10360 part 13 [15], as follows:

"Registration is the process that brings multiple point clouds taken from observations of the same scene in their correct, relative position within a shared coordinate system."

The process of fusing multiple point clouds is then defined by Abdelazeem et al. [16], based on the definition presented in ISO 10360 part 13 [15], as follows:

"Data fusion is the process of combining data from multiple sensors in order to obtain better 3D model of an object than that obtained from single sensor data."

An example point cloud that is the result of the fusion of multiple scans is shown in Figure 1. A review of latest research on point cloud registration (see Chapter 2) unveils a popularity of registering multiple point clouds in similar point densities and sizes. This trend provides a void for novel research on registering point clouds with a disparity in point density and size.

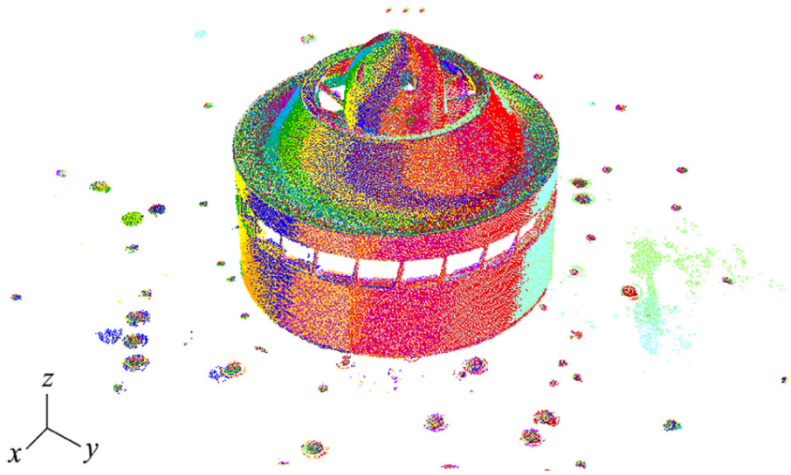


Figure 1 *Multiple point clouds registered in the same coordinate frame. Each individual point cloud is represented with a distinct colour (from* [17]*).*

In the designated application scenario of this research project, the two input point clouds are characterised by the following properties:

1) The large-sparse point cloud is collected within a working volume of $500 \times 500 \times 500$ mm, i.e. none of the object's dimensions should exceed any dimension of this working volume.
2) The large-sparse point cloud and the small-dense point cloud do not share the same coordinate system.
3) The precise location of the area on the surface of the engineered part, where the small-dense point cloud is captured, cannot be determined via any technical device or method because the coordinate frame of the instrument used to capture the small-dense point cloud and the coordinate frame of the instrument used to collect the large-sparse point cloud are not correlated.
4) There is a considerable disparity in point densities in the two point clouds. Specifically, the difference in point density is supposed to be > 1 order of magnitude (i.e. the small-dense point cloud is > 10 times "denser" than the large-sparse point cloud).
5) There is also a considerable difference in the sizes of the datasets (meaning the dimensions of the areas captured during the measurement). It is assumed that the areas covered by the large-sparse point clouds are 10 to 10^2 times the areas covered by the corresponding small-dense point clouds, though the gap in an industrial scenario can be even larger.

1.2. Objectives

The outcome of this research project is an algorithmic pipeline (denoted as *pipeline* in the following texts) which describes the steps for registering two individual point clouds of different characteristics in the same coordinate frame. In this work, the test scenario is identified by one of the two point clouds having a relatively low point density and representing the external shape of an engineered object as a whole (denoted as *large-sparse point cloud* in the following texts); the other point cloud has a relatively high point density and represents the surface texture details of a small area on the external surface of this engineered object (denoted as *small-dense point cloud* in the following texts). Aiming at the goal of proposing this pipeline, this thesis includes the following specific objectives:

1) Review the data fusion algorithms for metrological applications published since 2017[2] and summarise the critical trends and popular techniques among these latest publications.
2) Define the ideal output of this pipeline and dissemble the ultimate goal into individual tasks in relatively simpler tasks, i.e. proposing individual algorithms to be linked as a complete pipeline.
3) Determine the mathematical foundations on which this pipeline should be based.
4) Define the application scenarios of this pipeline in industrial contexts.

[2] The author regards the Year 2017 as the point where machine learning became a prevalent and ubiquitous subject widely employed in industrial and academic scenarios. Moreover, the total number of publications on data fusion since its first-time appearance, which was in the 1960s, exceeds 100,000 according to the results on Google Scholars, which is beyond a human's ability to review such a gargantuan among of materials.

5) Test the pipeline with points of a real engineered part to assess the performance of the pipeline in industrial scenarios.

1.3. Research questions and contributions

To attain the objectives of this research project, this thesis will answer the following questions:

1) Reviewing existing data fusion algorithms applied to optical coordinate measurement tasks, how do they differ in terms of the underlying mathematics, and how do these differences influence their performance?
2) Examining the methodology in testing an algorithm among existing publications: are there any common trends in the geometries of artefacts chosen to test them, and what are the common geometric characteristics of the artefacts?
3) When registering two point clouds collected by separate instruments in irrelevant coordinate frames, how can the computer determine the target area to register a point cloud? Since there is no existing correct answer (i.e. training datasets) available to the computer, how can the computer recognise the pattern of a dataset (the small-dense point cloud) in a dataset (the large-sparse point cloud)?
4) After the small-dense point cloud has been located in the target area in the large-sparse point cloud, how can the computer search for the correct orientation of the small-dense point cloud and hence adjust it to align with the geometrical features in that target area?

5) How does the pipeline perform with point clouds having different levels of geometrical complexity? What are the limitations of this pipeline shown when different geometrical complexities are given?
6) What are the potential application scenarios of this pipeline?

By answering these questions, this thesis will make the following contributions to the science community:

1) A new taxonomy of latest peer algorithms will be proposed, which can assist engineers and researchers to choose and design adequate algorithms for their specific application scenarios (section 2.1 and section 2.2);
2) A novel scenario of point cloud registration will be explored in this research. As mentioned in section 1.1, the smaller point cloud represents the geometries of a subsection in the larger point cloud; the former has a point density 10 to 100 times that of the later; they are located in two individual 3D coordinate systems, not sharing the same coordinate system; due to the inter-independent status of the instrumentation systems used to collect these two point clouds, the location of the observed spot on the object, where the small-dense point cloud is captured, cannot be systematically determined;
3) A new algorithm to recognise a user-defined 3D geometrical pattern, contained in the input dataset, in another 3D dataset, which has a point density 10 to 100 lower than that of the input dataset. This algorithm will not learn the user-defined pattern by receiving a large amount of training data; the only input the algorithm will receive is the point cloud data inputted by the user. Moreover, the functionality of this algorithm will not be diminished by the difference of point

densities between point clouds. As such, a novel technique to store the geometrical information of the input point cloud will be proposed, too (section 3.1 and section 4.2).

4) A novel algorithm to align the orientations of two point clouds regardless of the gap in point densities. This algorithm will assess the point distributions of the two point clouds in the 3D space, resistant to any interference from the difference in point density between the two point clouds (section 3.2 and section 4.3).

1.4. Thesis outline

The rest of this thesis is structured as follows:

In Chapter 2, the latest research since 2017 on the data fusion algorithms in metrological contexts is reviewed and discussed. In Chapter 3, the theoretical foundation of the pipeline is presented and explained, leading to Chapter 4 where the full structure of the pipeline, including the technical parameters, is listed. Chapter 5 presents the experimental studies of the pipeline applied to three point clouds with various geometrical complexities, the outputs of which are assessed and discussed in Chapter 6. The conclusions are drawn in Chapter 7, together with the hints on future works.

The contents of this chapter are partially covered by the following publication by the author of this thesis:

Zhang Z M, Catalucci S, Thompson A, Leach R, Piano S 2023 Applications of data fusion in optical coordinate metrology: a review *J. Adv. Manuf. Technol.* **124** 1341–56

Chapter 2

Literature review

In the previous chapter, there are seven questions listed to be answered in this thesis. In this chapter, I will answer the first two questions by reviewing existing research on data fusion algorithms proposed since 2017, with a particular focus on fusion of two or more datasets featuring 3D coordinate information (i.e. point clouds) of engineered parts measured by optical instruments. As defined in section 1.1, data fusion strictly refers to the registration of two or more 3D point clouds and, therefore, research on data fusion as a broader term, including its possible applications, is out of the scope of this chapter.

In this review, I investigated the latest proposed algorithms by observing their underlying mathematics, as I wanted to propose a method for selecting the most appropriate algorithm for an optical coordinate measurement task [2]. In the reviewing process, I categorised algorithms based on the same mathematical foundation into some category, where algorithms in that category have similar advantages and limitations in measurement applications. With such a taxonomy, researchers can choose the most

suitable algorithm for a given data fusion task. As I focus on the overall mathematics of a data fusion pipeline, individual techniques embedded in an algorithmic pipeline (such as Kalman Filters [18] and its variants, such as recursive filtering [19], particle filtering [20], inertial measurement unit [21] etc.) are not included in this discussion.

This chapter is structured as follows. In section 2.1.1, the taxonomy of existing data fusion algorithms is introduced and explained. In this taxonomy, I classify the existing algorithms into three types: Gaussian process (GP), weighted-least-square (WLS) and machine learning algorithms. In sections 2.1.2, 2.1.3 and 2.1.4, the mathematics behind each type of data fusion algorithms is presented and discussed. In section 2.2, the latest research, including experimental and simulation results, for each algorithm type is presented. In section 2.3, the benefits and limitations of each type of algorithm are discussed, in reference to the information presented in section 2.2. The main discoveries from the literature review are summarised in section 2.4.

The contents of this chapter are included in the following publication by the author of this thesis with updated information:

Zhang Z M, Catalucci S, Thompson A, Leach R, Piano S 2023 Applications of data fusion in optical coordinate metrology: a review *J. Adv. Manuf. Technol.* **124** 1341–56

2.1. Theoretical background

2.1.1. A new taxonomy of data fusion algorithms

Since the late 1980s, researchers have proposed many taxonomies classifying techniques for data fusion. One of the earliest and the most frequently quoted

classifications is Luo and Kay's "data-feature-decision" three-layer classification [22], in which the authors classified data fusion algorithms into three types according to the level of analysis: data level, feature level and decision level. This taxonomy was later elaborated into five fusion classes in Dasarathy's five-layer architecture [23]: in this taxonomy, the type of algorithm is classified according to which of the three levels, proposed in [22], the input and output data belong. A similar taxonomy is the Durrant-Whyte architecture [24], which consists of a data pre-processing level, data-refinement levels and human-computer interactions. These taxonomies do not emphasise the differences in the mathematical basis for each algorithm; instead, the focus is placed on the structure of the data inputted into the fusion system, or the connections between datasets or data fusion steps.

In this chapter, a new methodology for classifying data fusion algorithms is defined based on the mathematical principles that underpin existing data fusion algorithms. The mathematical basis of each algorithm classification is as follows:

- **GP algorithms:** need a GP "governing equation" to describe the calculation process and the geometric features of a surface. GPs are a mathematical tool used to express a stochastic process with Gaussian distribution equations. As such, a mathematical equation, or a set of equations, must be defined before applying the algorithm.
- **WLS algorithms:** designed to reduce the noise when applying data fusion by assigning weights to the measurement result, in the form of a matrix. The working principles of WLS algorithms are similar to those for GP algorithms, and similarly require a governing equation to start the data fusion process.

- **Machine learning algorithms:** able to learn patterns in the input datasets, instead of being defined by mathematical rules in advance.

GP and WLS algorithms require pre-defined mathematical expressions before being implemented, while machine learning algorithms detect patterns in data autonomously. Therefore, the major difference between the first two types of algorithms and machine learning algorithms is that the computing processes used in the first two are user-dependent, while machine learning algorithms do not require manual input to initiate the data analysis process [25]. Due to this distinction, in this review, I propose classifying GP and WLS algorithms as "user-dependent algorithms", while machine learning algorithms, together with other artificial intelligence techniques can be referred to as "user-independent algorithms". Here, I use the definition of machine learning defined by Eastwood et al. to decide whether an algorithm should be classified as a machine learning algorithm. [25]:

"Machine learning can be thought of as a system which is not specifically programmed to solve a problem; it is instead told what problem to solve, given a set of training data, and then learns how best to solve the given problem on its own."

According to this definition, I consider any statistical learning algorithms relying on pre-programmed GP and WLS models as GP and WLS algorithms, instead of machine learning. The proposed taxonomy is shown in Figure 2.

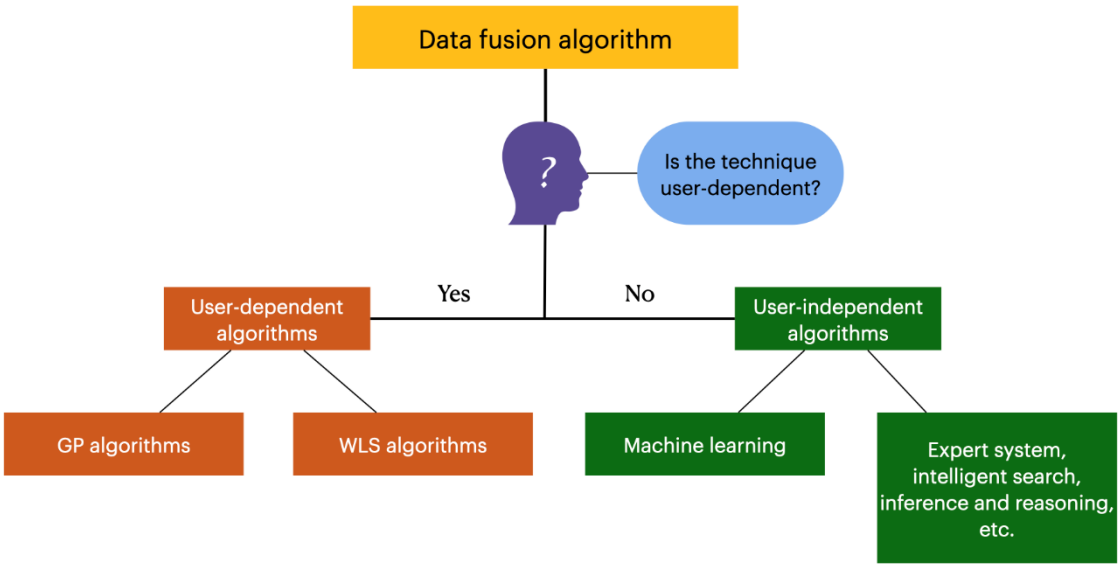


Figure 2 *Taxonomy of data fusion algorithms.*

2.1.2. Gaussian process algorithms

GP algorithms have been widely explored in the context of three-dimensional (3D) point clouds, i.e. sets of x, y, z positions in a 3D coordinate space [26]. GPs are a mathematical tool used to describe normally distributed stochastic processes that evolve in time according to probabilistic laws [27]. Each GP is a collection of random variables, any finite subset of which obeys a joint Gaussian distribution [28]. A GP is defined with the expression

$$GP(\boldsymbol{X}) \sim N(\mu(\boldsymbol{X}), K(\boldsymbol{X}, \boldsymbol{X})), \tag{1}$$

where N represents a normal distribution function and vector $\boldsymbol{X}$ indicates the locations of the data points collected by the sensor, expressed as

$$\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_n], \tag{2}$$

where $\mu(\boldsymbol{X})$ is the mean function. $K(\boldsymbol{X}, \boldsymbol{X})$ is the covariance matrix, defined as

$$K(\boldsymbol{X}, \boldsymbol{X}) = \begin{bmatrix} k(\boldsymbol{x}_1, \boldsymbol{x}_1) & k(\boldsymbol{x}_1, \boldsymbol{x}_2) & k(\boldsymbol{x}_1, \boldsymbol{x}_3) & \cdots & k(\boldsymbol{x}_1, \boldsymbol{x}_n) \\ k(\boldsymbol{x}_2, \boldsymbol{x}_1) & k(\boldsymbol{x}_2, \boldsymbol{x}_2) & k(\boldsymbol{x}_2, \boldsymbol{x}_3) & \cdots & k(\boldsymbol{x}_2, \boldsymbol{x}_n) \\ k(\boldsymbol{x}_3, \boldsymbol{x}_1) & k(\boldsymbol{x}_3, \boldsymbol{x}_2) & k(\boldsymbol{x}_3, \boldsymbol{x}_3) & \cdots & k(\boldsymbol{x}_3, \boldsymbol{x}_n) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ k(\boldsymbol{x}_n, \boldsymbol{x}_1) & k(\boldsymbol{x}_n, \boldsymbol{x}_2) & k(\boldsymbol{x}_1, \boldsymbol{x}_3) & \cdots & k(\boldsymbol{x}_n, \boldsymbol{x}_n) \end{bmatrix} \tag{3}$$

where $k(\boldsymbol{x}_i, \boldsymbol{x}_j)$ is the covariance kernel function [28].

2.1.3. Weighted least-squares algorithms

Algorithms using weighted least-squares (WLS) methods were introduced by Forbes [29], specifically in the context of coordinate metrology. Forbes's work aimed at reducing the noise in data by applying weights to each dataset. WLS fusion is based on a linear measuring system [29], given by

$$\boldsymbol{z} = H\boldsymbol{x} + \varepsilon, \tag{4}$$

where $\boldsymbol{x}$ is an n-vector comprised of the model parameters to be measured, H is an $m \times n$ ($m > n$) matrix of the measured points, $\boldsymbol{z}$ is an m-vector representing the measurement result and ε is a noise vector independent from the collected data, given by $\varepsilon \sim (\boldsymbol{0}, \sigma^2 \boldsymbol{I})$ [29]. Assume there is a sample set with K samples, given by $\{z_k\}_{k \in K}$ with noise level ε, the model parameter vector $\boldsymbol{x}$ can be solved by minimising the weighted squares cost function

$$\sum_{k \in K} w_k \| z_k - H_k \boldsymbol{x} \|^2, \quad (5)$$

where w_k are designed scalar weights [30]. The fusion process is based on solving the model parameter vector $\boldsymbol{x}$ by forcing the partial differential equation of this weighted squares cost function to be zero. Existing research has so far focused on proposing different methods forcing this result. Additionally, researchers have been exploring new methods of designing weights w_k (see section 2.2.2).

2.1.4. Machine learning algorithms

Machine learning has been an active research area in academia since late 1950s, but has more recently become an industrial focus [31–33], because available computational resources have significantly increased in the past two decades [34]. As such AI, and particularly machine learning, has become an efficient tool for data-intensive research [35], especially in the context of optical coordinate measurement.

According to the definition given by Eastwood et al. [25] (section 2.1.1), the computer can be used to predict trends in new data by learning patterns in existing data (training data) with a pre-programmed logic. The central idea surrounding machine learning algorithms is to create an autonomous data processing system, unlike GP or WLS methods, where a manually-defined mathematical description is required [36]. As such, machine learning algorithms are frequently used to solve problems that are difficult to model with predefined mathematical expressions [36,37].

There are three categories of machine learning, with each category depending on how the original data is pre-processed: supervised learning, unsupervised learning and semi-supervised learning [38]. These categories are as follows:

- **Supervised learning:** If the input datasets have been manually labelled, then supervised learning algorithms are used to learn such datasets. Two types of supervised learning are most frequently used: support vector machines (SVMs) and neural networks (NNs). SVMs are used to realise binary classification. NNs, also known as artificial neural networks (ANNs), learn certain parameters of a dataset by analysing the data with multiple layers of neurons (nodes), each of which have various statistical weights defined by the user [39].
- **Unsupervised learning:** In the case of unsupervised learning, the input datasets are unlabelled, meaning that the algorithms learn the patterns in the data without direct human input. Unsupervised learning algorithms recognise the hidden patterns in a given dataset by clustering data points. It should be particularly noted that NNs and their variants can also be applied in an unsupervised learning case, particularly when used to detect or extract patterns in data [39–42]. A typical example of NNs used as an unsupervised learning technique is discussed in section 2.2.3.
- **Semi-supervised learning**: Semi-supervised learning algorithms process partially-labelled datasets [39]. To our knowledge, in the context of optical coordinate measurement, no work has been published using semi-supervised learning for fusing datasets with incomplete labels. As such, discussion of semi-supervised learning research is not included in this review.

In addition to machine learning, there are other fields under the broader area of AI, some of which are shown in figure 3, but discussion of these areas is beyond the scope of the review.

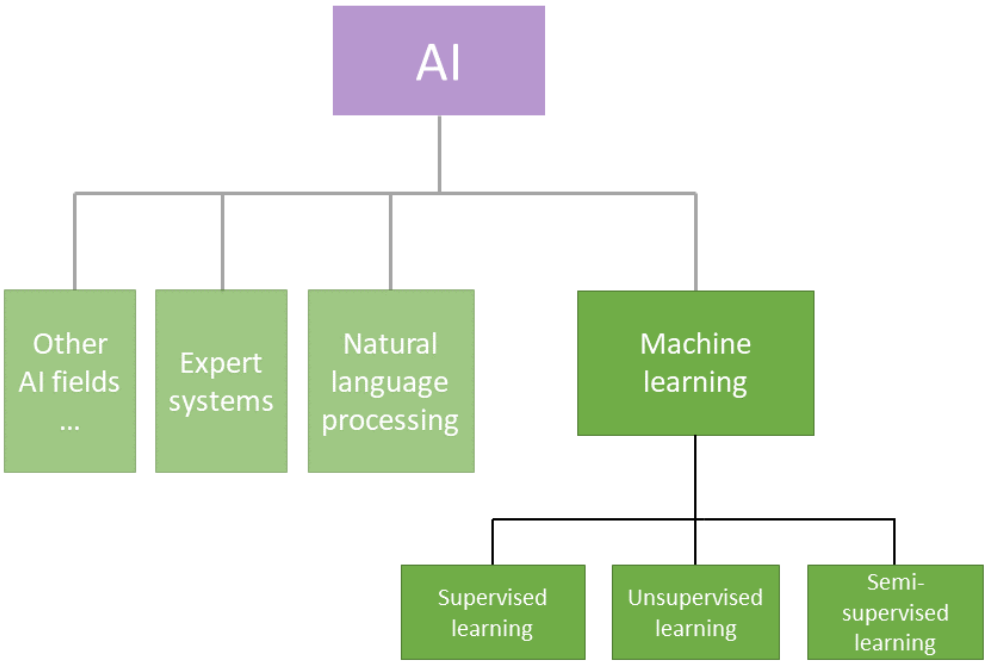


Figure 3 *Machine learning is a type of AI, together with other subjects such as expert systems and natural language processing. There are three categories of machine learning: supervised learning, unsupervised learning, and semi-supervised learning.*

2.2. Current state of the art

In the following sections, I present recent research on the application of each algorithm type introduced in section 2.1. The advantages and limitations of each algorithm, together with the general problems presented in existing research, are examined.

2.2.1. GP algorithms

Research on GP algorithms shows a broad range of applications in optical coordinate measurement, separated into two branches: those involving fusion of datasets collected

by multiple sensors (this is the most common branch), and those involving enhancement of the measurement processes to improve the quality of datasets.

Ji et al. [43] introduced the "adjustment model" to supplement GP methods. The adjustment model is designed to fuse inhomogeneous data: a low-accuracy dataset (here, a geometric dataset collected by an optical coordinate measurement system) with a high-accuracy dataset (here, a geometric dataset collected by contact coordinate measurement system). Before implementing the adjustment model, the external geometry of the measured surface is firstly predicted by applying GP to the low-accuracy dataset, forming a model of the surface coordinates. Then the adjustment model is used to describe the difference between the model of the surface coordinates and the high-accuracy dataset. In this process, the high-accuracy dataset acts as the basis for correction.

In their paper, Ji et al. [43] chose two artefacts: an array of spherical holes (see figure 4) and a machined freeform surface. For each of these artefacts, a low-accuracy (LA) dataset and a high-accuracy (HA) dataset were collected by a contact and an optical coordinate measurement system. The authors attempted to fuse the two datasets of each of these two artefacts employing their adjustment model. Results showed that the GP method with the adjustment model could fuse the inhomogeneous measurements of the complex surfaces. Additionally, the fusion process only required a small portion of the HA dataset and one LA dataset, which improved the efficiency of the measurement and fusion processes.

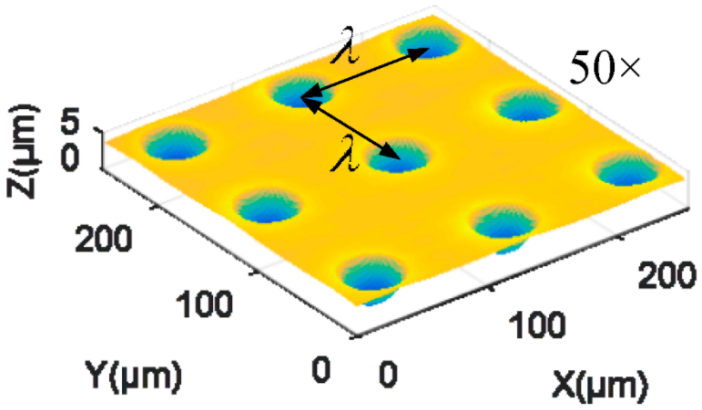


Figure 4 *Artefact comprising an array of spherical holes used in* [43].

In addition to the work presented by Ji et al., numerous GP algorithms have been proposed over the past five years. Ma et al. [44] developed a new method called "fused Gaussian process" (FGP) with a two-component covariance structure. Their algorithm was designed to fuse large spatial datasets (e.g. remote sensing data). Yin et al. [45] introduced a similar GP regression algorithm specifically designed for multi-sensor systems for the measurement of complex surfaces. Experimental tests indicated that the developed algorithm was able to perform intelligent sampling (i.e. autonomous selection of effective data points for analysis) when fusing datasets from various sensors.

2.2.2. WLS algorithms

Many algorithms based on WLS theory have been proposed in the past few years, aiming at improving the accuracy and efficiency of data fusion. The most common WLS algorithms are those developed by Forbes [29] and Ren et al. [46] (see figure 5 for the experimental settings in the research by Ren et al.). The former introduced a general Bayesian approach in order to balance the noise parameters; however, this solution relies on the fitting accuracy of the linear surface model, and its application in multi-sensor

fusion is limited [46]. To overcome this issue, the latter added a surface registration method into the general WLS algorithm. The results of both simulations and experiments indicated that their new method could improve the fidelity of the reconstructed surface, modelled using their experimental data and the algorithm.

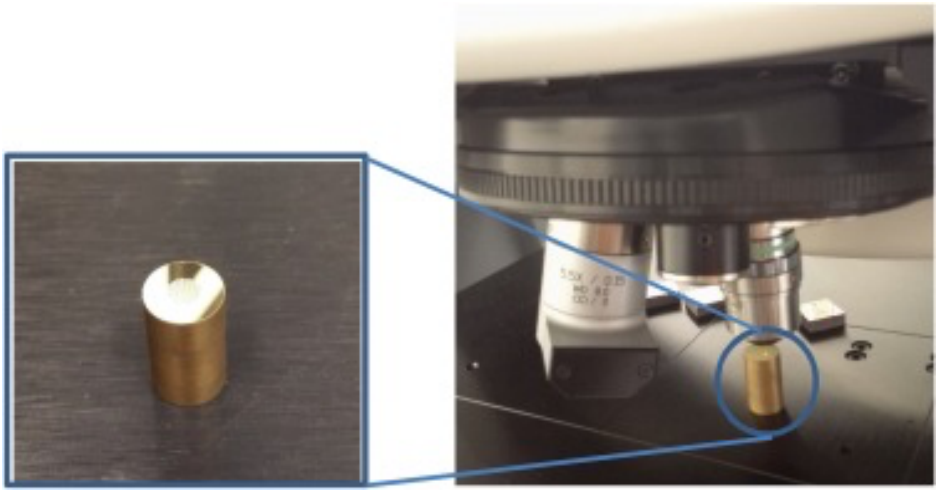


Figure 5 *Measurement of a sinusoidal structured surface using an optical surface topography measurement system. The data collected by the measurement system is used to test the WLS algorithm proposed by Ren et al. [46].*

Research on coordinate measurement has also highlighted the limitations of WLS algorithms in non-linear surface model and multi-sensor data fusion. Yu et al. [47] commented that, although the WLS method is capable of multi-sensor data fusion, showing noticeable reduction of the measurement uncertainty; these algorithms are still unable to provide comparable accuracy to GP algorithms, unless a large number of contact points have been measured, which limits the efficiency of the algorithms. Xiang et al. [48] pointed out that the performance of WLS algorithms is comparable to that of GP algorithms only when processing homogenous datasets, and they are not suitable to fuse datasets collected from large-scale surfaces (e.g. surfaces of major parts on the body of an aircraft or rocket [49]). Kong et al. [50] showed that WLS algorithms depend on

linear approximations of the geometry, which means they are not ideal for measuring objects of highly complex geometry (e.g. objects with sharp geometrical changes or smooth surfaces with micro-structures embedded).

2.2.3. Machine learning algorithms

A general review on machine learning in data fusion was recently presented by Meng et al. [39]. In their review, the authors noted that most of the existing research is focused on stochastic or time-series data analyses, covering topics such as autonomous vehicles, the Internet of Things and geographic information systems. In metrology, the most popular application is the fusion of 3D point clouds (the relevant definitions were given in section 1.1).

To apply machine learning to point cloud registration and fusion, Wang et al. [51] created a registration and fusion pipeline named “deep closest point” (DCP). This method is proposed as an alternative to the iterative closest point (ICP) algorithm, which is one of the most popular solutions for point cloud registration [51–53]. ICP is an iterative process that is employed to minimise the distance between two point clouds. A point cloud (identified as the reference or target) is kept fixed, while the second one (identified as the source or moving set) is transformed to best match the reference based on rigid motion [17,51]. The ICP algorithm can fail to reach the global minimum due to its non-convexity, i.e. a non-convex function may have multiple local minima in a certain range; a local minimum found in a certain range does not necessarily correspond to the global minimum [54]. The methods presented in [55–57], which are all variants of ICP supplemented with various statistical optimisation methods, were developed to address this issue. However, in some cases these methods have not been proven effective.

The DCP model aims to provide a solution to the local minima problem with ICP, and consists of three steps [51]:

1) Embed two individual point clouds into a common space and find the corresponding points in two clouds.
2) Create an attention module combined with a pointer generation layer to provide an initial matching, i.e. a probabilistic map from one point cloud to the other.
3) Extract the accurate alignment by analysing the results from step 2) with a differential singular value decomposition (SVD) layer.

Here, the attention module is a machine learning mechanism that highlights the key data points to increase the accuracy of prediction [58]. A pointer is the core element of a deep-learning technique called “pointer networks”, which uses attention as pointers to select input data for combinatorial matching [59]. In this research, pointer generation is a step used to expose matched pairs of points in two point clouds and create an initial matching [51]. The matrix of this matching is used to extract translation and rotation matrices for accurate alignment with SVD technique [51].

Wang et al. [51] compared the outcomes of DCP and ICP algorithms for registration of multiple point clouds. The results of this comparison are shown in figure 6. The comparison shows that ICP assisted by DCP can converge to the global optimum, and DCP can increase the accuracy of alignment when registering two point clouds with poor initial alignment [51]. Additionally, Wang et al. drew the conclusion that DCP is a capable algorithm for rigid registration tasks and can be used to replace ICP algorithms, considering its reduced registration errors.

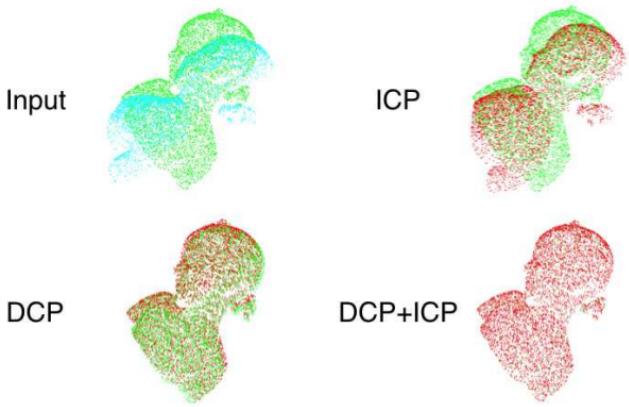


Figure 6 *The fusion of two point clouds by DCP and ICP* [51].

Later research by Gojcic et al. [60] proposed an NN model, a popular technique in machine learning (see section 2.1.4), to fuse two multi-view (i.e. *measurement of spatial coordinates through registration and fusion of multiple single-view measurements in different locations and orientations of the optical sensor relative to the workpiece,* as defined in ISO 10360 part 13 [15]) datasets into a single point cloud. The algorithm models the registration process using an end-to-end NN, whose accuracy is estimated by a specifically designed layer within the NN. In [60], Gojcic et al. defined the problem of aligning two point clouds as an iterative WLS problem, i.e. the 3D transformation matrices which adjust the orientations of two point clouds by iteratively refining by the NN with WLS method (see section 2.1.3 for WLS). To demonstrate the advantages of their NN algorithm, Gojcic et al. tested it with three common benchmark datasets: 3Dmatch [61], Redwood [62] and ScanNet [63]. Compared to non-machine learning algorithms, the NN model was superior in terms of run time, rotational error and translational error [60].

In addition to the research discussed above, Zhang et al. [64] gave a general introduction to machine learning algorithms for data fusion in optical coordinate measurement. In this review, Zhang et al. categorised all machine learning algorithms within the context of optical coordinate measurement into two types: machine learning as a step added to a traditional measurement pipeline and a complete substitute of traditional pipelines using machine learning technologies [31].

2.3. Discussion on existing research

2.3.1. Discussion on the state of the art

In this section, I will discuss the state of the art in user-dependent (i.e. GP and WLS algorithms) and user-independent (i.e. machine learning) algorithms.

In the studies on GP algorithms introduced in section 2.2.1, researchers usually tested an algorithm by fusing a 'high-accuracy' (HA) dataset with a 'low-accuracy' (LA) dataset. The HA data were collected by micro-scale pointwise measuring sensors, such as coordinate measuring machines (CMMs) in the study by Yin et al. [45]. The LA data were collected using optical inspection sensors such as fringe projection systems [43]. The data collection efficiency and the point density of an HA dataset are low due to the functionality of most pointwise measuring instruments; on the contrary, data collection for an LA dataset is rapid and its point density is high, but it cannot provide the coordinate information as accurate as an HA data [43,45]. The benefit of registering an HA dataset with an LA dataset, therefore, is to combine the accuracy of the former and the efficiency of the latter.

Existing studies in GP algorithms usually chose one of the following two paths to fuse an HA with an LA dataset. The first path is to register two datasets by optimising the point-to-point distance in both datasets. In the GP algorithm research discussed in this review, the most typical method used in this path is presented in Ji et al. [43]. This method fuses two datasets by unifying the coordinate systems through optimisation of the point distances. The other path is to propose new sampling methods based on GP for data fusion. In this path, the LA dataset is first subsampled, then GP is applied to the subsampled dataset to reconstruct the surface geometry. Finally, the HA dataset is registered with the reconstructed surface. The key to this research path, therefore, is the proposition of new data sampling methods. The works by Yin et al. [45] and Chen et al. [65] are typical examples of this path: they both proposed intelligent adaptive sampling methods (i.e. sampling size and positions vary according to the local changes in surface geometry [66]) based on GP inference. Chen et al. particularly discussed scenarios where the measured surface has sharp geometrical variations.

Like the existing studies on GP algorithms, typical research on WLS algorithms such as [46] also fused an HA dataset with an LA dataset. However, as mentioned in section 2.2.2, WLS is not generally useful for fusing complex geometric data because of its reliance on linear modelling [43].

Researchers in machine learning tend to test their algorithmic pipelines by fusing two or more datasets representing the same 3D object. For example, Wang et al. [51] tested their DCP pipeline by fusing multiple pairs of point clouds, with each pair of point clouds represented a 3D object with approximately the same point densities. The test datasets in [60], as another example, were multiple point clouds representing a building

interior. Unlike the datasets used for testing GP and WLS algorithms, these point clouds did not contain surface texture information, instead only representing the general shapes of the object or building interior. As such, existing research has not clarified whether machine learning is effective for registering point clouds containing tiny and dense surface textures to large, sparse point clouds. Moreover, as the benchmark datasets usually had similar densities, the performance of machine learning in registering dissimilar point clouds is also unknown. Fusion of dissimilar point clouds (e.g. small-and-dense point clouds to large-and-sparse point clouds) represents an area of future research.

2.3.2. Advantages and limitations of user-dependent and user-independent algorithms

As the most popular type used in optical coordinate metrology, GP algorithms are relatively simple to implement and can be used for flexible nonparametric inference (i.e. inferring the unknown quantities in the data while making as few assumptions as possible) [67–69]. GP-based methods have these advantages because GP is the mathematical basis of many statistical learning algorithms [69]. However, most of the research on GP methods is limited to tests and applications performed on objects with simple geometries. Consequently, whether these algorithms are capable of dealing with multiscale complex surfaces is not yet clear [26,45,70–72]. Additionally, the implementation of GP algorithms simplifies the real modelling process into a set of GP equations [67]. In industry, however, the measurement and data fusion processes are more complex than a set of equations can describe and predict (e.g. due to the continuous change of environmental conditions) [73,74].

As user-dependent algorithms, both GP and WLS algorithms rely on user-defined mathematical expressions to process the external geometry and surface topography data. Essentially, when applying these algorithms, it is assumed that the engineered part's geometry can be described with a set of GP or WLS equations. However, the measured part can have highly complex geometric features; more complex than those that GP and WLS algorithms are able to model. This limitation is particularly notable in the implementation of WLS algorithms, as presented in section 2.2.2 by [47,48,50]. The surface geometries can be far more complex than the mathematical equations can model, which is the general limitation for user-dependent algorithms, as [73,74] indicate.

As user-independent algorithms, machine learning algorithms define a model that allows learning of the patterns in the data autonomously after being trained with the training datasets, instead of using pre-defined mathematical expressions. Consequently, machine learning solutions potentially provide more flexibility than GP and WLS algorithms, particularly when measurement tasks cannot be modelled with specific equations.

Recent research has shown the potential of machine learning algorithms. The Elman ANN algorithm has been shown to be capable of nonlinear predictions in practical applications, for instance in the determination of the position of an object in 3D space [75]. Fahmy et al. [76] indicate that machine learning-enhanced techniques can overcome data imperfection better than GP and WLS algorithms. Tong et al. [77] demonstrated that machine learning showed robustness against noise compared to GP algorithms. Shu et al. [78] demonstrated that a machine learning model was robust in fusing datasets that had ambiguity and noise. A later work by Wang et al. [79]

demonstrated the efficiency and stability of machine learning algorithms with an object-tracking task. Alyannezhadi et al. [80] proposed a clustering algorithm to fuse datasets whose characteristics could not be identified by mathematical equations.

While the machine learning methods discussed were not all applied within the context of metrology, researchers have demonstrated that machine learning is capable of complex tasks and is robust against noise and data imperfections, which are important advantages for applications in metrology. One of the directions of future research is to explore machine learning models that can fuse dense point clouds. As Wang et al. [51] discuss, the latest machine learning models are only successful with point clouds of 500 to 5 000 points; ideal machine learning models should be able to process up to 300 000 points.

Regarding the precision of these algorithms and evaluation of their contributions to measurement uncertainty, the published research has not included much discussion, instead only comparing new machine learning algorithm performance with other earlier algorithms, such as ICP and fast global registration (FGR) [81]. In optical coordinate metrology, comparison with non-machine learning algorithms using the same benchmark datasets is also rarely seen in the literature. Because of these issues, it is difficult to comment on methods for evaluation of uncertainty and precision of existing machine learning algorithms, particularly when compared with non-machine learning algorithms.

To decrease errors, existing research on new machine learning methods commonly proposes iterative refinement, or “Multiple Run” [51,60,82]. These experimental outcomes indicate that multiple runs could improve accuracy after initial registration. To

our knowledge, researchers have not proposed more techniques particularly aimed at improving uncertainty and precision of machine learning. As such, examination of methods for evaluating the contribution to measurement uncertainty from these algorithms is a ripe area for future research.

Another constraint inherent to machine learning is its dependence on existing data (training data) for predicting trends or identifying patterns within datasets, as indicated by the definition of machine learning [25]. As of the completion of this thesis, to the latest knowledge of the author, machine learning algorithms for point cloud registration within metrological contexts without reliance on training data have not been identified.

Table 1 *Summary of the characteristics of all algorithms*

Types	Method	Advantages	Limitations	Literature sources
User-dependent	GP	Simple to implement and can be used for flexible nonparametric inference [70,71,83]	Replying on user-defined mathematical models [83]; ineffective for fusing data of complex geometries[72–74,84].	Ji et al. [43] Ma et al. [44] Yin et al. [45] Chen et al. [65] Wang et al. [84] Yang et al. [85]
	WLS	Simple to implement [29].	Low efficiency [47]; incapable of large-scale data fusion [48]; not ideal for fusing data of highly complex surface [50].	Forbes [29] Ren et al. [46]
User-independent	Machine learning	Does not require user-defined mathematical models, as it learns the patterns in the input data autonomously [86]; able to fuse data that cannot	Applications particularly in external coordinate measurement are rare; it has not been tested with highly	Wang et al. [51] Gojcic et al. [60] Kolanowski et al. [75] Fahmy et al. [76] Tong et al. [77]

		be described with user-defined mathematical models [80]; robust against noise in data fusion [78]; high effectiveness [64].	dense point clouds [51].	Shu et al. [78] Wang et al. [79] Alyannezhadi et al. [80]

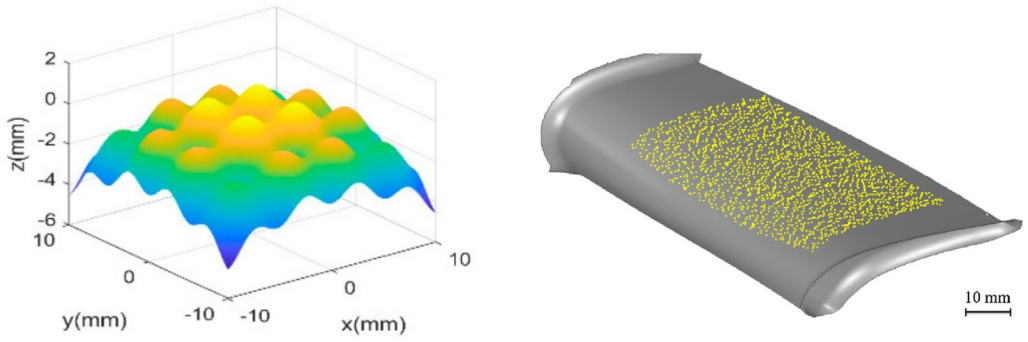
The advantages and limitations of all types of algorithms are summarised in Table 1, using the taxonomy I proposed in section 2.1.1. In this summary, user-dependent algorithms rely on user-defined mathematical models to learn the geometrical patterns in the input data and then fuse the input data based on the user-defined models. This feature makes user-dependent algorithms easy to deploy but also difficult to fuse data of complex geometries, because the surface geometries can be more complex than those that GP and WLS are able to describe, as [73,74] indicate.

User-independent algorithms, such as machine learning, can recognise geometrical patterns without user-defined mathematical models. These algorithms use techniques, such as NNs, to detect the patterns autonomously. This feature makes user-independent algorithms effective in fusing data of complex geometries that are difficult to analytically model. However, as existing research is still relatively rare, more experimental studies are in need to prove the effectiveness and reliability of machine learning used in optical coordinate measurement, as stated in section 2.3.1.

2.3.3. Problems with experimental studies in existing research

The geometries of the test artefact have rarely been discussed in existing reviews. In the literature collection process, I have noted that there are certain trends in the selections and designs of the test artefacts in existing research. In the research presented

in sections 2.2.1. 2.2.2 and 2.2.3, the artefacts or virtual surfaces that were used to test the algorithms usually had simple geometries. These artefacts and virtual surfaces generally exhibited the following characteristics: the virtual surfaces were defined using periodic mathematical patterns, i.e. the definitions of the surface used in each paper are usually in form of $z = \sin(Ax) + \cos(By)$, $A, B \in \mathbb{R}$. Similarly, artefacts used were either periodically patterned or had simple curvature. For example, artefacts with highly symmetrical geometries were frequently used for testing the newly proposed algorithms (see examples in Figure 4, Figure 5 and Figure 7).



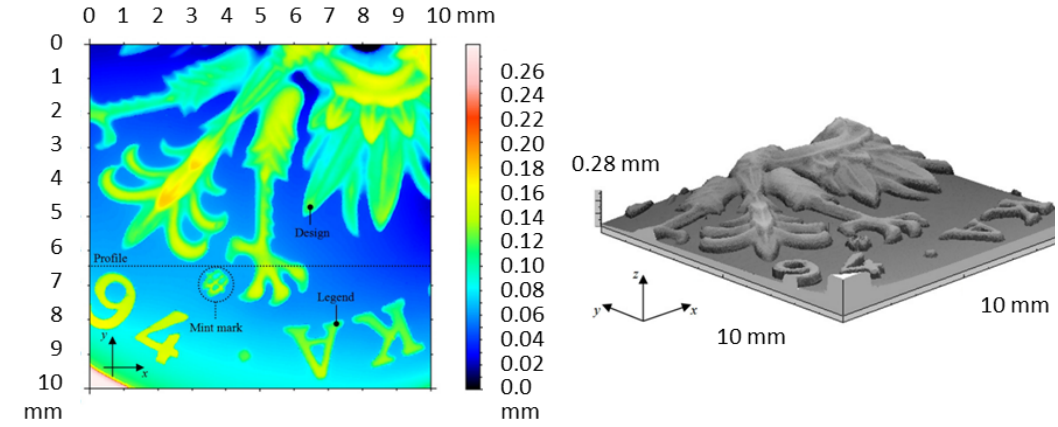
(a) (b)

Figure 7 *Test examples of recently developed algorithms: (a) shows a simulated surface generated by a mathematical equation in form of* $z = sin(Ax) + cos(By)$, $A, B \in \mathbb{R}$ *(from* [28]*), and (b) is an artefact with relatively smooth geometry, with the yellow dots representing the data points collected by the measurement instrument (from* [30]*).*

As discussed in section 2.3.1, the geometries of engineered parts are likely to be more complex than the geometries of simulated surfaces and custom-designed artefacts in laboratories. Therefore, in addition to simulated surfaces and artefacts, common objects and engineered parts used in industry should be employed in

industrial and research activities to test different measurement techniques. For instance, coins represent inexpensive examples of metal freeform surfaces that can be used for detection of defects in surface topography measurement [87].

Common objects used in practical circumstances, such as a coin, can challenge an algorithm more than a simulated surface or a specifically manufactured artefact. Taking a coin as an example, the patterns on it, e.g. motto, legend, mint mark, display many convex and concave geometries, which are characterised by a wide dimensional range (see figure 8). As such, a coin can provide various opportunities to challenge the capability of a coordinate measurement technique, including data fusion algorithms used in this process [88]. Additionally, coins with different materials and worn surfaces lead to different optical reflectivity conditions, which can influence the data collected by optical sensors [89]. Therefore, a coin is also an effective object to test the robustness and stability of a data fusion algorithm in coordinate measurement processes.



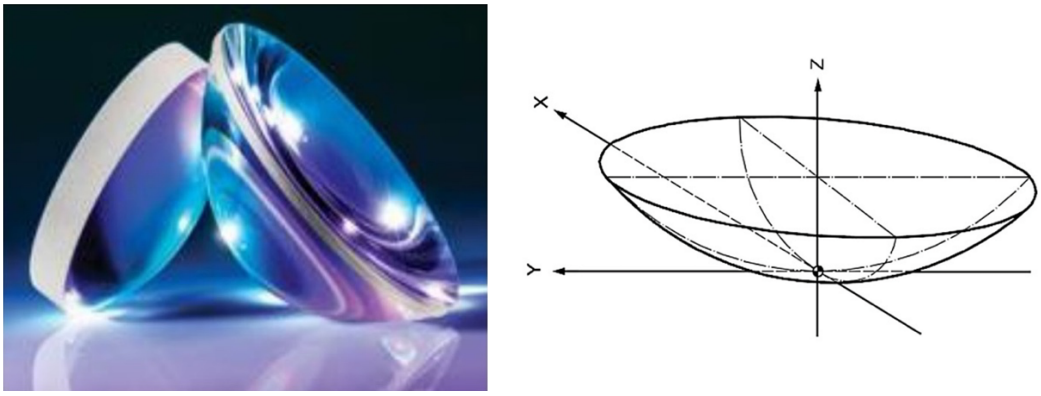
(a) (b)

Figure 8 *An example of surface profile of a part of a coin (1 zł, Poland) adapted from* [88]*, showing complex geometric features: (a) the colour map of height, in which the authors mark geometric features including design, legend and mint mark. The dot line marked with "Profile" is where the authors of* [88] *used for further research in their work on surface proliferation; (b) 3D model of the same area on the coin, rendered with data shown in (a).*

In terms of geometric complexity, a simulated surface may also display highly complex geometry if its mathematical expressions have numerous items, as Eastwood et al. [90] and Todhunter et al. [91] indicated in their work. If a 3D model is constructed with such polynomials and inputted into an additive manufacturing system, a highly complex artefact can be manufactured for testing data fusion algorithms. However, the more complex the surface, the higher the computational cost.

Another popular type of object used for metrological research is aspherical lenses [92,93] (see figure 9). Aspherical lenses have become increasingly common in industry because of the progress of manufacturing technology and, therefore, they are frequently used in medical imaging, optical systems, astrophysics, lithography, automotive and metrology [94]. The broad range of applications in high-precision optical fields demands better techniques for coordinate measurement in

manufacturing, designing and testing aspherical lenses [95,96]. As such, coordinate measurement of an aspherical lens challenges the accuracy of data fusion algorithms used in this process [97].



(a) (b)

Figure 9 *(a) A photo of two example aspherical lenses, (b) schema of an asphere* [92,98].

2.4. Summary

This chapter has reviewed a broad array of data fusion algorithms used for optical coordinate measurement proposed since 2017 and defined a new taxonomy for the classification of existing algorithms based on their mathematical backgrounds: user-dependent algorithms (GP methods and WLS methods) and user-independent algorithms (machine learning algorithms). The critical points are:

1) User-dependent algorithms are relatively easy to implement, as these algorithms assume that the geometry of a surface can be described by a series of mathematical equations. However, these algorithms are not capable of modelling highly complex surfaces. Recent research on user-independent

algorithms represented by machine learning and its derivatives such as deep learning is relatively scarce, but the results have already demonstrated its potential and value for further exploration.

2) Virtual surface plus an engineered object is generally used to test the algorithms. The problem with virtual surfaces is that they are usually created using simple mathematical expressions, consisting of a sine and a cosine term. The problem with engineered artefacts is that there is commonly a lack of complex geometric features on their surfaces, e.g. sharp changes in height and irregular scratches.
3) User-independent algorithms can recognise geometrical patterns without preprogrammed mathematical models, detecting geometrical patterns in point clouds autonomously. However, the autonomous pattern recognition of user-independent algorithms is based on training datasets, which are existing correct answers as inputs to the algorithms. Moreover, existing research and experimental studies on user-independent algorithms in metrological contexts are still relatively rare in recent research.

These findings indicate the following knowledge gaps that this research ought to fill:

1) The new algorithmic pipeline should be capable of processing point clouds with complex, irregular geometries, which cannot be described with analytical mathematics such as Gaussian process and weighted least square. This contribution will be critical for real-world industrial contexts, because mathematically modelling real-world surface textures, such as irregular scratches caused in usage and flaws caused by manufacturing processes, can be

a perplex and daunting task, which makes the point cloud registration process less efficient.

2) The existing studied introduced in sections 2.2 and 2.3 are focused on registering point clouds with similar point densities and spatial coverages (i.e. sizes). To the author's latest knowledge, there has not been a study particularly focusing on registering point clouds with a considerable level of point density difference, e.g. the point density of one point cloud is more than 10 times higher than that of the other one, and size difference, i.e. the area represented by one point cloud is merely a small part of the object represented by the other one;
3) The geometries of real-world surface textures are infinitely diverse, but the scenarios contained in the training dataset for a machine learning algorithm, no matter how massive and inclusive it is, are always limited by the imagination of the researchers who train the machine learning model; when there is a new task showing a geometrical feature outside the scope of the training dataset, the machine learning algorithm will be stumped by the geometrical novelty. As such, the new pipeline should be established on a new logic, by which the pipeline can recognise any geometrical pattern in a point cloud without depending on massive training data.

The literature review has provided nourishment to designing a new data fusion procedure for optical coordinate measurement, but more importantly it unveils the uniqueness of the scenario in this thesis. As such, in the next chapter, a data fusion

pipeline which functions specifically for the scenario in this research project will be proposed. Its mathematical principles will be particularly explained.

Chapter 3
A statistical-based pipeline

In Chapter 1, section 1.2, the characteristics of the point cloud registration pipeline proposed in this research are stated. The strategy is to register two point clouds which have considerably different sizes and point densities and do not share the same coordinate frame. Based on our knowledge, there is no existing database with correct answers as input of the pipeline as training data. The literature review in Chapter 2 indicates that, despite the diversity of mathematical principles and application scenarios, the latest presented algorithms cannot be suitably employed for the registration task investigated in this research project: they either fail in registering point clouds with a disparity in size and point density, or not applicable to analyse a given pattern in point clouds if not trained with pre-existing training data. As such, a new method, which can overcome these challenges, has to be proposed.

According to the characteristics of the designated point cloud registration, the steps in this pipeline can be concisely planned as:

1) find the target area in the large-sparse point cloud where the small-dense point cloud should be located;

2) move the small-dense point cloud into the target area in the frame of the large-sparse point cloud;

3) detect the correct orientation of the small-dense point cloud;

4) change the pose of the small-dense point cloud to the correct one.

The pivotal points are steps 2) and 3): how to find the target area and the correct orientation to register the small-dense point cloud, without training data available? The solution provided in this thesis is to compare the geometrical similarities between two point clouds: the area in the large-sparse point cloud which displays the most similar geometrical characteristics in statistical terms is determined as the target area to locate the small-dense point cloud; the orientation of the small-dense point cloud which gives the most similar spatial point distribution to the target area is determined as the correct orientation. The question is: how to measure the geometrical similarity between two point clouds in 3D space?

The solutions provided by this research are principal component analysis (PCA) and cross-sectional geometrical comparison. The area in the large-sparse point cloud which shows the most similar PCA results to the small-dense point cloud is determined as the target area to locate the latter. The orientation in which the cross-sectional geometry of the small-dense point cloud is most similar to that of the target area is determined as the correct orientation. The mathematical foundations behind the two techniques will be discussed in sections 3.1 and 3.2.

3.1. Principal component analysis (PCA)

PCA is widely used in data analysis for complexity reduction. In point clouds, the principal components (PCs) correspond to the axes along which the data points are distributed [99]. It should be noted that there are techniques for data dimension reduction other than PCA, which are widely implemented in complex data analysis tasks. The typical examples, which can be potentially applied to the scenarios in this research, are t-distribution stochastic neighbour embedding (t-SNE), linear discriminant analysis (LDA) and independent component analysis (ICA). However, due to the congenital characteristics of these techniques and the idiosyncrasies of the context of this research, PCA is still the winner over its potential alternatives.

3.1.1. Why PCA?

The potential competitors against PCA have their limitations for the scenario in this research, which are exposited as follows:

1) **T-SNE**: a non-linear dimensionality reduction technique which is particularly effective for complex data visualisation. It is not a suitable candidate for this research because it is a non-linear, stochastic method, which gives a different output in each new run. More importantly, it is focused on preserving local geometrical structures and hence potentially distorts global geometrical relationships crucial for detecting geometrical similarities in point clouds. Additionally, it is computationally intensive for massive spatial datasets in the scale of the ones in this research [100].

2) **LDA**: a linear technique which works by reducing data dimensions by finding feature combinations that best separate classes within labelled data. It can only be implemented for classification tasks, which require labelled input datasets [101].
3) **ICA**: a technique works by separating a multivariate signal into additive, independent components. Its most popular application is signal processing. ICA assumes statistical independence of components, which does not work effectively with the 3D spatial correlations in point clouds [102].
4) **Deep Learning**: Deep Learning programmes utilise neural networks to learn representations from data. Its pattern recognition ability is based on training data, which is unfeasible in this research [103].

In addition to the idiosyncrasies of the scenario in this research, PCA also exhibits the following advantages for this task compared to aforementioned options:

1) PCA is straightforward to implement and computationally efficient, making it suitable for massive 3D point clouds with different point densities.
2) It does not require labelled data, aligning well with the unsupervised nature of the detection of geometrical similarities.
3) It effectively figures out the principal axes (principal components) of variance, capturing the global geometrical structure in the point cloud that is necessary for geometrical similarity detection.
4) By focusing on the overall variance, PCA can handle considerable differences in point density, facilitating the geometrical similarity comparison between large-sparse and small-dense point clouds.

3.1.2. How PCA works in this task?

As preliminary step, the selection of the sub-sections in the large-sparse point cloud showing the most similar geometrical characteristics to the small-dense point cloud is done with PCA, used in two folds. First, the two dominant PCs (PC 1 and PC 2) are used to create a plane and then calculate the absolute distances between points and the PC-plane (as demonstrated in Figure 10 (a)); the statistical patterns of the point-to-PC-plane distances are plotted into histograms (as demonstrated in Figure 10 (b)), used as the evidence to measure the geometrical similarities. Figure 11 shows that the difference in geometrical characteristics of a point cloud causes the variation of the distribution pattern of the point-to-PC-plane distances: the 3D models in panels (a) and (b) feature the exact same geometry but in different orientations; the resulting point-to-PC-plane distances histograms appear similar. A third model shown in panel (c) with different geometrical features on the top surface has been simulated to test changes in the distribution pattern of the point-to-PC-plane distances (i.e. its histogram is recognisably different compared to those of models (a) and (b)).

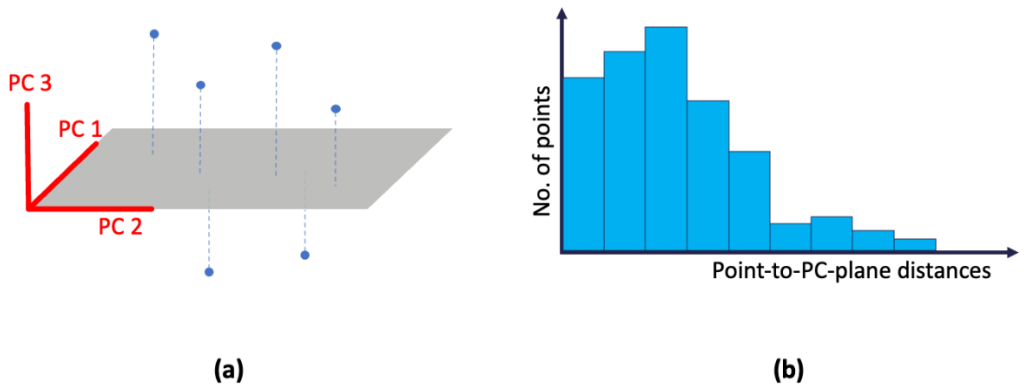


Figure 10 *A visual demonstration of using point-to-PC-plane distances as a statistical information of a point cloud. In image (a), the grey plane is formed by PC 1 and PC 2. The lengths of the dotted lines illustrate the absolute distances from points to the PC plane. The distribution of the absolute distances is plotted into a histogram for this point cloud like image (b).*

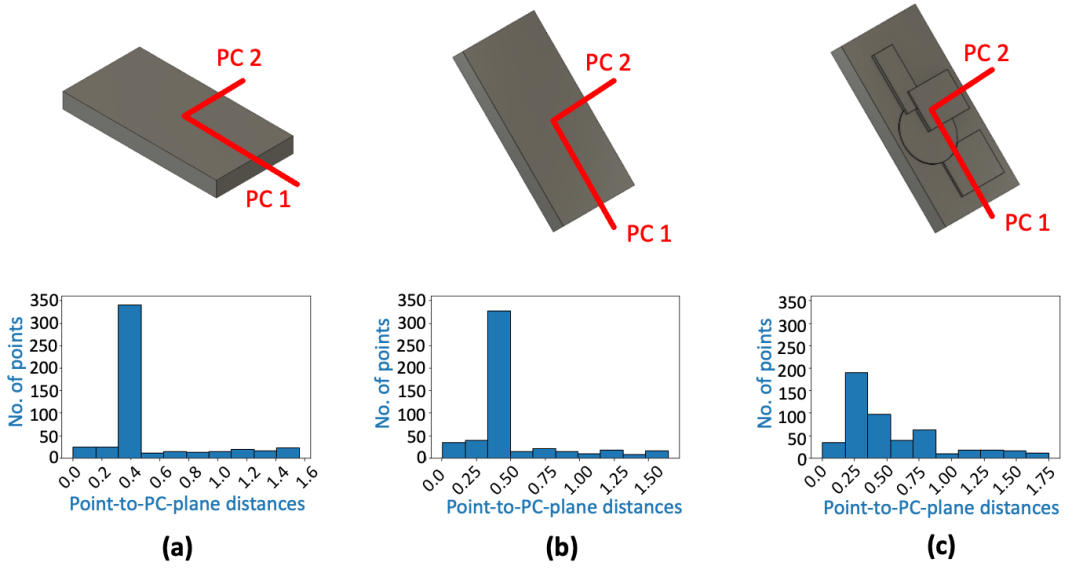


Figure 11 *The three images in the lower row are the histograms of the point-to-PC-plane distances of the point clouds representing the CAD models in the upper row. There are 1,000 points in each point cloud. The red lines in the upper row images represent the top two PCs for each point cloud. The 3D models in panel (a) and (b) are the same, obviously different from the model in column (c).*

Second, for the candidate sub-sections selected in the first fold and the small-dense point cloud, the angles between local norm vectors relative to the PC-plane of the point cloud are calculated within the range of $[0^{\circ}, 90^{\circ}]$. Similarly to the processing of the point-to-PC-plane distances, the angles of local norm vectors are sorted and visualised into a histogram. As shown in Figure 12 (a), the blue arrows emitted from the points represent the local norm vectors; in Figure 12 (b), the angles between local norm vectors and the PC-plane are plotted into a histogram for this point cloud. Point clouds with similar geometrical characteristics have similar local-norm-to-PC-plane angles distribution patterns, as it can be seen in Figure 13: the histograms of the local-norm-to-PC-plane angles of models in panels (a) and (b) are the same, even if their position and orientation are different. The geometrical characteristics of the model in panel (c) are different from (a) and (b), and its histogram shows once again a different distribution. In conclusion, the sub-sections in the large-sparse point cloud selected at this stage as best candidates are determined as the correct location for registration of the small-dense point cloud.

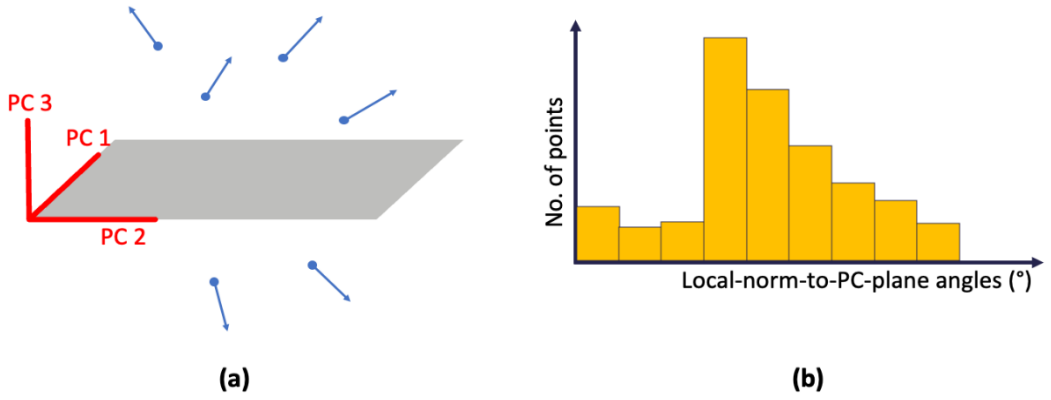


Figure 12 *A visual demonstration of using local-norm-to-PC-plane angles as a statistical information of a point cloud. In image (a), the grey plane is formed by PC 1 and PC 2. The arrows from the points represent the local norm vectors at these points. The distribution of these angles is plotted into a histogram for this point cloud like image (b).*

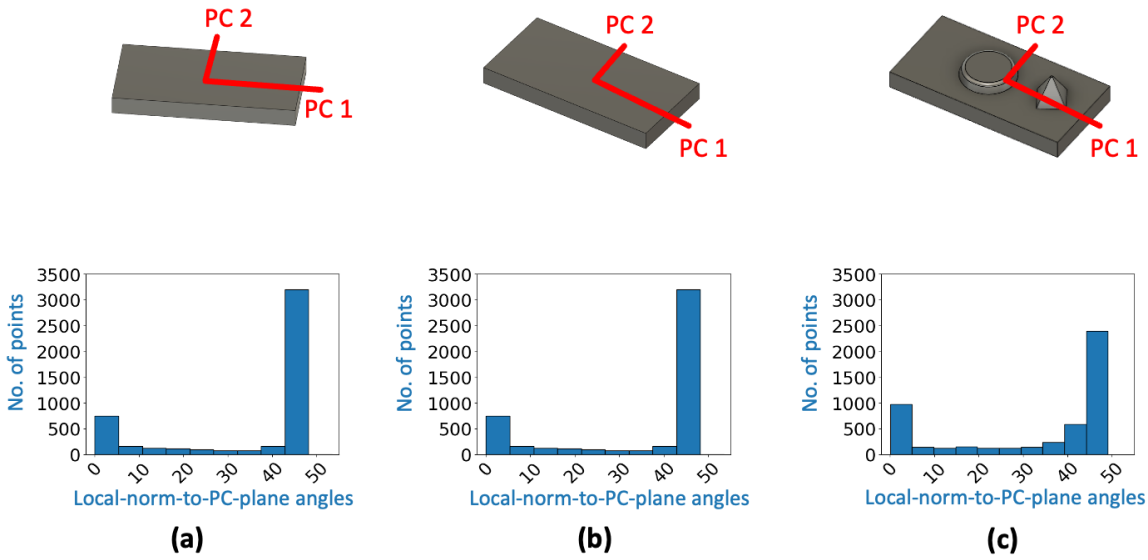


Figure 13 *The three images in the lower row are the histograms of the local-norm-to-PC-plane angles of the point clouds representing the CAD models in the upper row. The 3D models in column (a) and (b) are the same, obviously different from the model in column (c).*

As introduced above, to detect the potentially matched sub-clouds in the large-sparse point cloud, there are two folds of PCA implementation: the first time is to examine the distribution of point-to-PC-plane distances; the second time is to assess the local-norm-to-PC-plane angles. For the rationale behind this two-fold PCA, consider the following situation visualised in **Figure 14**: there are two 3D point clouds with rotational symmetrical geometries, whose side views are shown in **Figure 14** (a) and (b). Regarding the geometrical symmetry displayed by these two point clouds, the distributions of their point-to-PC-plane distances appear similar. However, their geometries are obviously different. As such, solely point-to-PC-plane distance is not sufficiently safe to detect the target sub-clouds which are potentially matched with the small-dense point cloud. The assessment of local-norm-to-PC-plane angles is a necessary complement.

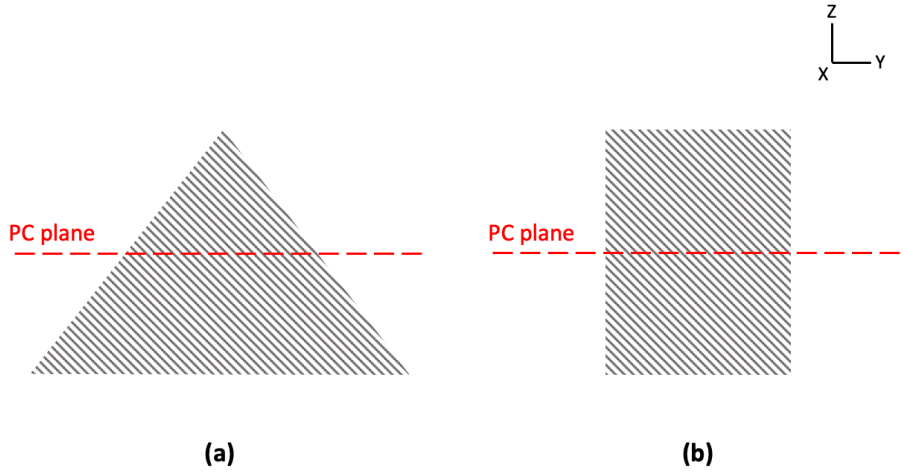


Figure 14 *The side view of two 3D point clouds, both of which are of rotational symmetrical shapes in 3D. The red dotted lines are their PC planes. Their histograms of point-to-PC-plane distances will appear ostensibly similar, but they are apparently not matched in terms of geometrical characteristics.*

3.2. Cross-sectional geometrical comparison

PCA can help determine the correct location (i.e. in this work, the best-matching sub-section of the large-sparse point cloud) for optimal registration but, in this pipeline, it appears as independent to the orientation of the datasets. For this reason, to determine the correct orientation of the small-dense point cloud with respect to the large-sparse, an additional statistical analysis has been performed to complete registration. Although the iterative closest point (ICP) algorithm and its variants have shown promising results for orientation determination, their application scenarios are restricted to point clouds with similar point densities [104–107]. As such, an orientation-searching algorithm designed specifically for datasets with significant disparity in point density is proposed. The algorithm searches for geometrical similarities in cross-sections of the selected geometry. The outline is as follows:

1) **Determine the *z*-axis of the small-dense point cloud**: the direction of the *z*-axis is perpendicular to the PC-plane of the small-dense point cloud; the origin of the *z*-axis overlaps the geometrical centroid of the small-dense point cloud (it is also the geometrical centroid of the target area in the large-sparse point cloud). The z-axis of the small-dense point cloud will be used for orientation changing.
2) **Voxelise the space**: after locating the small-dense point cloud into the target area, the space mutually occupied by the small-dense point cloud and the target area is subdivided into equal-size boxes (i.e. voxels), as Figure 15 (a) demonstrates. The directions in which the space is voxelised are defined as temporary *x*- and *y*-axis. The voxels on the same *z*-height form a "level" along the *z*-axis, as Figure 15 (b) shows.
3) **Count the points in each voxel**: For both point clouds, the points falling into each voxel are counted. Then, the number of points in each voxel is converted into the percentage relative to the total number of points in each point cloud. This process is shown in Figure 15 with a simplified example: a 3D point cloud (red object in (a)) is voxelised into a grid of ($2 \times 2 \times 3$) voxels. Along the *z*-direction, there are three levels of voxels each containing (2×2) voxels. The statistical result of point distribution in Level 1 is shown in (b): the number of points falling into each voxel is converted into a percentage, indicated as A %, B %, C % and D % in this figure. This process can be generalised to more complex shaped point clouds and voxelisations in real cases.
4) **Compare the percentages in each voxel**: the percentages of points in each voxel belonging to the small-dense and the target area in the large-sparse clouds are compared. If the percentages of these two point clouds in a voxel are similar, this

voxel is marked as "matched" voxel. The number of matches on each level is denoted as n_{level}: the larger the n_{level} value, the more similar the cross-sectional shapes in this level of z-height.

5) **Assess the level of geometrical similarity**: with all levels of voxels assessed, the similarity score between the small-dense point cloud in this specific orientation and the target sub-section of the large-sparse point cloud is determined as:

$$similarity\ score = \sum_{i=1}^{n_z} n_i W_i, \quad (6)$$

where i is the level index, n_z is the total number of levels and W_i is the weight for each level. The values of the weights are customised by the user. In this specific research project, weights are assigned in the open interval $(0, 1)$.

6) **Find the optimal orientation**: rotate the small-dense point cloud about its z-axis determined in Step 1) for a constant user-defined angle, giving the next orientation of the small-dense point cloud; repeat Step 2) to 5) for this orientation to obtain a similarity score. After all orientations have been assessed, compare all similarity scores; the orientation that results in the highest similarity score is determined as the correct one for registration.

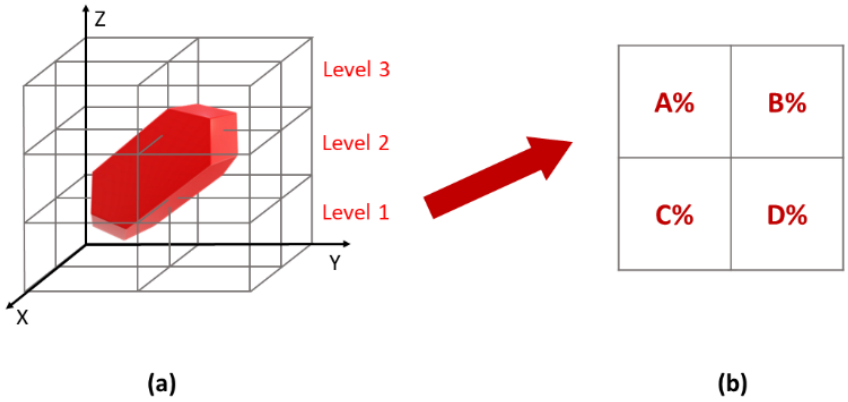


Figure 15 *This is a visual demonstration of the point cloud voxelisation process. (a) The 3D visualisation of voxelisation of the space occupied by a point cloud (red object). (b) The percentages of points, relative to the total number of points in the red point cloud, in the voxels in Level 1.*

In abstract, the pillar idea of this spatial geometrical similarity algorithm is to assess the point distributions of the two point clouds in 3D space. It should also be noted that the number of orientations attempted, i.e. the degrees between two consecutive orientations, is defined by the user; the more orientations are defined (i.e. the smaller the angle between two consecutive orientations is), the more accurate the final registration becomes. However, it is also worth noting that a large number of orientations requires a higher computational cost.

3.3. Summary

In this chapter, the methods to determine the target area and orientation of registration were introduced. By converting the point distributions in 3D space into histograms based on PCA, the comparison between two 3D datasets is converted into a comparison between two histograms. This theory can construct a pattern recognition algorithm without relying on massive training data. By voxelising the mutually occupied 3D space by two point clouds into identically sized voxels, and converting the number of

points falling into each voxel into a percentage relative to the total points in the corresponding point cloud, the detection of the optimal orientation becomes robust against disparity in point density.

The mathematical foundation has been established, but there is a gap between mathematics and practicable computer algorithms. In the next chapter, this gap will be filled.

Chapter 4

Structure of the pipeline

In Chapter 3, the theoretical methods to detect the correct area and orientation to register the small-dense point cloud were founded. In this chapter, the theory will be engineered into an algorithmic pipeline which is programmable. The flow of whole registration pipeline consists of the steps shown in Figure 16. The technical details of these algorithms will be introduced in sections 4.1, 4.2 and 4.3.

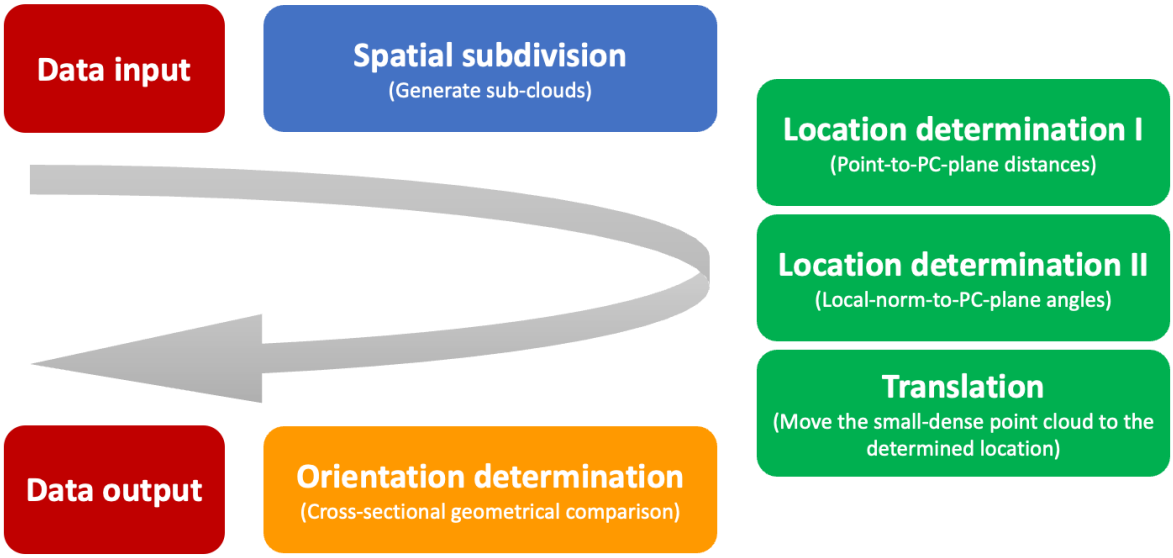


Figure 16 *The structure of the algorithmic pipeline.*

4.1. Initial data processing

First of all, within working volume for this pipeline is a 500 × 500 × 500 mm cubic space, the large-sparse point cloud is collected by an optical measurement instrument while the small-dense point cloud is captured by another optical measurement instrument. Regarding the general configuration of the optical instruments utilised for surface texture capturing tasks (such as the facility in the laboratory employed for this research as specified in section 5.1), the x and y dimensions of the captured small-dense point cloud are always equal. The input point clouds should be noise-free. The z-direction of the small-dense point cloud should be roughly aligned with the z-direction of the area on the engineered object from which the small-dense point cloud is captured; the z-direction is determined by the norm vector of the PC-plane. In other words, it is assumed

that the user, who used optical instruments to capture the two point clouds from the engineered object, remembers from which surface of the engineered object the small-dense point cloud is captured. It is assumed that the input point clouds have been pre-processed according to the aforementioned assumptions.

After the input point cloud datasets have been loaded into the pipeline, the two point clouds are processed in two independent coordinate systems. The sizes of both point clouds are measured and determined. Then, the large-sparse point cloud is subdivided equally into sub-sections, called "sub-clouds" from this point onwards. Each sub-cloud has the same dimensions as the small-dense point cloud. The large-sparse point cloud is not divided in a grid pattern, as it will decompose the potential target area into several separate sub-clouds and hence make the pipeline miss the target area. Instead, the neighboured sub-clouds share overlapped areas (i.e. there are points contained in both neighbouring sub-clouds): the distance between two consecutive areas is denoted as s. Figure 17 shows the technical details of the subdivision process in 2D on the x-y plane: the red square represents the size of the corresponding small-dense point cloud and the starting location of the subdivision process, i.e. the first sub-cloud divided from the large-sparse point cloud (the grey object). The orange square in dotted lines represents the area covered by the neighbouring sub-cloud to the red one along the x-direction. The step size between these two sub-clouds is s_x. The purple square in dotted lines represents the area covered by the neighbouring sub-cloud to the red one along the y-axis. The step size between these two sub-clouds is s_y. The same subdivision method is repeated in the z-direction. As Figure 17 demonstrates, neighbouring sub-clouds share overlapped areas.

The trivial sub-clouds (sub-clouds with few or no point) are automatically detected and deleted by the pipeline after the subdivision.

It should be noted that the values of s_x, s_y and s_z are defined by the user based on their preference of accuracy: the smaller these steps are, the larger the area overlapped between two neighbouring sub-clouds, and hence the detection of the borders of the correct target area in future steps becomes more accurate. However, smaller subdivision steps also lead to a larger amount of sub-clouds, and hence the computational cost for analysing and comparing them can increase rapidly.

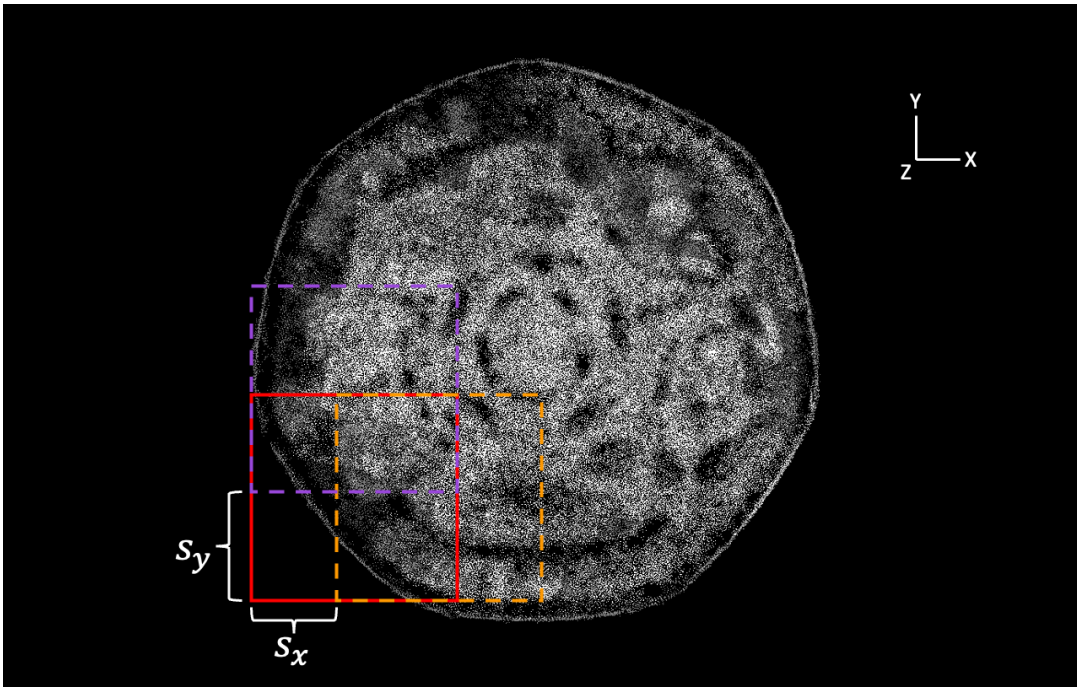


Figure 17 *2D visualisation of the subdivision on x-y plane of a large-sparse point cloud (grey). The area enclosed by the red square is the first sub-cloud divided from the large-sparse point cloud. The area enclosed by the dotted orange square covers the sub-cloud neighbouring the first one along x-axis. The area enclosed by the dotted purpose square covers the sub-cloud neighbouring the first one along y-axis.*

4.2. Detect the target area

Suppose there are N sub-clouds generated from the large-sparse point cloud. These sub-clouds are fed into the similarity comparison stage using PCA, as discussed in section 3.1. Their geometrical characteristics are firstly compared to those of the small-dense point cloud using the point-to-PC-plane distances. When generating the histograms for the point-to-PC-plane distances, considering the difference in point densities between point clouds, the number of points on the *y*-axis of the histogram is converted to the percentage of points relative to the total number of points in the point cloud under comparison (for the convenience of expression, it will be called "the percentage of point" in the following text). As such, the height of each bin represents the percentage of points falling into it.

When comparing the histograms of the point-to-PC-plane distances, the algorithm firstly plots histograms with T_1 ($T_1 \in \mathbb{Z}^+$) bins for each pair of sets, with *x*-axis as the point-to-PC-plane distances and *y*-axis as the percentage of points. The sum of bin heights is 1 (i.e. 100% of the points in the point cloud). The heights of these bins are denoted as $\{h_{1,i}\}_{i=0,1,\dots,T_1}$. The positions of these bins in their histogram are determined by their mid x-location, i.e.

$$x_{1,i} = \frac{x_{1,i,lower-boundary} + x_{1,i,upper-boundary}}{2}, i = 0, 1, \dots, T_1. \quad (7)$$

Then, the T_1 bins in the histogram of the sub-cloud and those in the histogram of the small-dense point cloud are compared. Among all T_1 bins in both histograms, if there are exactly or more than T_1' ($T_1' \leq T_1$) bins that satisfy

$$\left\{\frac{h_{1,i+1}}{h_{1,i}}\right\}_{sub-cloud} \leq \left\{\frac{h_{1,i+1}}{h_{1,i}}\right\}_{small-dense} \times \delta_{large} \text{ and } \left\{\frac{h_{1,i+1}}{h_{1,i}}\right\}_{sub-cloud} \geq \left\{\frac{h_{1,i+1}}{h_{1,i}}\right\}_{small-dense} \times \delta_{small}, \quad for\ i = 1, 2, \dots, T_1' - 1\ ; \tag{8}$$

and

$$\{x_{1,i}\}_{sub-cloud} \leq \{x_{1,i}\}_{small-dense} \times \varepsilon_{large} \text{ and } \{x_{1,i}\}_{sub-cloud} \geq \{x_{1,i}\}_{small-dense} \times \varepsilon_{small}, \quad for\ i = 1, 2, \dots, T_1', \tag{9}$$

then this sub-cloud will be selected as a candidate for potential matching with the small-dense point cloud ($\delta_{large}, \delta_{small}, \varepsilon_{large}$ and ε_{small} are called tolerance parameters, with $\delta_{large} > \delta_{small}$ and , $\varepsilon_{large} > \varepsilon_{small}$). The mathematics in this comparison step can be intuitively explained as follows: I compare the ratios between the heights of each two neighbouring bins in the two histograms; then I compare the *x*-locations of corresponding bins in the two histograms. If the comparison results satisfy the thresholds listed above, these two histograms are determined as similar and hence the sub-cloud is selected as a candidate for further comparison. The mathematical principle for the step below (the second round of PCA) is in the same manner. Here, it is worth noting that this step is not an iterative process as it is not aimed to converge to a certain static value. This is a comparison step between two sets of ratios reflecting the relative heights between each two neighbouring bins in two histograms: one of a sub-cloud, the other of the small-dense point cloud.

All sub-clouds are then fed into the next round of PCA comparison, which is based on the local-norm-to-PC-plane angles (section 3.1). The algorithm firstly plots

histograms with T_2 ($T_2 \in \mathbb{Z}^+$) bins for each candidate sub-cloud and the small-dense point cloud, with *x*-axis as the local-norm-to-PC-plane angles in degree and *y*-axis as the percentage of points. The sum of bin heights equals to 1. Then, the top T_2' bins (i.e. bins with the largest T_2' heights, $T_2' \leq T_2$) are found by the algorithm. The positions of these top bins in their histogram are determined by their mid x-location, i.e.

$$x_{2,i} = (x_{2,i,lower-boundary} + x_{2,i,upper-boundary})/2, i = 0, 1, 2, \dots, T_2'. \quad (10)$$

If the two histograms show top bins which satisfy

$$\{x_{2,i}\}_{sub-cloud} \leq \{x_{2,i}\}_{small-dense} \times \theta_{large} \text{ and } \{x_{2,i}\}_{sub-cloud} \geq \{x_{2,i}\}_{small-dense} \times \theta_{small}, \quad for\ i = 1, 2, \dots, T_2', \quad (11)$$

then this sub-cloud will be selected as a match with the small-dense point cloud (θ_{large} and θ_{small} are tolerance parameters, with $\theta_{large} > \theta_{small}$). The positions of the candidates selected in this step are then compared to the positions of the ones selected in the previous step, the analysis based on the point-to-PC-plane distances. If one sub-cloud selected in the assessment of local-norm-to-PC-plane angles is identical or neighbouring a sub-cloud selected in the assessment of the point-to-PC-plane distances, both sub-clouds will be considered matches to the small-dense point cloud.

As the matched sub-clouds selected by two analysis based on PCA are anticipated to be closely neighbouring and partially overlapping each other, forming a target area in the large-sparse point cloud, the location for registering the small-dense point cloud is the geometrical centroid of the target area covered by the matched sub-clouds, i.e. the small-dense point cloud will be translated to the mutual geometrical centroid of all matched

sub-clouds. The location of the geometrical centroid of the small-dense point cloud will be at the geometrical centroid of the target area. The coordinate of the target area is calculated as

$$Registration\ location = \frac{\sum_{i=1}^{\mathcal{N}}(x_i, y_i, z_i)}{\mathcal{N}} \quad (12)$$

where $\mathcal{N}$ is the total number of unique points covered in the matched sub-clouds and (x_i, y_i, z_i) are the coordinates of the unique points covered in the matched sub-clouds (as the sub-clouds have overlapped points, a unique point is counted only once).

4.3. Determine the correct orientation

After the small-dense point cloud has been translated to the registration location in the large-sparse point cloud, the next round of similarity comparison is aimed at determining the correct orientation of the former. The small-dense point cloud is duplicated into O ($O \in \mathbb{Z}^+$) copies, with each rotated anticlockwise for $\left(\frac{360}{O}\right)°$ from the previous one about the z-axis through its geometrical centroid. The parameter O is defined by the user, depending on the level of accuracy of final registration they desire. As one can see, the larger the number of positions O, the smaller the difference in orientations between each two neighbouring positions and hence more optimal the registered orientation. However, a larger number of attempts also lead to higher computational costs. As such, it is reasonable to say that the user should configure the value of O as larger as possible observing the computational capability they have.

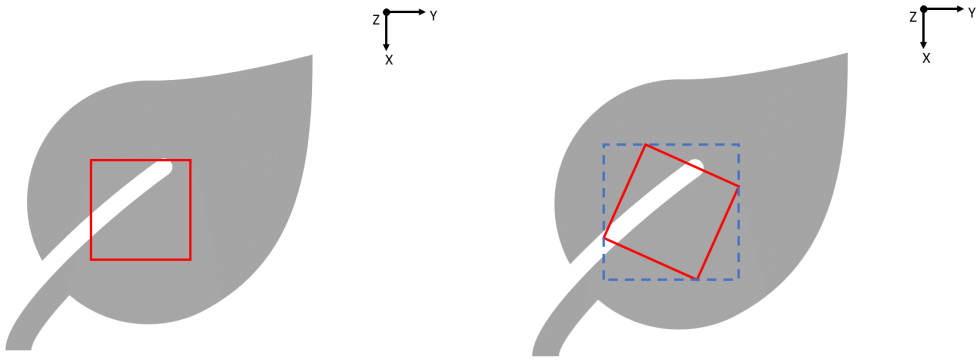
For the small-dense point cloud in each orientation, denoted as SD_o ($o = 1, 2, \ldots, O$), the small-dense point cloud and the target area in the large-dense point

cloud covered by the dimensions of the small-dense point cloud are both voxelised into the same number of voxels along *x*-, *y*- and *z*- directions. For both the small-dense point cloud and the target area in the large-sparse point cloud, the numbers of voxels on *x*-, *y*- and *z*-directions are denoted respectively as n_x, n_y and n_z. As such, there are totally $n_x \times n_y \times n_z$ voxels for each point cloud. The values of n_x, n_y and n_z are user-defined; the more voxels are divided in the 3D space, the more meticulous the assessment of spatial point distribution will be. However, the user is advisedly reminded that the more voxels they generated in this stage, the more computational cost will be generated, too. In addition, when the difference of point density between two point clouds is sufficiently large and the number of voxels is sufficiently large, a phenomenon will occur: the number of points contained in the large-sparse point cloud falling into each voxel will be minimal and hence the statistical meaning of this procedure will be lost. This is what the user should be aware when implementing this step.

To make this rotation-and-voxelisation process more approachable, there is a simplified visual demonstration given in Figure 18 in a 2D scale observed from the z-axis: the red square in solid lines represents the orientation and location of the small-dense point cloud; the blue square in dotted lines represents the space that is voxelised for both the small-dense point cloud and the large-sparse point cloud (grey object). For the small-dense point cloud in orientation (a), the space for voxelisation is the same as the space enclosed by the red square. However, when the small-dense point cloud is in orientation (b), the space for voxelisation is the space enclosed by the blue square, which is different from the space enclosed by the dimensions of the small-dense point cloud.

Then, as illustrated and explained in section 3.2, for both SD_o and the target area, the percentage of points falling into each voxel is counted and calculated. The SD_m, i.e. the small-dense point cloud that is rotated for $(m-1) \times \left(\frac{360}{o}\right)°$ about the z-axis from its initial orientation, which gives the highest similarity score (to review the mathematics of similarity score, please see section 3.2) in this comparison process is determined as the correct orientation for registration.

Finally, the small-dense point cloud with the registered location and orientation is exported as a text file with three columns of (x, y, z) coordinates of the points. It can be displayed together with the coordinate dataset of the large-sparse point cloud in any visualisation software tool. The user can visualise the detailed surface information on the area covered by the small-dense point cloud.



(a) (b)

Figure 18 *The area for voxelisation changes with the orientation of the small-dense point cloud, as the area in the large-sparse point cloud covered by the dimensions of the small-dense point cloud in a different orientation is different. From (a) to (b), the small-dense point cloud (red square) is rotated from an angle, and hence the area for voxelisation (dotted blue square) is changed.*

4.4. Overview of the pipeline

To give a big-picture view of how each algorithm is integrated into this pipeline, as stated in Chapter 3 and sections 4.2 and 4.3 in this chapter, all parameters in this data fusion pipeline are listed in Table 2. A visual explanation of the coordination between algorithms is shown in Figure 19, in which the reader can see the stage where each mathematical equation mentioned sections 4.2 and 4.3 is implemented.

Table 2 *Parameters of the data fusion pipeline**

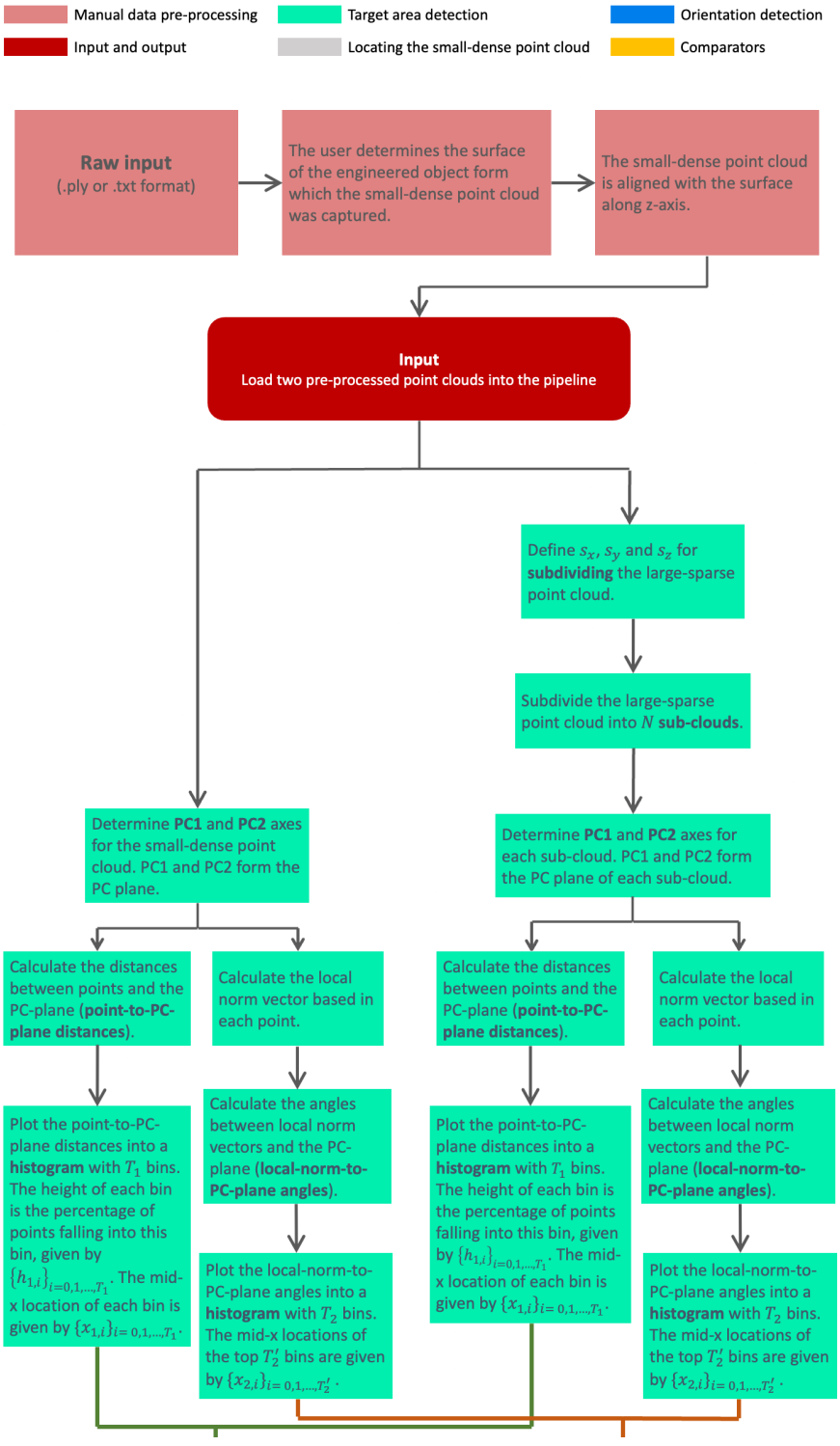
** The parameters are listed in the order of first appearances in this paper.*

*** the index i of W_i is enumerated from the lowest z-coordinate in the point cloud.*

**** Please be aware of the difference between $\mathcal{N}$ and N in terms of calligraphy.*

Parameters	If defined by user?	Explanation
W_i **	Y	The weight for calculating the similarity score for each level of voxels.
n_i	N	The number of matched pairs of voxels in each level of voxels.
s_x, s_y, s_z	Y	The sizes of steps of subdividing the large-sparse point cloud into sub-clouds on *x*-, *y*- and *z*-directions.
N	N	The number of sub-clouds.
T_1	Y	The number of bins in the histograms of the point-to-PC-plane distances.
T_1'	N	The number of bins that are matched in the histograms of the point-to-PC-plane distances for the two point clouds.
$\{h_{1,i}\}_{i=0,1,2,\dots,T_1}$	N	The heights of the bins in the histograms of the point-to-PC-plane distances.
$\{x_{1,i}\}_{i=0,1,2,\dots,T_1}$	N	The locations (the midlines on the *x*-axis) of the bins in the histograms of the point-to-PC-plane distances.
$\delta_{large}, \delta_{small}$	Y	The tolerance parameters for comparing the bin height ratios in the histograms of the point-to-PC-plane distances

$\varepsilon_{large}, \varepsilon_{small}$	Y	The tolerance parameters for comparing the bin locations in the histograms of the point-to-PC-plane distances
T_2	Y	The number of bins in the histograms of the local-norm-to-PC-plane angles.
T_2'	Y	The number of bins with the largest heights (top bins) in the histograms of the local-norm-to-PC-plane angles.
$\{x_{2,i}\}_{i=0,1,2,\ldots,T_2'}$	N	The locations of the top bins in the histograms of the local-norm-to-PC-plane angles.
$\theta_{large}, \theta_{small}$	Y	The tolerance parameters for comparing the locations of the top bins in the histograms of the local-norm-to-PC-plane angles.
$\mathcal{N}^{***}$	N	The total number of unique points in the matched sub-clouds (target area in the large-sparse point cloud selected with PCA)
O	Y	The number of orientations of the small-dense point cloud.
o	N	The index of the small-dense point cloud in a certain orientation.
n_x, n_y, n_z	Y	The numbers of voxels on *x*-, *y*- and *z*-directions.


Manual data pre-processing
Target area detection
Orientation detection
Input and output
Locating the small-dense point cloud
Comparators
Raw input
(.ply or .txt format)
The user determines the surface of the engineered object form which the small-dense point cloud was captured.
The small-dense point cloud is aligned with the surface along z-axis.
Input
Load two pre-processed point clouds into the pipeline
Define s_x, s_y and s_z for **subdividing** the large-sparse point cloud.
Subdivide the large-sparse point cloud into N **sub-clouds**.
Determine **PC1** and **PC2** axes for the small-dense point cloud. PC1 and PC2 form the PC plane.
Determine **PC1** and **PC2** axes for each sub-cloud. PC1 and PC2 form the PC plane of each sub-cloud.
Calculate the distances between points and the PC-plane (**point-to-PC-plane distances**).
Calculate the local norm vector based in each point.
Calculate the distances between points and the PC-plane (**point-to-PC-plane distances**).
Calculate the local norm vector based in each point.
Plot the point-to-PC-plane distances into a **histogram** with T_1 bins. The height of each bin is the percentage of points falling into this bin, given by $\{h_{1,i}\}_{i=0,1,\dots,T_1}$. The mid-x location of each bin is given by $\{x_{1,i}\}_{i=0,1,\dots,T_1}$.
Calculate the angles between local norm vectors and the PC-plane (**local-norm-to-PC-plane angles**).
Plot the point-to-PC-plane distances into a **histogram** with T_1 bins. The height of each bin is the percentage of points falling into this bin, given by $\{h_{1,i}\}_{i=0,1,\dots,T_1}$. The mid-x location of each bin is given by $\{x_{1,i}\}_{i=0,1,\dots,T_1}$.
Calculate the angles between local norm vectors and the PC-plane (**local-norm-to-PC-plane angles**).
Plot the local-norm-to-PC-plane angles into a **histogram** with T_2 bins. The mid-x locations of the top T_2' bins are given by $\{x_{2,i}\}_{i=0,1,\dots,T_2'}$.
Plot the local-norm-to-PC-plane angles into a **histogram** with T_2 bins. The mid-x locations of the top T_2' bins are given by $\{x_{2,i}\}_{i=0,1,\dots,T_2'}$.

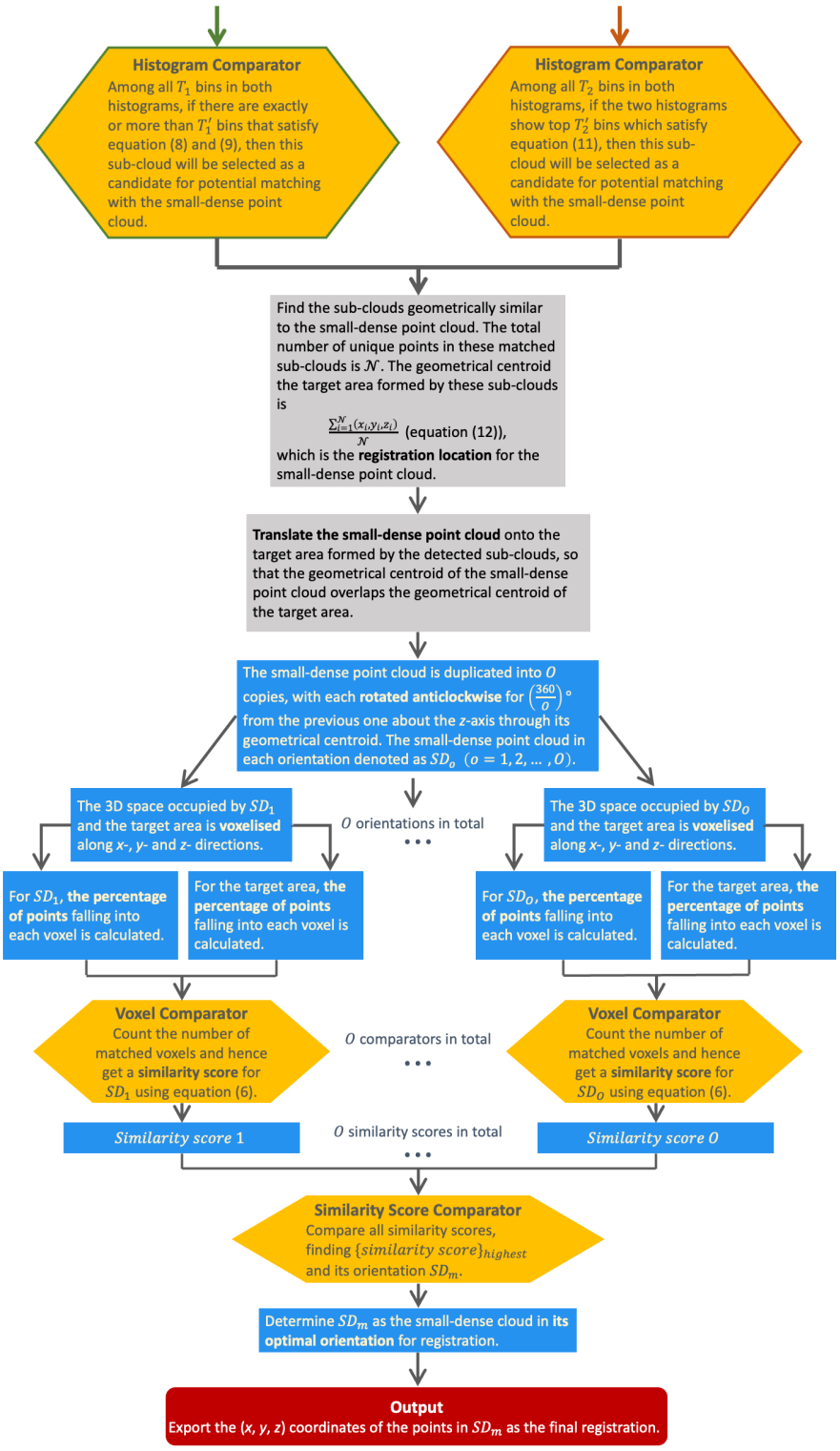


Figure 19 *A detailed flowchart of the pipeline. All mathematical equations mentioned in this chapter are shown on this flowchart at their functioning stages.*

4.5. Summary

In this chapter, the theories and concepts introduced in Chapter 3 were further exposited in a concrete manner, i.e. how these new theories and concepts should he implemented with mathematical expressions so that a computer can understand. More importantly, in this chapter, the theoretical solution introduced in Chapter 3 was translated into a pipeline with systematically structured algorithms and steps; the practical assumptions and application settings were stated. In the next chapter, the performances of this pipeline in synthetic point clouds and real engineered parts will be introduced and visualised.

Chapter 5

Experimental study

In history, I have frequently witnessed the tedious outcomes of beautiful theories in practical scenarios; and this sort of catastrophe what should be avoided in this research. Before getting implemented in a real commercial case, the performance of this pipeline must be tested and assessed in various complexities. In this chapter, the design of test artefacts and the outputs will be presented.

5.1. Experimental setup

To prove the performance of the algorithmic pipeline, three experimental tests were proposed, featuring test cases with different geometrical complexities. The first two test artefacts (indicated as Case I and II) are point clouds generated from synthetic 3D models designed in CAD software. Between these two 3D models, the one in the first study is geometrically simpler than the one in the second study. The third case study (indicated as Case III) is a real measurement, more specifically the external shape coordinates and surface texture point clouds of a 20 pence coin issued by the Royal Mint (UK, 1989) [6,18].

5.1.1. 3D model design

The purpose of testing the pipeline with synthetic 3D models first (Case I and II) is to prove its feasibility and potential for real industrial scenarios. The proposed case studies are characterised by fundamental geometrical primitive features, such as tapered cylinders, cones, pyramid-like shapes, sharp edges, chamfers and fillets. Given the designated working volume of this pipeline, the dimensions of the 3D model are always within the cubic space of $500 \times 500 \times 500$ mm.

The models are converted into point clouds (i.e. the surface of the models is sampled into 3D points) of large size and low point density (i.e. sparse). For each case, a small area in the 3D model is isolated and used to generate the synthetic small-dense point cloud, simulating a surface texture measurement. However, CAD models have smooth surfaces (i.e. surfaces without textures generated during manufacturing procedures), which is an unrealistic scenario. To make the experimental study as close to reality as possible, the CAD software was used to generate patches of synthetic engineered surface patterns and paste them on the surface of the selected area in the 3D model (as will be introduced in section 5.1.2). As such, the small-dense point cloud generated from this surface-texture-covered area can better imitate the geometrical features of the surface texture point cloud of a real engineered part.

5.1.2. Software

The 3D models were designed and exported with Autodesk® Fusion 360 for Education [108]. The synthetic surface texture patches were generated with Image2Surface, an add-in to Fusion 360 [109]. The add-in converted a surface texture

image to a 3D model of surface texture, which could be attached to the surface of a 3D model. An example of image-converted surface texture model is demonstrated in Figure 20.

After the 3D models had been joined with the synthetic surface textures patches, the completed models were exported as .OBJ files, and then sampled as point clouds (.PLY format) with CloudCompare [110], also used to visualise and analyse the outputs of the registration attempts.

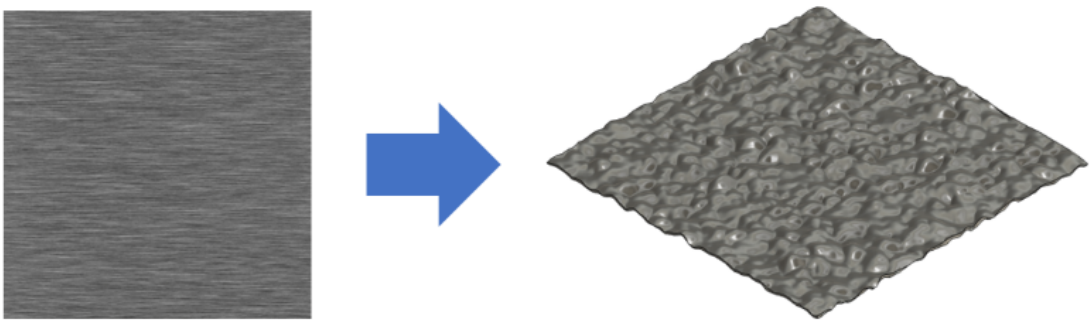


Figure 20 *An image of a metal surface texture (left) is converted to a 3D surface texture (right) by Image2Surface* [109].

5.1.3. Measuring instruments

For the real case scenario (Case III), the instrument used to collect the coordinate data points of the coin is GOM ATOS Core 300. Its technical specifications are listed in Table 3, as provided by the manufacturer. Because of the small size of the coin, it was captured at 20 different poses to get a detailed result (i.e. the final point cloud is the result of 20 consecutive scans, stitched together by the commercial software paired to the instrument).

Table 3 *Technical details of GOM ATOS Core 300*

Parameter	Value
Field of View (FoV)	(300 x 230) mm
Probing Error Form (σ)	0.006 mm
Probing Error Size	0.027 mm
Sphere Spacing Error	0.020 mm
Length Measurement Error	0.047 mm

The surface texture data of the coin was collected by Zygo NexView NX2, a commercial coherence scanning interferometer. Its technical specifications are listed in Table 4, as provided by the manufacturer. The original data collected by Zygo NexView NX2 is in form of 2.5D, i.e. the data points are distributed homogeneously in a 2D grid, and each point is given a height value on the observed surface, mathematically $z \Leftarrow f(x, y)$). The data were converted into a 3D point cloud before further processes. The surface texture point cloud of the coin was collected with 1.4× magnification, giving an area of observation of (6318.986 × 6318.986) μm^2.

Table 4 Technical details of Zygo NexView NX2

Parameter	Value
Surface Topography Repeatability	0.06 nm (for all magnifications)
Repeatability of the RMS	0.005 nm
Tilt	± 4°
Sphere Spacing Error	0.020 mm
Capacity	20 lbs

5.2. Case I: a relatively simple 3D model

5.2.1. Geometrical features

The synthetic 3D model, from which the large-sparse and small-dense point clouds are generated, is displayed in Figure 21: the model in (a) is converted into the large-sparse point cloud. The maximum *z*-height in this model is marked with point A, where the z-coordinate is 3.14 mm. The area in (a) for generating the small-dense point cloud is enclosed in the red square, which is displayed in (b) on the right-hand side. The square patches on the surface of the model in (b) are synthetic surface textures (section 5.1.2). The properties of the point clouds generated from the model are in Table 5.

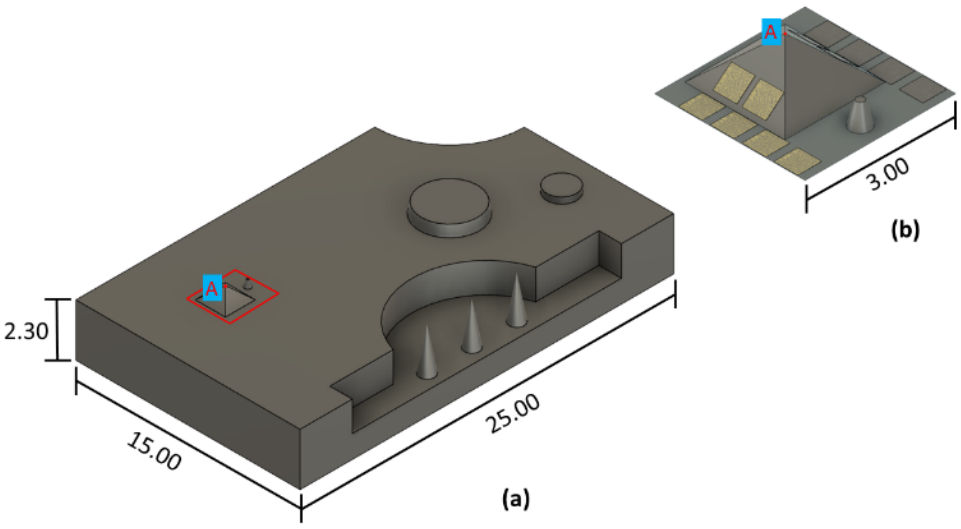


Figure 21 *The dimensions of the 3D model used to generate point clouds for Case I (Length unit: mm). (a) is the CAD model of the whole engineered part, (b) is the CAD model of the interested region in (a), which is enclosed by a red square. Point A marks the highest altitude of the model.*

Table 5 *The properties of the point clouds in Case I*

** The point clouds only represent the external geometries of an engineered part, which means the points are on the surface area of this object. Same for the point densities of the point clouds in Case II and Case III.*

Point clouds	Point count	Point density* /mm²
Large-sparse	49,992	44
Small-dense	99,677	4,375

5.2.2. Pipeline parameters

The user-defined parameters of the pipeline for Case I are listed in Table 6 (to review the parameters of this pipeline, please see section 4.4).

Table 6 *User-defined parameters of the data fusion pipeline in Case I*

** The small-dense point cloud is rotated 10° for each attempt.*

Parameters		Values
W_i	W_1	35
	W_2	35
	W_3	15
	W_4	10
	W_5	5
s_x, s_y, s_z	s_x	0.5 mm
	s_y	0.5 mm
	s_z	0.5 mm
T_1	8	
$T_1{}'$	4	
$\delta_{large}, \delta_{small}$	δ_{large}	1.18
	δ_{small}	0.82
$\varepsilon_{large}, \varepsilon_{small}$	ε_{large}	1.276
	ε_{small}	0.724
T_2	8	
T_2'	2	
$\theta_{large}, \theta_{small}$	θ_{large}	1.0964
	θ_{small}	0.9036
O	36 *	
$n_x,\ n_y,\ n_z$	n_x	10
	n_y	10
	n_z	5

5.2.3. Registration

With the parameters given in section 5.2.2, the algorithm subdivided the large-sparse point cloud into 7,201 sub-clouds, two of which were detected as the target sub-clouds for registration. These two target sub-clouds are adjacent, forming a continuous neighbouring area that covers the matched region in the large-sparse point cloud, marked as the area enclosed by a white rectangle in Figure 22 (for the convenience of visualisation, the area covered by the detected sub-clouds is marked on the CAD modelled used for generating the large-sparse point cloud, same for the following case study). As explained in section 4.2, the mutual geometrical centroid of these two sub-clouds is the centroid of the registered small-dense point cloud. The best orientation of the small-dense point cloud was detected with the procedure presented in section 3.2.

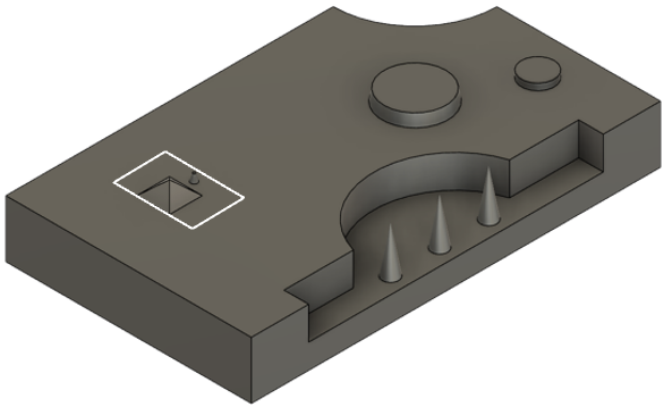


Figure 22 *The target area for registration in the large-sparse point cloud formed by the two target sub-clouds detected by the algorithm in Case I (enclosed in the white rectangle).*

The results of the registration can be seen in Figure 23 (the white object is the registered small-dense point cloud onto the surface of the 3D model) and the colourmap of the deviations after registration is displayed in Figure 24 (i.e. colourmap of the point

to surface signed distances). The smaller the value of the absolute deviations, the better the registration is. The histogram of the signed distances at each point in the small-dense point cloud is displayed in Figure 25.

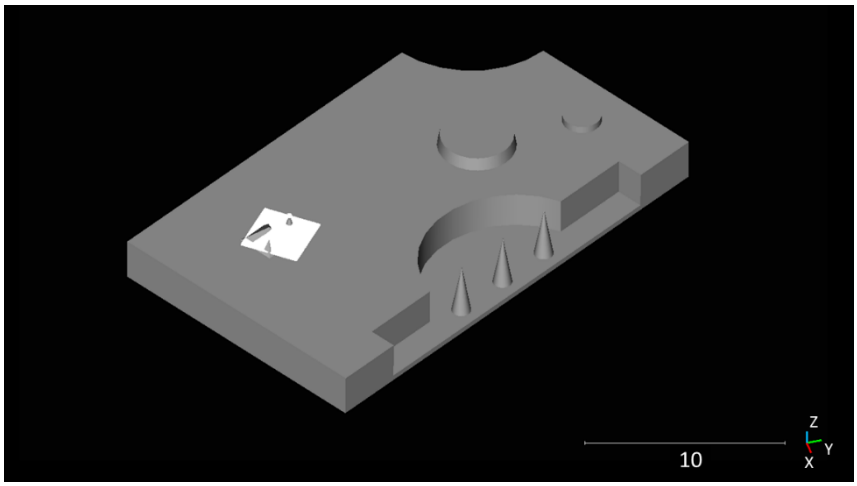


Figure 23 *The registration result of Case I. For a better visualisation, I use the CAD model to represent the large-sparse point cloud (same for the following chapters). Due to the high point density of the small-dense point cloud, geometrical features cannot be seen in this image.*

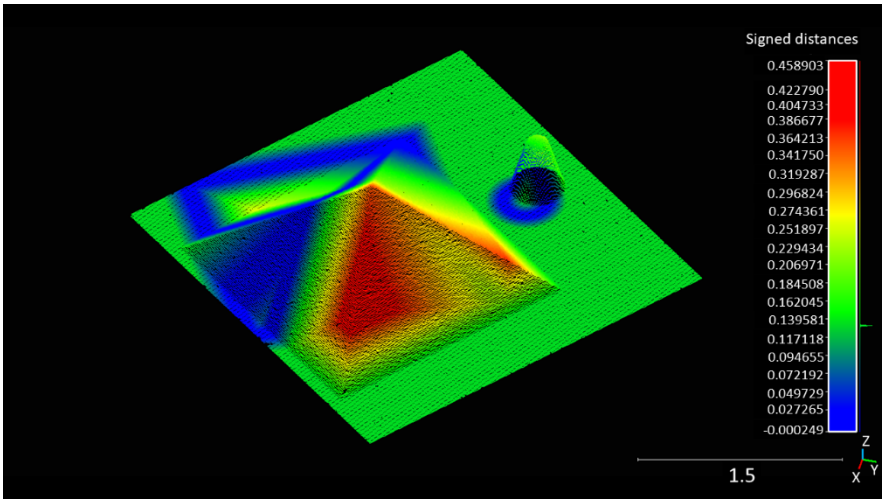


Figure 24 *The colourmap of the deviations from the 3D model (i.e. signed distances between the points in the small-dense point cloud and the surface of the 3D model). The unit is mm (same for the peer image in sections 5.3.3).*

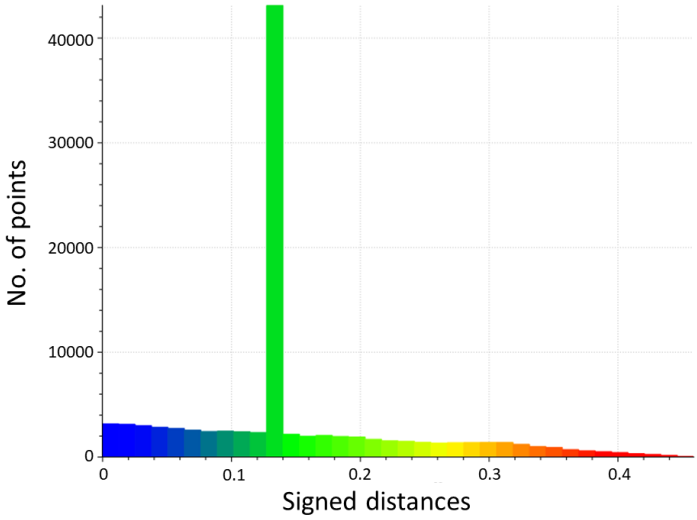


Figure 25 *Histogram of the signed distances in Case I. Mean value: 0.15 mm; standard deviation: 0.09 mm.*

5.3. Case II: a 3D model with more geometrical features

5.3.1. Geometrical features

The synthetic 3D model in Case II is displayed in Figure 26: the model in (a) is converted into the large-sparse point cloud. The maximum z-height of this model is marked with point A, where the z-coordinate is 3.25 mm. The area in (a) to generate the small-dense point cloud is enclosed in the red square, which is displayed in (b). The square patches on the surface of the model in (b) are synthetic surface textures. The properties of the point clouds generated from the model are in Table 7.

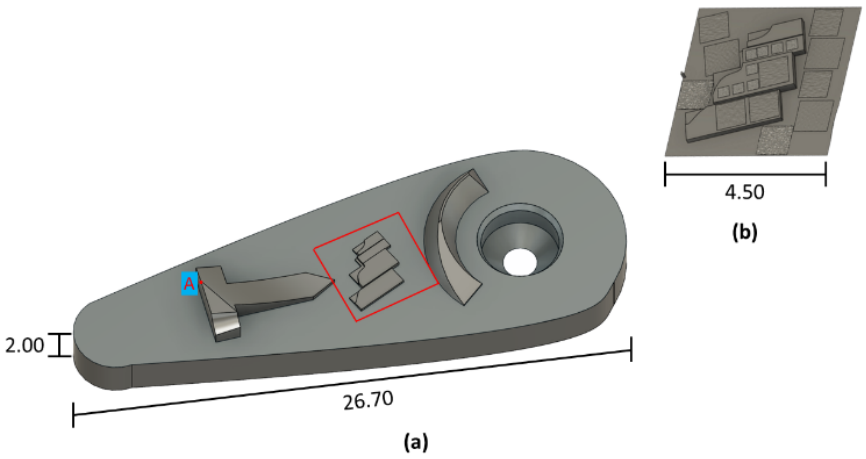


Figure 26 *The dimensions of the 3D model used to generate point clouds for Case II. (a) shows the external shape of the model represented by the large-sparse point cloud; (b) is the surface area represented by the small-dense point cloud.*

Table 7 *The properties of the point clouds in Case II*

Point clouds	Point count	Point density /mm^2
Large-sparse	24,990	36
Small-dense	34,904	768

5.3.2. Pipeline parameters

The user-defined parameters of the pipeline for Case II are listed in Table 8 (to review the parameters of this pipeline, please see section 4.4).

Table 8 *User-defined parameters of the data fusion pipeline in Case II*

** The small-dense point cloud is rotated 10° for each attempt.*

Parameters		Values
W_i	W_1	0.05
	W_2	0.05
	W_3	0.1
	W_4	0.2
	W_5	0.6
s_x, s_y, s_z	s_x	1 mm
	s_y	1 mm
	s_z	0.1 mm
T_1	10	
$T_1{}'$	4	
$\delta_{large}, \delta_{small}$	δ_{large}	3.5
	δ_{small}	0.4
$\varepsilon_{large}, \varepsilon_{small}$	ε_{large}	1.18
	ε_{small}	0.82
T_2	10	
T_2'	2	
$\theta_{large}, \theta_{small}$	θ_{large}	1.0041215
	θ_{small}	0.998785
O	36 *	
n_x, n_y, n_z	n_x	10
	n_y	10
	n_z	5

5.3.3. Registration

With the parameters given in section 5.3.2, the algorithm subdivided the large-sparse point cloud into 2,164 sub-clouds, six of which were detected as the target sub-

clouds for registration. These detected sub-clouds form two continuous areas in the large-sparse point cloud: three of them form the area containing the designated target area whereas the other three fall into the irrelevant space as erroneous anomalies. These two areas are shown in Figure 27 (1 and 2).

The general visualisation of the registration can be seen in Figure 28 (white point cloud onto the surface the 3D model) and the colourmap of the deviations after registration is displayed in Figure 29. The histogram of the signed distances at each point in the small-dense point cloud is displayed in Figure 30.

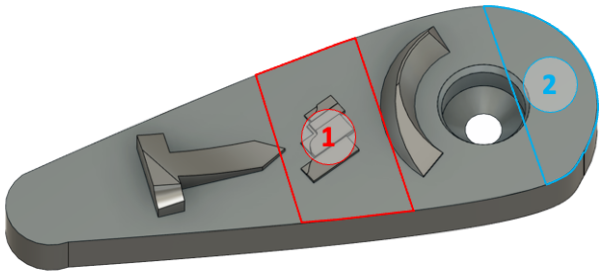


Figure 27 *The algorithm detected six potentially correct (matching) sub-clouds in Case II, forming two continuous areas. Three of them form area 1 (red), which contains the correct registration area. The other three form area 2 (blue), which is an anomaly and should be ruled out via manual selection.*

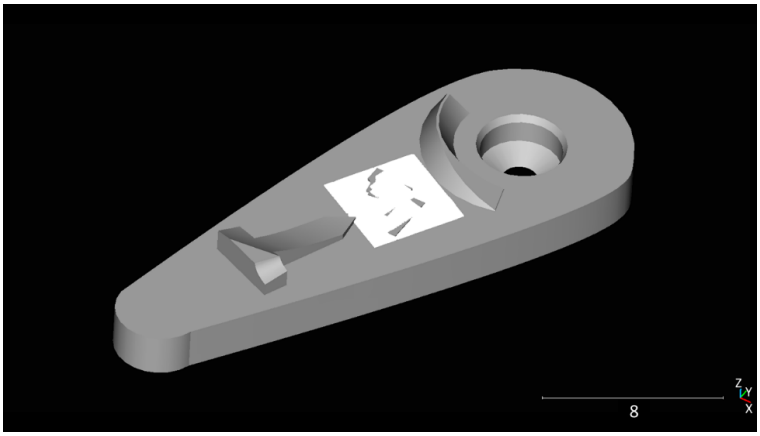


Figure 28 *The registration result of Case II. The white, rectangular object is the registered small-dense point cloud; due to its high point density, its geometrical features cannot be seen in this image.*

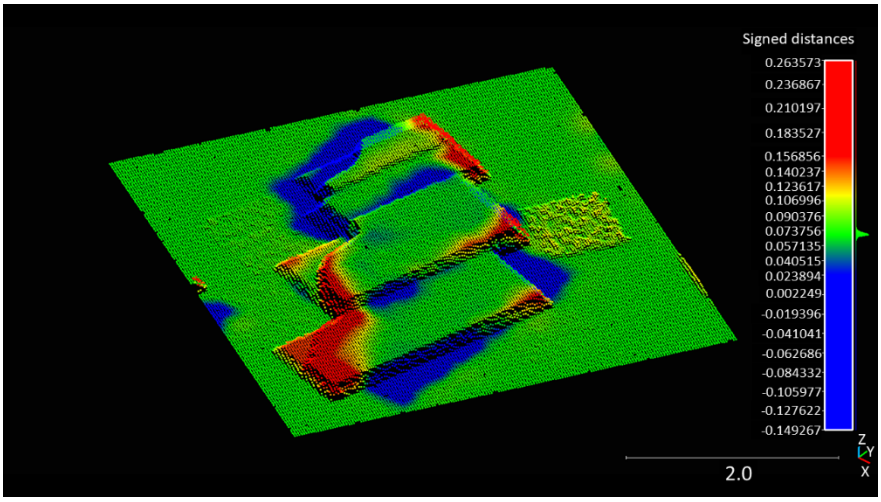


Figure 29 *The colourmap of the deviations from the 3D model (i.e. signed distances between the points in the small-dense point cloud and the surface of the 3D model). The unit is mm.*

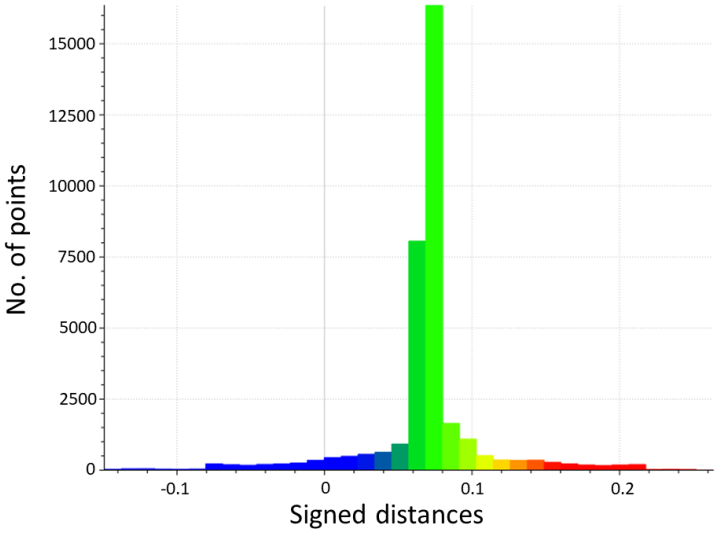


Figure 30 *Histogram of the signed distances in Case II. Mean value: 0.07 mm; standard deviation: 0.04 mm.*

5.4. Case III: a coin

As Case III, real measurements of a 20 pence coin were obtained. As mentioned in section 5.1.3, the large-sparse point cloud of the coin and the small-dense point cloud of a certain area on the coin were collected by two different instruments. As such, it is worth pointing out that the precise location of the surface texture measurement across the entire shape of the part is hard to define; it is possible to estimate an approximate location of the undertaken measurements based on the operator's observation, as it is normally done in routine practice. The same strategy mentioned previously is adopted for the proposed pipeline, however another limitation in real case scenarios is added as the large quantity of sub-clouds created by the proposed algorithm (the number of sub-clouds subdivided in the large-sparse point cloud will be introduced in section 5.4.3). It is impossible to plot all sub-clouds in an acceptable time window to check which ones among them are indeed located in the target registration area but missed by the geometrical comparison algorithm. A manual examination of all the outputs generated is beyond human capacity. As such, once receiving the outputs from the pipeline, I only manually examined the candidate sub-clouds selected by the pipeline and determine which ones were correctly located in the target area in the large-sparse point cloud.

The method to examine the generated outputs in this case is as follows: after receiving the candidates selected by the pipeline, the location of each selected sub-cloud was manually checked, picking only the ones that fall into the target area (as subjectively observed). The continuous area in the large-sparse point cloud which was formed by the manually picked sub-clouds was defined as the target area for the registration.

Before introducing the technical information about Case III, it is worth noting that, due to the default configurations of the optical instruments used in this experiment (for the details of these instruments, please review section 5.1.3), the unit of length is micrometre (μm).

5.4.1. Geometrical features

The coin selected in this case study is shown in Figure 31: the location of the surface texture measurement (red square) on the coin is an estimate because the instrument does not provide the coordinate system relative to the whole coin. The small-dense point cloud was processed with Mountains® (right-hand side in Figure 31) [111]. In this case study, only the tail side and the edges of the coin were scanned to generate the large-sparse point cloud. The properties of the point clouds collected from the coin are in Table 9.

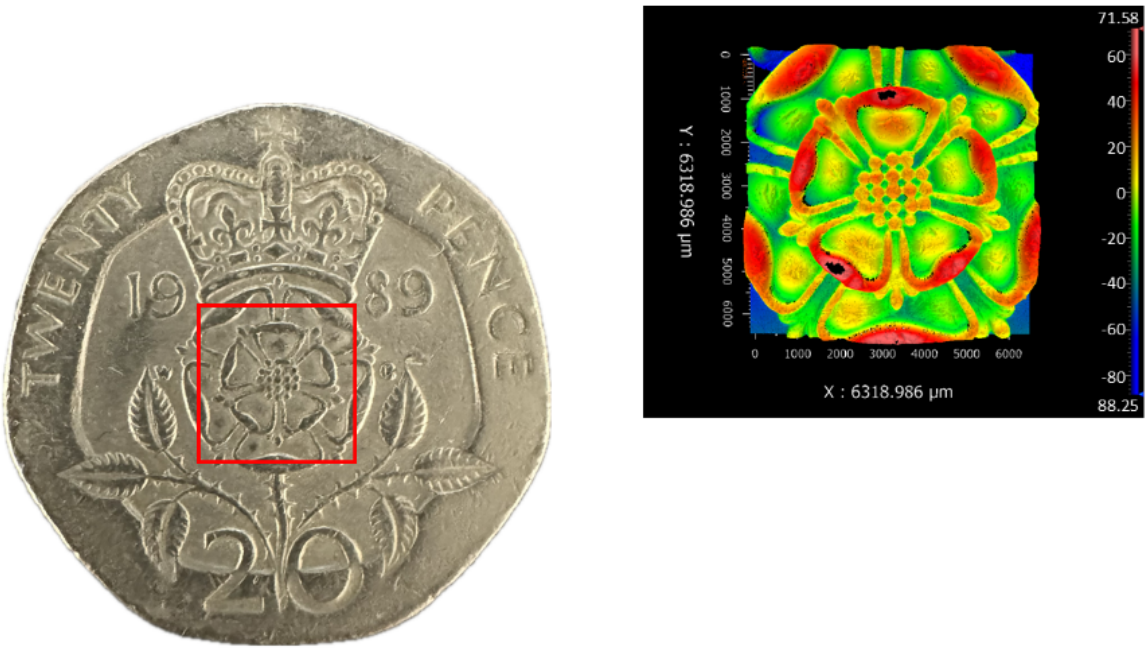


Figure 31 *The image of the 20 pence coin in Case III. The small-dense surface texture point cloud is collected from the area enclosed by the red square, which is shown in the image on the right-hand side. The colour scale in the image of the small-dense point cloud is in µm.*

Table 9 *The properties of the point clouds in Case III*

** The scanned areas of both point clouds are reconstructed by CloudCompare. Hence, the point densities are calculated according to the reconstructed areas.*

Point clouds	Point count	Point density* /mm²
Large-sparse	210,303	502
Small-dense	1,000,000	23,159

5.4.2. Pipeline parameters

The user-defined parameters of the pipeline for Case III are listed in Table 10 (to review the parameters of this pipeline, please see section 4.4).

Table 10 *User-defined parameters of the data fusion pipeline in Case III*

** 1 μm = 10^{-3} mm*

*** The small-dense point cloud is rotated 10° for each attempt.*

Parameters		Values
W_i	W_1	0.05
	W_2	0.05
	W_3	0.1
	W_4	0.2
	W_5	0.6
s_x, s_y, s_z	s_x	1000 μm*
	s_y	1000 μm
	s_z	10 μm
T_1	10	
$T_1{}'$	3	
$\delta_{large}, \delta_{small}$	δ_{large}	1.048
	δ_{small}	0.952
$\varepsilon_{large}, \varepsilon_{small}$	ε_{large}	1.3
	ε_{small}	0.7
T_2	10	
T_2'	2	
$\theta_{large}, \theta_{small}$	θ_{large}	1.0060898
	θ_{small}	0.9939102
O	36**	
$n_x,\ n_y,\ n_z$	n_x	5
	n_y	5
	n_z	5

5.4.3. Registration

With the parameters given in section 5.4.2, the algorithm subdivided the large-sparse point cloud into 25,459 sub-clouds, 186 of which were detected by the pipeline as the target sub-clouds for registration. After manual assessment, 22 out of these 186 candidates were verified as the correct ones locating in the target area. The 22 correctly detected sub-clouds have their major parts fallen into the estimated target area observed by the CSI instrument, which can be visualised in Figure 32. The results of the registration (in the form of signed distances) is displayed in Figure 33 in the length unit of μm. The histogram of the deviations is shown in Figure 34.

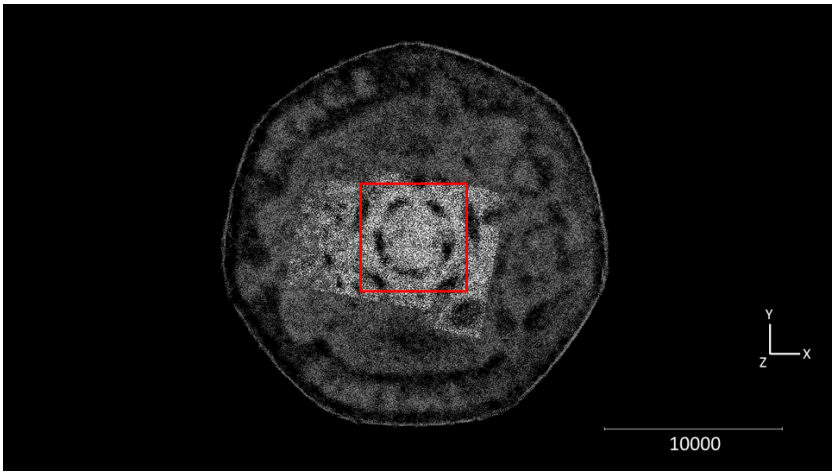


Figure 32 *The sub-clouds covering the target area in the large-sparse point cloud. The grey points form the large-sparse point cloud representing the shape of the coin. The 22 verified sub-clouds are coloured in white. The estimated target area is enclosed in the red square. The unit is μm.*

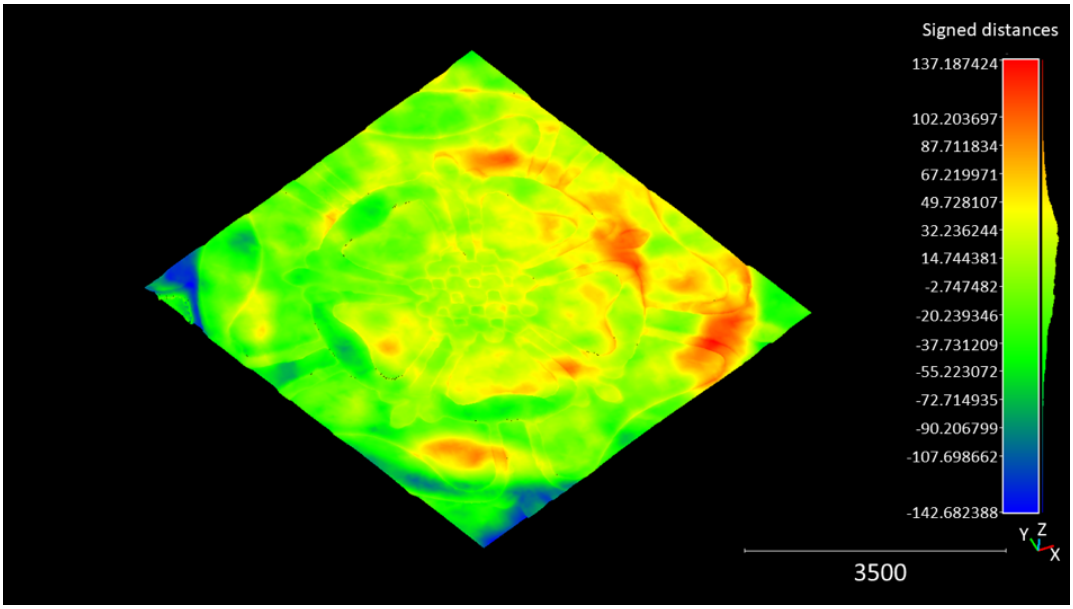


Figure 33 *The colourmap of the deviations (i.e. signed distances between the points of the surface texture measurement and the point cloud captured with fringe projection). The unit is μm.*

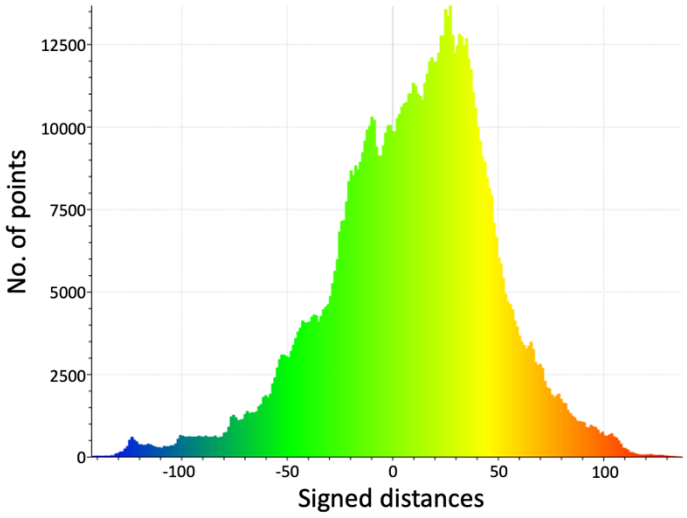


Figure 34 *Histogram of the signed distances in Case III. Mean value: 9.37 μm; standard deviation: 38.61 μm.*

5.5. Summary

In this chapter, the output of each registration attempt was presented in numerical and visual manners. The data fusion pipeline visioned in Chapter 3 and Chapter 4 was challenged by three pairs of 3D point clouds at different levels of geometrical complexity. The outputs of registration shown in sections 5.2.3, 5.3.3 and 5.4.3 demonstrated that 1) the algorithm introduced in sections 3.1 and 4.2, which is based on PCA, is adequate in detecting the sub-clouds which potentially form the target area to register the small-dense point cloud, and 2) the algorithm introduced in sections 3.2 and 4.3, which assesses the spatial distribution of points in a point cloud with voxelisation, is effective in detecting the most optimal orientation of the small-dense point cloud. As such, the three experiments demonstrated in this chapter have exhibited the functionality and potential of this data fusion pipeline.

In the next chapter, these outputs together with the configurations of the pipeline will be discussed.

Chapter 6

Discussion

As presented in Chapter 5, the pipeline was challenged by three pairs of point clouds with a diversity of geometrical complexity and outputted a diversity of results. Now, it is time to numerically analyse the registrations and discuss the results. To provide a clear overview of the registrations, the numerical information is summarised in Table 11. To have an initial perception of the registration accuracy, a new parameter "mean distance/side length" is listed in this table. A declining ratio between mean signed distance versus the size of the small-dense point cloud shows an increase in registration accuracy, though the point density of the point clouds and the sizes of the geometrical features are also critical factors.

Table 11 *Statistical summary of the experimental study in Chapter 5*

** For each study case, the first row refers to the large-sparse point cloud, the second row refers to the small-dense point cloud. Same for the "Point density" column.*

*** The small-dense point cloud is always a square shape on the x-y plane. This is the length of its sides on the x-y plane.*

**** The length originally collected by the instrument is 6318.986 μm.*

Case	Point count*	Point density /mm²	Mean distance mm	Standard deviation mm	Side length** mm	Mean distance / side length
I	49,992	44	0.15	0.090	3.0	0.05
	99,677	4,374				
II	24,990	36	0.070	0.040	4.5	0.016
	34,904	768				
III	210,303	502	0.0094	0.039	6.3***	0.0015
	1,000,000	23,159				

6.1. Requirements on the object and point clouds

The designated working volume for this pipeline is a cubic space dimensioned 500 × 500 × 500 mm. Within this space, there is no specific requirement on the size of the object and corresponding point clouds. There is also no specific numerical requirement on the characteristics of the point clouds collected from this object, but two principles must be observed:

1) The point density of the large-sparse point cloud should be high enough to show its fundamental geometrical features on the surface of the object, though not on the level of detail like the small-dense point cloud shows;
2) The difference in point density between the two point clouds should be in a range that the average point-to-point distance in the large-sparse point cloud is smaller than the *x-y* dimensions of the small-dense point cloud. As Figure 35 demonstrates, the large point-to-point distances in the large-sparse point cloud let few points (the blue dots) fall into the space covered by the small-dense point cloud (turquoise square). As such, it will be impossible to measure the geometrical similarity between this area in the large-sparse point cloud and the small-dense point cloud with the algorithm explained in sections 3.1 and 4.2.

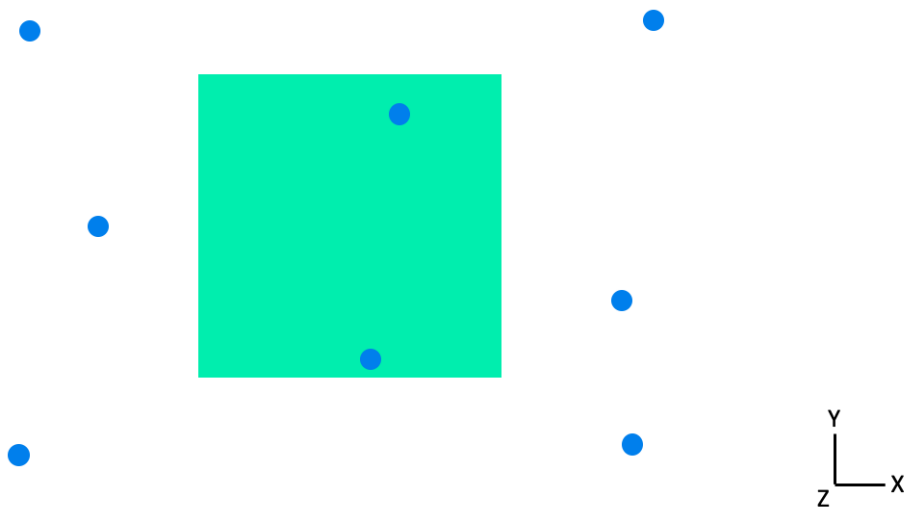


Figure 35 *A visual demonstration of a flawed input. The blue dots are the points in the large-sparse point cloud; the turquoise square represents the x-y dimensions of the small-dense point cloud. In this demonstration, there are two few (only two) points in the large-sparse point cloud falling in the space covered by the small-dense point cloud. The unreasonably large point-to-point distances in the large-sparse point cloud makes the geometrical similarity comparison impossible to proceed.*

6.2. Plane alignment in manual pre-processing

As stated in section 4.1, it is assumed that the user remembers from which surface of the engineered object the small-dense point cloud is captured; as such, before feeding the point cloud datasets into the pipeline, the user has to coarsely align the small-dense point cloud and the corresponding surface of the engineered object, which is formed of the points in the large-sparse point clouds, along *z*-axis. Unlike a CAD model, there is no "solid" surface in a point cloud. As such, the alignment of the small-dense point cloud and the corresponding surface on the measured object relies on the subjective observation of the user. This step naturally leads to uncertainty: there are complicated geometrical features on the small-dense point cloud, including but not limited to artificial designs and scratches due to long-term use. As such, finding a standardised or analytical method to align the small-dense point cloud with the corresponding "surface" on the measured object and a standard to measure how "well" the small-dense point cloud is aligned becomes unfeasible. At the moment, the only method to align the small-dense point cloud and the corresponding "surface" on the large-sparse point cloud is to trust the subjective judgement of the user.

6.3. Configuration of parameters

It is worth noting that the author of this work, at the moment, has not managed to propose an analytical method to determine the value of each parameter. The configuration of the tolerance parameters ($\delta_{large}, \delta_{small}, \varepsilon_{large}$ and ε_{small}) relied on manual adjustment: an arbitrary value was given to each parameter, then I observed the

output; if there were falsely detected sub-clouds in the output, the values of parameters would be adjusted in small increments until the lowest number of redundantly detected sub-clouds was observed. For different cases, parameters, especially the tolerance parameters, need different values.

6.4. Accuracy vs complexity

Regarding the three cases introduced in sections 5.2, 5.3 and 5.4, a trend in the detection of the correct sub-clouds can be noted: the performance of the proposed pipeline is affected as the geometrical complexity of the point cloud increases. This trend is summarised in Table 12. To measure the accuracy of target area detection, the rate of correct detection, which is equal to the number of correctly detected sub-clouds divided by the total number of sub-clouds detected by the pipeline, is proposed. From Case I to III, as the geometrical complexity increases, there are also increasingly more redundant sub-clouds, i.e. the sub-clouds detected by the pipeline as falling in the target area in the large-sparse point cloud but actually not, detected by the pipeline. As such, it is necessary for the user to assess the automatic outputs of the sub-cloud selection algorithm in this pipeline before proceeding into the registration stage, as not all sub-clouds selected by the pipeline are correct, i.e. falling into the target area in the large-sparse point cloud.

Table 12 *A numerical demonstration of the accuracy of target area detection in all study cases*

** Rate of correct detection = (Number of sub-clouds correctly selected by the pipeline) / (Total number of sub-clouds selected by the pipeline)*

Case	Total number of sub-clouds selected by the pipeline	Number of sub-clouds correctly selected by the pipeline	Rate of correct detection*
I	2	2	100%
II	6	3	50%
III	186	22	12%

6.5. Computational cost

The computational cost of a successful run of this pipeline depends on the quantity of points in the small-dense point clouds (in our cases, it ranges from 3.5×10^4 to 1.0×10^6), the number of sub-clouds subdivided in the large-sparse point cloud (in our cases, it ranges from 10^3 to 10^5), the number of orientations of the small-dense point cloud to assess (in our cases, there are 36 orientations to assess) and the number of voxels generated for the orientation detection (in our cases, the number of voxels for each orientation to compare is between 10^2 to 10^3). Additionally, the geometrical complexities of the point clouds also influence the time of running.

There is the art in balancing accuracy and efficiency: the more sub-clouds that are subdivided from the large-sparse point cloud (i.e. the smaller the step between two

neighbouring sub-clouds is), the higher the accuracy of target area detection can be; the more orientations of the small-dense point cloud that are tried (i.e. the smaller the rotational angle between two neighbouring orientations), the more accurate that the ultimate registration will be. However, this will lead to more iterations of calculations and hence higher computational cost.

6.6. Noise

Due to the functional characteristics of the instruments used in this research, there is no conspicuous noise in the point clouds. As such, the user is required to feed point clouds without conspicuous noise points into the pipeline. However, it should be noticed that the large-sparse point cloud of the coin, as per investigated in section 5.4, shows a mild level of noise on the "basin areas" between artistic patterns. This was caused by the small size of the coin compared to the designated working volume of ATOS GOM, and the light reflection on smooth areas. With a low level of noise as such, this pipeline exhibited the robustness in recognising the geometrically similar area.

6.7. Summary

In this chapter, the performances of the pipeline in the cases detailed in Chapter 5 were discussed. The requirements of the input were stated and explained in sections 6.1 and 6.2. The limitations and difficulties reflected by experimental cases were noted in sections 6.3, 6.4 and 6.5. In section 6.6, the question about potential noise points was specifically discussed: the instruments used in this research did not generate

conspicuous noise points, and hence the future user should expect to input their point clouds without significant noise.

The discussion in this chapter can answer the question about the functionality, reliability and robustness of this pipeline. In summary, that is: although further improvements are waiting, this pipeline has displayed its capability of detecting certain geometrical patterns in a geometrically complex point cloud, and register two point clouds in different sizes, spatial coverages, point densities and initial orientations.

In the next chapter, the research outcomes will be comprehensively summarised and concluded. In addition, two big-picture questions about general methodology, which were frequently raised by scholarly who had paid attention to this research project, will be discussed and concluded.

Chapter 7
Conclusions and future work

The progresses, diversification and expansion of the research on data fusion have been witnessed in both academia and industrial applications in the past decades. The scope of its applications is still expanding as this thesis was penned down. By combining sensor-collected data reflecting different physical properties or observation angles of an object, researchers and users can acquire comprehensive information about this object. The power of data fusion defines it a field with immense potentials for further exploration, including its applications in metrology. In this chapter, the questions raised in section 1.3 will have solid answers.

7.1. The review of existing research

In the past decade, researchers in different academic paths have proposed numerous algorithms to explore how data fusion can be effectively implemented in tasks of optical coordinate measurement. These algorithms can be categorised into three families observing their mathematical foundations: GP, WLS and machine learning; the first two can be further categorised as user-dependent algorithms and the latter is user-

independent. Among these three grand categories, GP is the mathematical basis of the most prevalent algorithms in recent years. Its popularity is mainly based on its mature mathematics and convenience of implementation. Compared to GP algorithms, WLS algorithms have not been frequently implemented in data fusion in the context of optical coordinate measurement due to its limitations in geometrical complexity. Nonetheless, as user-dependent algorithms, they have similar drawbacks, particularly their reliance upon pre-programmed models to acquire the geometrical patterns in given datasets. This makes them impotent when being applied to complex geometries which are beyond the scope of the pre-programmed model, such as sharp edges and irregular surface textures.

Machine learning, as a trendy technique under the umbrella of artificial intelligence, has shown enormous functionality in a vast range of scenarios, though its potential in optical coordinate metrology is still at a fermentation stage. As a type of user-independent algorithm, machine learning can learn the patterns in training datasets and then detect similar patterns in a newly given dataset. However, a machine learning algorithm is unable to detect a desired pattern in a new input if this pattern is not included in its training datasets, i.e. it cannot recognise a new pattern that it has not trained for.

Now, back to the designated scenarios of this research project. The purpose of this research is to propose a data fusion algorithmic pipeline which can register a small-dense point cloud into the right position in a large-sparse point cloud with a correct orientation. The two point clouds waiting be to be registered are 1) of different sizes, 2) of different point densities and 3) in two separate and irrelevant coordinate systems (i.e. measured with two individual instruments). A specific example of the large-sparse point cloud is

the global point cloud showing the external coordinate of an engineered object; correspondingly, the small-dense point cloud represents the surface texture details of a small area on the surface of that engineered object. This scenario is obviously beyond the capability of GP, WLS and machine learning techniques. A new algorithmic pipeline was consequently proposed.

7.2. Two coordinate frames; no training data

The pipeline must face three challenges: 1) determine the correct location to register the small-dense point cloud, 2) determine the correct orientation of the small point cloud, and 3) there is no training data, i.e. existing correct answers, to train a machine-learning neural network for patter recognition in a dataset.

The methods proposed by this thesis to surmount these two challenges is geometrical similarity comparison, a theory based on statistics. The large-sparse point cloud is subdivided into equally sized sub-clouds; the geometrical similarity between each sub-cloud and the small-dense point cloud is assessed by PCA, including the point-to-PC-plane distances and the local-norm-to-PC-plane angles. The sub-clouds which exhibit the most similar PCA to the small-dense point cloud and form a continuous area in the large-sparse point cloud are determined as the target area for registration. To detect the correct orientation to align the small-dense point cloud with the target area, the spatial point distributions of both are examined by counting the percentages of points falling into each voxel; the orientation which gives the small-dense point cloud the most similar spatial point distribution is determined as the correct orientation. In summary,

the theory on which this pipeline is constructed is focused on the comparison of the spatial point distributions of two point clouds.

7.3. Performances with various geometrical complexities

To prove the functionality of this pipeline and the theory of geometrical similarity comparison, three pairs of point clouds in various geometrical characteristics and complexities were designed: the first two cases were synthetic point clouds generated from 3D CAD models; the last one consisted of two point clouds collected from a 20-pence coin by real optical instruments. The synthetic point clouds, also their CAD prototypes, had relatively regular and simple geometrical features, with synthetic surface texture patches covering their surfaces. The coin, as a real engineered object, had irregular surface textures such as scratches and complex aesthetical designs such as flowers. The successes and drawbacks hinted by the three study cases are summarised as follows:

1) The pipeline is capable of detecting the target area for registering the small-dense point cloud by examining each sub-area of the large-sparse point cloud, though manual reassessment is necessary when there is a considerable geometrical complexity in both point clouds. As shown in section 6.4, the mistakenly detected sub-clouds increased along with the increasing geometrical complexity; the rate of correct detection decreased from 100% to 12%. In other words, in industrial practices, the user is required to handpick which areas detected by the pipeline are correctly the target area. However, the

pipeline did not miss the sub-clouds which were genuinely located inside the target area;

2) With signed distances as the standard of measurement, the pipeline can register the small point cloud with an acceptable level of accuracy in terms of the registered orientation (i.e. the values of mean distances / side length are remarkably small, varying from 0.0015 to 0.05). However, it should be noted that the value of mean distances / side length can decline because of the increase of point density. Additionally, a point cloud with small surface textures (the heights of the surface textures are ignorable relative to the overall size of the engineered object) can also lead to small mean distances / side length value;
3) Given the steps and mechanism in this pipeline, the requirements on the computing device can be demanding. The philosophy of this pipeline is a brute-force methodology, i.e. examining every single sub-section of a large-sparse point cloud and detect the ones that are matched to the small-dense point cloud; then attempting every single orientation of the small-dense point cloud until a correct one is found. The computational cost will be dramatically massive when the quantity of points and the number of sub-clouds is huge. As such, at this early stage, this pipeline is not applicable to any ad-hoc missions. On the contrary, it is a tool suitable for post-measurement data processing.

In conclusion, the proposed pipeline has potential for further improvements and a wider scope of application scenarios.

7.4. Contributions to science

The concepts, theories and experimental outcomes have supplied the following novelties to the science community:

The taxonomy of the existing data fusion algorithms in the context of metrology and advanced manufacturing, which was proposed based on mathematical principles, has provided a structured, rational guideline for researchers and engineers to design and choose algorithms for their application scenarios. In summary, for measured surfaces with geometrical characteristics which can be analytically modelled with Gaussian distribution or weighted least square method, one should choose user-dependent algorithms for their adequacy, maturity and efficiency; for the surfaces with geometrical characteristics which cannot be modelled with a mathematical formula, one should refer to user-independent methods such as machine learning, though training data showing similar geometrical patterns should be prepared.

1) The characteristics and challenges in a novel point cloud registration scenario, where a small-dense point cloud represents a detailed view of a certain part of a large-dense point cloud and there is a point density gap larger than 10 times, have been investigated in this research. By the time this thesis was completed, to the latest knowledge of the author, this scenario has not been investigated by researchers in the science community. The motivation to explore this novel scenario was induced by its potential application contexts, which will be stated in section 7.5.

2) A novel pattern recognition for 3D dataset has been proposed by this research, proving a unique view that pattern recognition without training data (i.e. prepared correct answers) is possible. As mentioned in Chapter 1, this is due to the specific application settings of this research, where a large amount of correctly registered point cloud pairs is impossible to obtain. To recognise a user-defined 3D geometrical pattern in a massive 3D dataset, the pipeline has to store the user-defined pattern, which is contained in the small-dense point cloud, in a simple, effective format based on the statistical characteristics in this user-defined pattern. My solution is PCA: the geometrical characteristics in a point cloud is simplified as histograms showing the point-to-PC-plane distances and the local-norm-to-PC-plane angles. The comparison of geometrical similarity is reduced to the comparison of 2D histograms, which makes large amount of training data unnecessary.
3) An orientation alignment algorithm to align two 3D point clouds with considerably different point densities. As such, this algorithm is point-density-difference-proof. By analysing the 3D point distribution of the two point clouds, based on the technique of space voxelisation (section 3.2 and section 4.3), the algorithm can detect the specific orientation of the small-dense point cloud which enables it to have the most similar spatial point distribution to the target area in the large-sparse point cloud. As the number of points falling into each voxel is converted to a percentage, which is calculated relative to the total number of points in the corresponding point cloud, this method is robust against the difference of point density between the two point clouds.

7.5. Application scenarios

There are two potential application scenarios of this data fusion pipeline in precision manufacturing industry. The first one is a “map of surface texture” of an engineered object, such as a metal component on a piece of aircraft or spacecraft. This pipeline is capable of registering multiple small-dense point clouds, covering the surface textures in different areas, onto a single large-sparse point cloud. The output of multiple inputs will be a surface map of an engineered object. It can be immensely informative to examine the quality of surface processing in precision manufacturing.

The second application scenario is the detection of imperfections and damages on the external surface of an engineered object. The flaws on the surface can be tiny; the user who recognises an imperfection on the surface using instruments such as CSI may not be able to recognise the location of the imperfection due to its minuscule size. With this data fusion pipeline, the user can find the location of the imperfection or damage on the surface.

Sections 7.1, 7.2, 7.3, 7.4 and 7.5 are aimed at answering the research questions defined in section 1.3. However, there are two big-picture questions that need ultimate, clear answers. These two big-picture questions are not enclosed to the scope of this research, but to the author’s knowledge, they have been frequently propounded to question the methodology of this pipeline. As such, it is the author’s duty to respond to these frequent enquiries.

7.6. The bigger picture I: the most accurate solution in theory and why it is practically unfeasible

The proposed method to determine the most appropriate location in the large-sparse point cloud of the small-dense cloud is an area-by-area brute-force searching approach, based on the geometrical similarities between datasets. As the experimental studies proposed in this work have shown, this method has potential, but its limitations increase with the complexity of geometrical features. In theory, the most accurate method to determine the location should be proceeded as follows:

Suppose the numbers of points in the large-sparse and small-sparse point clouds are denoted as N_L and N_s respectively. The points in each point cloud are denoted as set Σ_L and set Σ_s respectively. As such, I have $|\Sigma_L| = N_L$ and $|\Sigma_S| = N_S$. The areas covered by the large-sparse and small-sparse point clouds are denoted as A_L and A_S respectively. Now, I randomly select a group of points in the large-sparse point cloud, with each selection contains the points denoted with the set $\Gamma \in \Sigma_L$; the number of points contained in each selection is always the same, denoted as $|\Gamma| \equiv \frac{A_S N_L}{A_L}$. Between any two random selections, there must be at least one point in each selection does not belong to their intersection, denoted as $\left|\Gamma_i \vartriangle \Gamma_j\right| \geq 2$. All selections are then stored in a set denoted as

$$\mathbb{O} = \{\Gamma_1, \Gamma_2, \Gamma_3, \cdots, \Gamma_t\}, t = C_{N_L}^{\frac{A_S N_L}{A_L}} \tag{13}$$

In other words, the total number of possible unique selection is $C_{N_L}^{\frac{A_S N_L}{A_L}}$, or $|\mathbb{O}| = C_{N_L}^{\frac{A_S N_L}{A_L}}$.

Then, there is a function G for storing the 3D geometrical pattern of point distribution of a point cloud, which can be a neural network in engineering scenarios. For instance, the distribution in the 3D space of the small-dense point cloud is denoted by $G(\Sigma_S)$. The similarity of the point distribution in the 3D space between two point clouds is measured by a function $\mathfrak{D}(G_i, G_j)$. Among all selections in the set $\mathbb{O}$, there must be exactly one selection, denoted as Γ_k, which satisfies the following condition:

$$\exists!\, \Gamma_k \in \mathbb{O}\ \ \mathfrak{D}\big(G(\Sigma_S), G(\Gamma_k)\big) > \mathfrak{D}\big(G(\Sigma_S), G(\Gamma_i)\big), \Gamma_i \in \complement_{\mathbb{O}}\Gamma_k \tag{14}$$

Ideally, the points in Γ_k are concentrated in the target area for registration, where the geometrical pattern of the point distribution is exactly the same as that of the small-dense point cloud. As such, the next step is to determine the geometrical centroid of Γ_k and locate the small-dense point cloud accordingly. In theory, the algorithmic pipeline described above should have the best accuracy of location determination and final registration.

However, I must not celebrate the victory of mathematics too early as the feasibility of this theory is restricted by realistic technical limitations. To give a concrete comprehension of the computational load and complexity of this ideal algorithmic pipeline, I will proceed some basic arithmetic calculations with the data in the experimental study Case I, the geometrically simplest case in this research. If I register the two point clouds in Case I via this theoretical pipeline, given the values of the parameters mentioned in section 5.2.1, I should have 925 points in each selection and the total number of selections will be $C_{49992}^{925} \approx 8.06 \times 10^{1998}$, i.e. there will be 8.06×10^{1998} iterations of comparisons and calculations. There is no necessity to proceed into further calculations to understand the computational cost of this theoretical

pipeline, because I should note that there are only approximately 10^{80} atoms in the whole universe (known as Eddington Number) [112,113]. As such, the area-by-area scanning method, though may lead to erroneous detection of the target area and is ostensibly awkward in theory, is an effective and, most importantly, practicable method for the point cloud registration tasks of the type described in our experimental study cases.

For the detection of the point cloud orientation, the most established and widely used method for registration is the ICP algorithm. However, ICP and its variants are more successful when used to register two point clouds with similar point density and equal shape [105,114]. In the context of the presented work, there is a relatively large disparity of point densities between the two point clouds. For instance, in the three experimental study cases, the differences in point densities between the inputted point clouds range from a magnitude of 1 to 2, i.e. 10^1 to 10^2.

7.7. The bigger picture II: why machine learning is unapplicable

Machine learning and its variants have exhibited their faces ubiquitously in object detection and pattern recognition. As such, when tackling problems such as recognising a geometrical pattern in a point cloud, training a machine learning programme with abundant training data will be the first tactic occurring to researchers' minds. However, researchers are frequently forgetting a fact: the learning ability of a machine learning programme is based on "training"; the neural network must be "fed" with a large quantity of labelled data (i.e. correct answers) before it is enabled to recognise a certain pattern

in a new input. This nature of machine learning makes it an unfeasible method for target area detection in our pipeline. The reasons are explained as follows:

Firstly, it is impossible to get a large quantity of "correct answers" in our designated scenarios to train a machine-learning neural network. Researchers using CSI devices are unable to know where exactly and precisely on the observed part the surface texture point cloud (small-dense point cloud) is collected, though they can roughly recognise the area from which the small-dense point cloud is collected. This is exactly the situation of Case III: the red square in Figure 31 marks the estimated area where the CSI looked at, but it is impossible for us to reassure that this area is precisely where the CSI collected the surface texture data; because, as I have to repeat here, the CSI system employed to collect surface texture data and the device employed to collect the coordinate point cloud are unrelated.

Secondly, machine learning can recognise a category of objects (objects sharing a same range of properties or creatures belonging to a single species) from an assorted dataset. Contrastingly different from this application scenario, in our case, the algorithmic pipeline has to recognise the area in the large-sparse point cloud which is exactly the same or similar to the small-dense point cloud. In more colloquial words, this pipeline is to determine the exact one object given only one dataset about this object, instead of recognising a category of objects after being trained with a large quantity of data describing the mutual properties of the objects this category.

As such, given the reality that there is no (and it is impossible to have) a large amount of "correctly registered point clouds" as training datasets, machine learning is not an option for our application context.

7.8. Future work

Future tasks awaiting further exploration and investigation can be categorised into theoretical and industrial sides. For theorists, the following tasks should be delved in:

1) There should be a systematic and analytical method to determine the parameters of this pipeline, particularly the tolerance parameters, for each individual registration task.
2) More statistical methods for comparing the geometrical similarity between the small-dense point cloud and the sub-clouds of the large-sparse point clouds can be explored. As the study cases have reflected, measuring the geometrical similarity between two point clouds based on PCA will only lead to false outputs when there is a high level of geometrical complexity.
3) The working volume of this pipeline can be expanded into a space larger than $500 \times 500 \times 500$ mm. This volume expansion will be particularly critical for large-scale machinery productions, such as the manufacturing aerofoils and fuselages. Future researchers are advised to investigate unforeseeable challenges in this pipeline after its working volume has been massively extended.
4) As mentioned in section 6.6, the instruments used in this research provided "clean" point clouds as the input. However, facilities in other contexts might not provide point clouds free from noise. As such, future researchers are advised to propose new algorithms to detect the "true" geometries of a point cloud surmounting the interference from noise.

5) Future researchers are encouraged to rewire the algorithmic structures of this pipeline to optimise the computational cost.

For professionals in industrial circumstances, the following application scenarios deserve more attention:

1) This pipeline is potentially capable of improving the level of intelligence in industrial robots; an industrial robot installed with this pipeline can recognise a user-input object without referring to a large amount of training data. This application requires the microcomputer in the robot to proceed this pipeline at a fast speed in order to achieve a live motion. Additionally, the pipeline in such a robot should be re-programmed to work parallel with a proper computer vision system.
2) For the assessment of the quality of the engineered surface, this pipeline can also help to determine the location of the flawed area, which is usually observed via microscopic observation and hence unable to precisely be located by natural eyes. In this context, this pipeline can help the user to locate the defects whose details are collected by a microscopic observation instrument. The locations of defects can convey rich information about the problems in manufacturing processes. However, to make this pipeline function impeccably in this scenario, the precision and accuracy of this pipeline should be further improved.
3) For geoscientists, this pipeline can be employed for terrestrial observation. For instance, it can locate a certain type of landscape geometries, particularly the one which are formed after natural disasters. This application is important for geomorphological research.

Bibliography

[1] Castanedo F 2013 A review of data fusion techniques *Sci. World J.* **2013** 704504

[2] Leach R 2020 *Advances in Optical Form and Coordinate Metrology* (IOP Publishing)

[3] Diez-Olivan A, Del Ser J, Galar D, Sierra B 2019 Data fusion and machine learning for industrial prognosis: Trends and perspectives towards Industry 4.0 *Inform. Fusion* **50** 92–111

[4] Liu J, Li T, Xie P, Du S, Teng F, Yang X 2020 Urban big data fusion based on deep learning: An overview *Inform. Fusion* **53** 123–33

[5] Dautov R, Distefano S, Buyya R 2019 Hierarchical data fusion for Smart Healthcare *J. Big Data* **6** 19

[6] Zahra S R 2021 Securing the Defense Data for Making Better Decisions Using Data Fusion. In: T J Saleem, M A Chishti *Big Data Analytics for Internet of Things* (Wiley) 321–31

[7] White F E 1987 *Data Fusion Lexicon* (The Data Fusion Subpanel of the Joint Directors of Laboratories, Technical Panel for C3: San Diego)

[8] White F E 1991 *Data Fusion Lexicon* (The Data Fusion Subpanel of the Joint Directors of Laboratories, Technical Panel for C3: San Diego)

[9] Leach R K 2014 *Fundamental principles of engineering nanometrology* (Elsevier: Amsterdam)

[10] Leach R 2020 *Advances in Optical Surface Texture Metrology* (IOP Publishing)

[11] Saha S, Foryś P, Martusewicz J, Sitnik R 2020 Approach to analysis the surface geometry change in cultural heritage objects *ICISP 2020*, *Marrakesh, Morocco* 3–13

[12] Weckenmann A, Jiang X, Sommer K D, Neuschaefer-Rube U, Seewig J, Shaw L, Estler T 2009 Multisensor data fusion in dimensional metrology *CIRP Ann Manuf Technol* **58** 701–21

[13] Xu B J, Willomitzer F, Yeh C K, Li F, Gupta V, Tumblin J, Walton M, Cossairt O 2019 3D Surface Measurement and Analysis of Works of Art *Conf. Rec. Asilomar Conf. Signals. Syst. Comput.*, *Pacific Grove, USA* 1779–82

[14] Catalucci S, Senin N 2020 State-of-the-art in point cloud analysis. In: R K Leach *Advances in Optical Form and Coordinate Metrology* (IOP Publishing) 2-1-2–48

[15] ISO 10360-13:2021 *Geometrical product specifications (GPS) — Acceptance and reverification tests for coordinate measuring systems (CMS) — Part 13: Optical 3D CMS* 2–6

[16] Abdelazeem M, Elamin A, Afifi A, El-Rabbany A 2021 Multi-sensor point cloud data fusion for precise 3D mapping *Egypt. J. Remote Sens. Space Sci.* 835–44

[17] Catalucci S, Senin N 2020 State-of-the-art in point cloud analysis. In: R K Leach *Advances in Optical Form and Coordinate Metrology* (IOP Publishing) 2-1-2–48

[18] D'Errico G E 2012 À la Kalman filtering for metrology tool with application to coordinate measuring machines *IEEE Trans. Ind. Electron.* **59** 4377–82

[19] Amamra A, Aouf N, Stuart D, Richardson M 2016 A recursive robust filtering approach for 3D registration *Signal Image Video P.* **10** 835–42

[20] Sandhu R, Dambreville S, Tannenbaum A 2010 Point set registration via particle filtering and stochastic dynamics *IEEE Trans. Pattern Anal. Mach. Intell.* **32** 1459–73

[21] Chen Z, Li Q, Li J, Zhang D, Yu J, Yin Y, Lv S, Liang A 2022 IMU-Aided Registration of MLS Point Clouds Using Inertial Trajectory Error Model and Least Squares Optimization *Remote Sens. (Basel)* **14** 1365

[22] Luo R C, Kay M G 1989 Multisensor integration and fusion in intelligent machines and systems *IEEE Trans. Syst. Man. Cybern.* 901–31

[23] Dasarathy B V 1997 Sensor fusion potential exploitation-innovative architectures and illustrative applications *Proc. IEEE* **85** 24–38

[24] Luo R C, Yih C C, Su K L 2002 Multisensor fusion and integration: Approaches, applications, and future research directions *IEEE Sens. J.* **2** 107–19

[25] Eastwood J, Sims-Waterhouse D, Piano S 2020 Machine learning approaches. In: R K Leach *Advances in Optical Form and Coordinate Metrology* (IOP Publishing) 6-1-6–20

[26] Colosimo B M, Pacella M, Senin N 2015 Multisensor data fusion via Gaussian process models for dimensional and geometric verification *Precis Eng* **40** 199–213

[27] Hida T, Hitsuda M 1993 *Gaussian processes* (Providence, R.I., American Mathematical Society)

[28] Ren M J, Cheung C F, Xiao G B 2018 Gaussian process based bayesian inference system for intelligent surface measurement *Sensors (Switzerland)* **18** 4069

[29] Forbes A B 2012 Weighting observations from multi-sensor coordinate measuring systems *Meas. Sci. Technol.* **23** 025004

[30] Wang J, Pagani L, Leach R K, Zeng W, Colosimo B M, Zhou L 2017 Study of weighted fusion methods for the measurement of surface geometry *Precis. Eng.* **47** 111–21

[31] Eastwood J, Sims-Waterhouse D, Piano S 2020 Machine learning approaches. In: R K Leach *Advances in Optical Form and Coordinate Metrology* (IOP Publishing) 6-1-6–20

[32] Nilsson N J 1965 *Learning Machines: Foundations of Trainable Pattern-Classifying Systems* (McGraw-Hill Companies)

[33] Samuel A L 1959 Some studies in machine learning using the game of checkers *IBM J. Res. Dev.* **3** 210–29

[34] Alippi C, Ferrero A, Piuri V 1998 Artificial intelligence for instrument & measurement applications *IEEE Instrum. Meas. Mag.* **1** 9–17

[35] Halevy A, Norvig P, Pereira F 2009 The unreasonable effectiveness of data *IEEE Intell. Syst.* **24** 9–12

[36] Liu S, Zhang L, Yan Z 2018 Predict pairwise trust based on machine learning in online social networks: A survey *IEEE Access* **6** 51297–318

[37] Wei L, Luo W, Weng J, Zhong Y, Zhang X, Yan Z 2017 Machine learning-based malicious application detection of android *IEEE Access* **5** 25591–601

[38] Jing W, Kang J, Liu M 2018 Mining taxi trajectories for most suitable stations of sharing bikes to ease traffic congestion *IET Intell. Transp. Sys.* **12** 586–93

[39] Meng T, Jing X, Yan Z, Pedrycz W 2020 A survey on machine learning for data fusion *Inform. Fusion* **57** 115–29

[40] Lin K C, Lin C H and Lin V C 2009 A planar multiband antenna with parasitic-element design for multistandard mobile terminals *IEEE Antennas Propag. Soc. AP S. Int. Symp.* 1–4

[41] Julisch K 2003 Clustering intrusion detection alarms to support root cause analysis *ACM T. Inform. Syst. Se.* **6** 443–71

[42] Völker C, Shokouhi P 2015 Data aggregation for improved honeycomb detection in concrete using machine learning–based algorithms *Int. Symp. NDT-CE* (Berlin)

[43] Ji D, Liu Q, Bai M, Sun P 2020 A multisensor data fusion method based on gaussian process model for precision measurement of complex surfaces *Sensors (Switzerland)* **20** 278–93

[44] Ma P, Kang E L 2020 A Fused Gaussian Process Model for Very Large Spatial Data *J. Comput. Graph. Stat.* **29** 479–89

[45] Yin Y, Ren M J, Sun L 2017 Dependant Gaussian processes regression for intelligent sampling of freeform and structured surfaces *CIRP Ann. Manuf. Technol.* **66** 511–4

[46] Ren M J, Sun L J, Liu M Y, Cheung C F, Yin Y H, Cao Y L 2017 A weighted least square based data fusion method for precision measurement of freeform surfaces *Precis. Eng.* **48** 144–51

[47] Yu Z, Wang T, Wang P, Tian Y, Li H 2019 Rapid and precise reverse engineering of complex geometry through multi-sensor data fusion *IEEE Access* **7** 165793–813

[48] Xiang B, Li Y, Chen G, Liu X, Yang W 2020 Multi-source integrated fusion for surface measurement *Int. J. Adv. Manuf. Technol.* **109** 1815–23

[49] Zhou G, Li Y, Liu C, Hao X 2018 A posture adjustment optimization method of the laser inspection device for large complex surface parts *Proc. Inst. Mech. Eng. B J. Eng Manuf.* 2375–85

[50] Kong L B, Ren M J, Xu M 2017 Development of data registration and fusion methods for measurement of ultra-precision freeform surfaces *Sensors* **17** 01110

[51] Wang Y, Solomon J 2019 Deep closest point: Learning representations for point cloud registration *Proc. IEEE Int. Conf. Comput. Vis.* (Soeul, South Korea) 3523–32

[52] Besl P J, McKay N D 1992 A Method for Registration of 3-D Shapes *IEEE Trans. Pattern Anal. Mach. Intell.* **14** 239–56

[53] Segal A V, Haehnel D, Thrun S 2009 Generalized ICP In: J Trinkle, Y Matsuoka, J A Castellanos *Proc. Robot.: Sci. Syst.* (MIT Press: Seattle) 21

[54] Boyd S, Vandenberghe L 2004 Convex Functions. In: Boyd S, Vandenberghe L *Convex Optimization* (Cambridge University Press: New York) 67–113

[55] Fitzgibbon A W 2003 Robust registration of 2D and 3D point sets *Image Vis. Comput.* 1145–53

[56] Rusinkiewicz S, Levoy M 2001 Efficient variants of the ICP algorithm *Proc. Internat. Conf. 3DIM* (Quebec City, QC, Canada) 145–52

[57] Yan J, Yin X C, Lin W, Deng C, Zha H, Yang X 2016 A short survey of recent advances in graph matching *Proc. ACM ICMR, New York, USA* 167–74

[58] Zhang H, Zhang Q, Shao S, Niu T, Yang X 2020 Attention-based LSTM network for rotatory machine remaining useful life prediction *IEEE Access* **8** 132188–99

[59] Vinyals O, Fortunato M, Jaitly N 2015 Pointer networks *Proc. NIPS* (Montreal, Canada) 2692–2700

[60] Gojcic Z, Zhou C, Wegner J D, Guibas L J, Birdal T 2020 Learning Multiview 3D Point Cloud Registration *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.* (Seattle) 1756–66

[61] Zeng A, Song S, Nießner M, Fisher M, Xiao J, Funkhouser T 2017 3DMatch: Learning local geometric descriptors from RGB-D reconstructions *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. 2017, Honolulu, USA* 199–208

[62] Choi S, Zhou Q Y, Koltun V 2015 Robust reconstruction of indoor scenes *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern. Recognit., Boston, USA* 97–104

[63] Dai A, Chang A X, Savva M, Halber M, Funkhouser T, Nießner M 2017 ScanNet: Richly-annotated 3D reconstructions of indoor scenes *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. 2017, Honolulu, USA* 5828–39

[64] Zhang Z, Dai Y, Sun J 2020 Deep learning based point cloud registration: an overview *Virtual Real. Intell. Hardw.* **2** 222–46

[65] Chen Y, Peng C 2017 Intelligent adaptive sampling guided by Gaussian process inference *Meas. Sci. Technol.* **28** 105005

[66] Wang J, Jiang X, Blunt L A, Leach R K, Scott P J 2012 Intelligent sampling for the measurement of structured surfaces *Meas. Sci. Technol.* **23** 085006

[67] Rasmussen C E, Williams C K I 2008 *Gaussian processes for machine learning* (MIT Press: Cambridge)

[68] Park C, Huang J Z, Ding Y 2011 Domain decomposition approach for fast Gaussian process regression of large spatial data sets *J. Mach. Learn. Res.* **12** 1697–728

[69] Corder G W, Foreman D I 2014 Nonparametric Statistics: An Introduction. In: Corder G W, Foreman D I *Nonparametric Statistics for Non-Statisticians: A Step-by-Step Approach* (John Wiley & Sons, Inc: Hoboken, New Jersey) 1–10

[70] Song X, Jiang X, Gao J, Cai Z 2019 Gaussian process graph-based Discriminant Analysis for hyperspectral images classification *Remote Sens. (Basel)* **11** 2288

[71] Dumas A, Echard B, Gayton N, Rochat O, Dantan J Y, Van Der Veen S 2013 AK-ILS: An active learning method based on Kriging for the inspection of large surfaces *Precis. Eng.* **37** 1–9

[72] Richardson R R, Osborne M A, Howey D A 2017 Gaussian process regression for forecasting battery state of health *J. Power Sources* **357** 209–19

[73] Lázaro-Gredilla M, Titsias M K, Verrelst J, Camps-Valls G 2014 Retrieval of biophysical parameters with heteroscedastic Gaussian processes *IEEE Geosci. Remote S. Lett.* **11** 838–42

[74] Ghaffari Jadidi M, Valls Miro J, Dissanayake G 2018 Gaussian processes autonomous mapping and exploration for range-sensing mobile robots *Auton. Robots* **42** 273–90

[75] Kolanowski K, Świetlicka A, Kapela R, Pochmara J, Rybarczyk A 2018 Multisensor data fusion using Elman neural networks *Appl. Math. Comput.* **319** 236–44

[76] Fahmy M S, Atiya A F, Elfouly R S 2008 Biometric fusion using enhanced SVM classification *Proc. - 2008 4th IIH-MSP, Harbin, China* 1043–8

[77] Tong W G, Li B S, Jin X Z, Yang Y Q, Zhang Q 2006 A study on model of multisensor information fusion and its application *Proc. 2006 ICMLC, Dalian, China,* 3073–7

[78] Shu H, Wang Y, Jiang J 2007 Multi-rada data fusion algorithm based on K-central clustering *Proc. FSKD 2007, Haikou, China* 4406311

[79] Wang H, Liu T, Bu Q, Yang B 2016 An algorithm based on hierarchical clustering for multi-target tracking of multi-sensor data fusion *Chinese Control Conference, Chengdu, China* 5106–11

[80] Alyannezhadi M M, Pouyan A A, Abolghasemi V 2017 An efficient algorithm for multisensory data fusion under uncertainty condition *J. Electr. Syst. Inf. Technol.* **4** 269–78

[81] Zhou Q Y, Park J, Koltun V 2016 Fast global registration *ECCV 2016, Amsterdam, The Netherlands* 766–82

[82] Agamennoni G, Fontana S, Siegwart R Y, Sorrenti D G 2016 Point Clouds Registration with Probabilistic Data Association *IEEE Int. Conf. Intell. Robots Syst., Daejeon, South Korea* 4092–8

[83] Rasmussen C E, Williams C K I 2006 *Gaussian processes for machine learning* (MIT Press: Cambridge)

[84] Wang X, Qian X 2018 Gaussian process model for touch probing *ASME 2018 13th Internat. MSEC 2018, College Station, Texas, USA* MSEC2018-6548, V002T07A003

[85] Yang C, Peng C, Chen Y, Luo T, Chu J 2018 Space-filling scan paths and Gaussian process-aided adaptive sampling for efficient surface measurements *Precis. Eng.* **54** 412–9

[86] Ongsulee P 2018 Artificial intelligence, machine learning and deep learning *Int. Conf. ICT Knowl. Eng., Bangkok, Thailand* 17

[87] Ekberg P, Su R, Leach R 2017 High-precision lateral distortion measurement and correction in coherence scanning interferometry using an arbitrary surface *Opt. Express* 18703–12

[88] Kapłonek W, Sutowska M, Ungureanu M, Çetinkaya K 2018 Optical profilometer with confocal chromatic sensor for high-accuracy 3D measurements of the uncirculated and circulated coins *J. Mech. Energy Eng.* **2** 181–92

[89] Parra Escamilla G A, Kobayashi F, Otani Y 2017 Three-dimensional surface measurement based on the projected defocused pattern technique using imaging fiber optics *Opt. Commun.* **390** 57–60

[90] Eastwood J, Newton L, Leach R, Piano S 2022 Generation and categorisation of surface texture data using a modified progressively growing adversarial network *Precis. Eng.* **74** 1–11

[91] Todhunter L, Senin N, Leach R, Lawes S, Blateyron F, Harris P 2018 A programmable software framework for the generation of simulated surface topography *EUSPEN 2018* 138400

[92] Arezki Y, Zhang X, Mehdi-Souzani C, Anwer N and Nouira H 2018 Investigation of minimum zone assessment methods for aspheric shapes *Precis. Eng.* **52** 300–7

[93] Arezki Y, Nouira H, Anwer N, Mehdi-Souzani C 2018 A novel hybrid trust region minimax fitting algorithm for accurate dimensional metrology of aspherical shapes *Measurement (London)* **127** 134–40

[94] Karow H H 2004 *Fabrication Methods for Precision Optics* (Wiley-Interscience : Hoboken, NJ)

[95] Wang Z, Qu W, Yang F, Tian A, Asundi A 2017 Absolute measurement of aspheric lens with electrically tunable lens in digital holography *Opt. Lasers Eng.* 313–8

[96] Shao G, Hai R, Sun C 2020 3D Printing Customized Optical Lens in Minutes *Adv. Opt. Mater.* 1901646

[97] Adams D, Ament S 2018 Understanding aspheric lenses: key specifications and their impact on performance *Optik & Photonik* 60–3

[98] *ISO 10110* 2007 - *Part12, Optics and photonics – Preparation of drawings for optical elements and systems*

[99] Lever J, Krzywinski M, Altman N 2017 Points of Significance: Principal component analysis *Nat. Methods* **14** 641-2

[100] Van Der Maaten L, Hinton G 2008 Visualizing data using t-SNE *Journal of Machine Learning Research* **9** 2579-605

[101] Fisher R A 1936 The use of multiple measurements in taxonomic problems *Ann. Eugen.* **7** 179-188

[102] Hyvärinen A, Oja E 2000 Independent component analysis: Algorithms and applications *Neural Networks* **13** 411-430

[103] Guo Y, Wang H, Hu Q, Liu H, Liu L, Bennamoun M 2021 Deep Learning for 3D Point Clouds: A Survey *IEEE Trans. Pattern Anal. Mach. Intell.* **43** 4338-64

[104] Segal A V, Haehnel D, Thrun S 2009 Generalized ICP. In: J Trinkle, Y Matsuoka, J A Castellanos *Proc. Robot.: Sci. Syst.* (MIT Press: Seattle) 21

[105] Rusinkiewicz S, Levoy M 2001 Efficient variants of the ICP algorithm *Proc. Internat. Conf. 3DIM, Quebec City, Canada* 145–52

[106] Zhang Z M, Catalucci S, Thompson A, Leach R, Piano S 2022 Applications of data fusion in optical coordinate metrology: a review *Int. J. Adv. Manuf. Technol.* **2023** 1341-56

[107] Wang Y, Solomon J 2019 Deep closest point: Learning representations for point cloud registration *Proc. IEEE Int. Conf. Comput. Vis.*, Soeul*, South Korea* 3523–32

[108] Autodesk Fusion 360 for Education https://www.autodesk.co.uk/products/fusion-360/education

[109] Image2Surface https://github.com/hanskellner/Fusion360Image2Surface

[110] CloudCompare https://www.danielgm.net/cc/

[111] Mountains® 2023 https://www.digitalsurf.com/

[112] Vopson M M 2021 Estimation of the information contained in the visible matter of the universe *AIP Adv.* **11** 105317

[113] Aoyama T, Hayakawa M, Kinoshita T, Nio M 2012 Tenth-Order QED Contribution to the Electron g-2 and an Improved Value of the Fine Structure Constant *Phys. Rev. Lett.* **109** 111807

[114] Pomerleau F, Colas F, Siegwart R 2015 A review of point cloud registration algorithms for mobile robotics *Found. Trends Robotics* **4** 1–104