Towards Improved Targetless Registration and Deformation Analysis of TLS Point Clouds Using Patch-based Segmentation

A thesis accepted by the
Faculty of Aerospace Engineering and Geodesy of the University of Stuttgart
in fulfillment of the requirements for the degree of
Doctor of Engineering Sciences (Dr.-Ing.)

by
Yihui Yang, M.Sc.
born in Anhui, China

Main referee:	Prof. Dr.-Ing. habil. Dr. h.c. Volker Schwieger
Co-referee:	Prof. Dr. Corinna Harmening
Co-referee:	Prof. Dr.-Ing. Christoph Holst
Day of defense:	20.07.2023

Institute of Engineering Geodesy (IIGS)
University of Stuttgart
2023

Abstract

The geometric changes in the real world can be captured by measuring and comparing the 3D coordinates of object surfaces. Traditional point-wise measurements with low spatial resolution may fail to detect inhomogeneous, anisotropic and unexpected deformations, and thus cannot reveal complex deformation processes. 3D point clouds generated from laser scanning or photogrammetric techniques have opened up opportunities for an area-wise acquisition of spatial information. In particular, terrestrial laser scanning (TLS) exhibits rapid development and wide application in areal geodetic monitoring owing to the high resolution and high quality of acquired point cloud data. However, several issues in the process chain of TLS-based deformation monitoring are still not solved satisfactorily. This thesis mainly focuses on the targetless registration and deformation analysis of TLS point clouds, aiming to develop novel data-driven methods to tackle the current challenges.

For most deformation processes of natural scenes, in some local areas no shape deformations occur (i.e., these areas are rigid), and even the deformation directions show a certain level of consistency when these areas are small enough. Further point cloud processing, like stability and deformation analyses, could benefit from the assumptions of local rigidity and consistency of deformed point clouds. In this thesis, thereby, three typical types of locally rigid patches — small planar patches, geometric primitives, and quasi-rigid areas — can be generated from 3D point clouds by specific segmentation techniques. These patches, on the one hand, can preserve the boundaries between rigid and non-rigid areas and thus enable spatial separation with respect to surface stability. On the other hand, local geometric information and empirical stochastic models could be readily determined by the points in each patch.

Based on these segmented rigid patches, targetless registration and deformation analysis of deformed TLS point clouds can be improved regarding accuracy and spatial resolution. Specifically, small planar patches like supervoxels are utilized to distinguish the stable and unstable areas in an iterative registration process, thus ensuring only relatively stable points are involved in estimating transformation parameters. The experimental results show that the proposed targetless registration method has significantly improved the registration accuracy. These small planar patches are also exploited to develop a novel variant of the multiscale model-to-model cloud comparison (M3C2) algorithm, which constructs prisms extending from planar patches instead of the cylinders in standard M3C2. This new method separates actual surface variations and measurement uncertainties, thus yielding lower-uncertainty and higher-resolution deformations. A coarse-to-fine segmentation framework is used to extract multiple geometric primitives from point clouds, and rigorous parameter estimations are performed individually to derive high-precision parametric deformations. Besides, a generalized local registration-based pipeline is proposed to derive dense displacement vectors based on segmented quasi-rigid areas that are corresponded by areal geometric feature descriptors. All proposed methods are successfully verified and evaluated by simulated and/or real point cloud data. The choice of proposed deformation analysis methods for specific scenarios or applications is also provided in this thesis.

Kurzfassung

Die geometrischen Veränderungen in der realen Welt können durch Messung und Vergleich der 3D-Koordinaten von Objektoberflächen erfasst werden. Herkömmliche punktuelle Messungen mit geringer räumlicher Auflösung können inhomogene, anisotrope und unerwartete Verformungen nicht erkennen und somit komplexe Verformungsprozesse nicht aufdecken. 3D-Punktwolken, die durch Laserscanning oder photogrammetrische Verfahren erzeugt werden, haben die Möglichkeit eröffnet, räumliche Informationen flächendeckend zu erfassen. Insbesondere das terrestrische Laserscanning (TLS) weist aufgrund der hohen Auflösung und Qualität der gewonnenen Punktwolkendaten eine rasche Entwicklung und breite Anwendung in der flächenhaften geodätischen Überwachung auf. Allerdings sind einige Probleme in der Prozesskette der TLS-basierten Deformationsüberwachung noch nicht zufriedenstellend gelöst. Diese Arbeit konzentriert sich hauptsächlich auf die Zielzeichen-freie Registrierung und Deformationsanalyse von TLS-Punktwolken und zielt darauf ab, neue datengestützte Methoden zu entwickeln, um die aktuellen Herausforderungen zu bewältigen.

Bei den meisten Deformationsprozessen natürlicher Szenen treten in einigen lokalen Bereichen keine Formveränderungen auf (d. h. diese Bereiche sind starr), und selbst die Deformationsrichtungen sind, wenn die Bereiche klein genug gewählt sind, konsistent. Die weitere Verarbeitung von Punktwolken, wie Stabilitäts- und Deformationsanalysen, könnte von den Annahmen der lokalen Starrheit und Konsistenz der deformierten Punktwolken profitieren. In dieser Arbeit werden daher drei typische Arten von lokal starren Flächen — kleine ebene Flächen, geometrische Primitive und quasi-starre Bereiche — aus 3D-Punktwolken durch spezielle Segmentierungsverfahren erzeugt. Diese Patches können einerseits die Grenzen zwischen starren und nicht-starren Bereichen erkennen und ermöglichen so eine räumliche Trennung in Bezug auf die Oberflächenstabilität. Andererseits können lokale geometrische Informationen und empirische stochastische Modelle anhand der Punkte in jedem Patch leicht bestimmt werden.

Auf Grundlage dieser segmentierten starren Flächen kann die Zielzeichen-freie Registrierung und Deformationsanalyse von deformierten TLS-Punktwolken hinsichtlich der Genauigkeit und der räumlichen Auflösung verbessert werden. Insbesondere werden kleine ebene Flächen wie Supervoxel verwendet, um die stabilen und instabilen Bereiche in einem iterativen Registrierungsprozess zu unterscheiden und so sicherzustellen, dass nur relativ stabile Punkte in die Schätzung der Transformationsparameter einbezogen werden. Die experimentellen Ergebnisse zeigen, dass die vorgeschlagene Zielzeichen-freie Registrierungsmethode die Registrierungsgenauigkeit erheblich verbessert hat. Diese kleinen ebenen Flächen werden auch genutzt, um eine neuartige Variante des Algorithmus für den Multiscale Model-to-Model Cloud Comparison (M3C2) Algorithmus zu entwickeln, der Prismen konstruiert, die sich von ebenen Flächen aus erstrecken, anstatt Zylinder im Standard-M3C2. Diese neue Methode trennt tatsächliche Oberflächenvariationen und Messunsicherheiten, was zu Deformationen mit geringerer Unsicherheit und höherer Auflösung führt. Eine grob-zu-fein Segmentierung wird verwendet, um mehrere geometrische Primitive aus Punktwolken zu extrahieren. Strenge Parameterschätzungen werden individuell durchgeführt, um hochpräzise parametrische Deformationen abzuleiten.

Außerdem wird eine verallgemeinerte, auf lokaler Registrierung basierende Methode vorgeschlagen, um dichte Verschiebungsvektoren abzuleiten, die auf segmentierten, quasi-starren Bereichen basieren, die durch flächenhafte geometrischen Merkmalsdeskriptoren beschrieben werden. Alle vorgeschlagenen Methoden werden durch simulierte und/oder reale Punktwolkendaten erfolgreich verifiziert und bewertet. Die Entscheidung für eine der vorgeschlagenen Deformationsanalysemethoden für spezifische Szenarien oder Anwendungen wird in dieser Arbeit ebenfalls vorgestellt.

Table of Contents

List of Acronyms

4PCS	Four-points Congruent Set
ALS	Airborne Laser Scanning
BP-to-SVP	Boundary Point-to-Supervoxel Plane
C2C	Cloud-to-Cloud
C2M	Cloud-to-Mesh
CD-PB M3C2	Correspondence-driven Plane-based M3C2
DBSCAN	Density-Based Spatial Clustering of Applications with Noise
DEM	Digital Elevation Model
DoD	DEM of Difference
DT	Distance Threshold
ED	Euclidean Distance
F2S3	Feature-to-Feature Supervoxel-based Spatial Smoothing
FEO	Fusion—Exchange Optimization
G-ICP	Generalized-ICP
GNSS	Global Navigation Satellite System
ICP	Iterative Closest Point
LiDAR	Light Detection And Ranging
LMDD	Local Minimum Detectable Deformation
LoDetection	Level of Detection
M2M	Mesh-to-Mesh
M3C2	Multiscale Model-to-Model Cloud Comparison
M3C2-EP	M3C2 with Error Propagation
MLS	Mobile Laser Scanning
PB-M3C2	Patch-based M3C2
PC-1	Point cloud in epoch-1
PC-2	Point cloud in epoch-2
PCA	Principal Component Analysis
PFH	Point Feature Histogram
RANSAC	Random Sample Consensus
RNN	Radius of the Nearest Neighbors

ROI	Region of Interest
SDK	Software Developer Kits
SFD	Synthetic Feature Distance
SOR	Statistical Outlier Removal
Std.	Standard Deviation
SVD	Singular Value Decomposition
TAD	Threshold of Angle Difference
TCM	Threshold of Curvature Median
TIN	Triangulated Irregular Network
TLC	Threshold of Local Curvature
TLS	Terrestrial Laser Scanning
TOF	Time-Of-Flight
Tr-ICP	Trimmed-ICP
TSR	Threshold of Spatial Ranges
VCCS	Voxel Cloud Connectivity Segmentation
VCM	Variance-Covariance Matrix

List of Figures

List of Tables

1 Introduction

1.1 General Background

We live in a dynamic 3D world with ubiquitous geometric changes occurring on object surfaces induced by internal or external physical mechanisms. Accurate detection and analysis of these geometric changes enable investigating their distribution and pattern, and thus reasoning causative factors (e.g., gravity) or influences (e.g., temperature changes) and predicting potential trends. When these objects involve, for example, engineering structures (e.g., buildings or bridges), industrial infrastructures (e.g., power stations or radio telescopes), or natural environment (e.g., landslides or glaciers), measuring and quantifying their geometric changes between several epochs can be regarded as a geodetic deformation monitoring task in engineering geodesy (Heunecke *et al.*, 2013). The spatial data acquired from the geodetic monitoring campaign can be further used for deformation analysis, safety maintenance, and predictive alarming (Moore, 1992; Mukupa *et al.*, 2017).

Traditional point-wise geodetic monitoring strategies using measuring techniques like leveling, total station, or Global Navigation Satellite System (GNSS) can provide high-precision deformation results with rigorous statistical analyses, benefiting from convenient acquisitions of identical points and their stochastic information. Nevertheless, the representative measured points require a careful selection with *a priori* knowledge of the expected deformations, and usually result in a sparse distribution of derived deformations (Harmening, 2020). Besides, accessing these points with measuring instruments (e.g., GNSS antennas) or signalized targets (e.g., prisms) is time- and labor-intensive, especially for objects with large scales or complex deformations (Li *et al.*, 2015). In order to overcome these drawbacks and limitations, area-wise (areal) geodetic monitoring strategies have been proposed and widely applied in engineering geodesy (Heunecke *et al.*, 2013; Kuhlmann *et al.*, 2014). They are realized by sampling with negligibly small discretization intervals instead of carefully planning single measurements on the surfaces of monitored objects (Kuhlmann *et al.*, 2014). Densely distributed measurements (e.g., 2D pixels or 3D points) are thereby yielded from these areal acquisitions within a short time. In most cases, these area-wise monitoring methods do not need to attach signalized targets on monitored surfaces. This contactless acquisition way enables a quasi-continuous representation of measured surfaces but brings more challenges for the following processing and analyses due to the absence of semantic information of the observations (Harmening, 2020).

3D point clouds are one of the most common and straightforward modalities of area-wise spatial data, which can be fast generated from laser scanning or photogrammetric techniques (Qin *et al.*, 2016; Gojcic, 2021; Stilla and Xu, 2023). Figure 1.1 presents an overview of the current approaches to acquiring point clouds and some of their common properties. Point clouds acquired from different sensors (e.g., cameras or laser scanners) on different platforms (e.g., tripods or vehicles) display generally similar representations (i.e., scenes in the form of a set of points lying on objects' surfaces (Gojcic, 2021)) but with dissimilar geometric or radiometric characteristics (e.g., point density, accuracy, intensity, etc.). The choice of methods for capturing point clouds of object surfaces depends on the requirements of data qualities (e.g., resolution and accuracy) as well as the geometry of monitored objects (e.g., measuring distance or object scales). For instance, terrestrial laser scanning (TLS) is normally employed to detect small deformations of engineering constructions or infrastructures due to its high accuracy and high resolution (Kermarrec *et al.*, 2020), while airborne laser scanning (ALS) is often used to monitor geomorphological changes in large-scale mountain areas for its wide scanning coverage and fast acquisition (Xiong *et al.*, 2018).

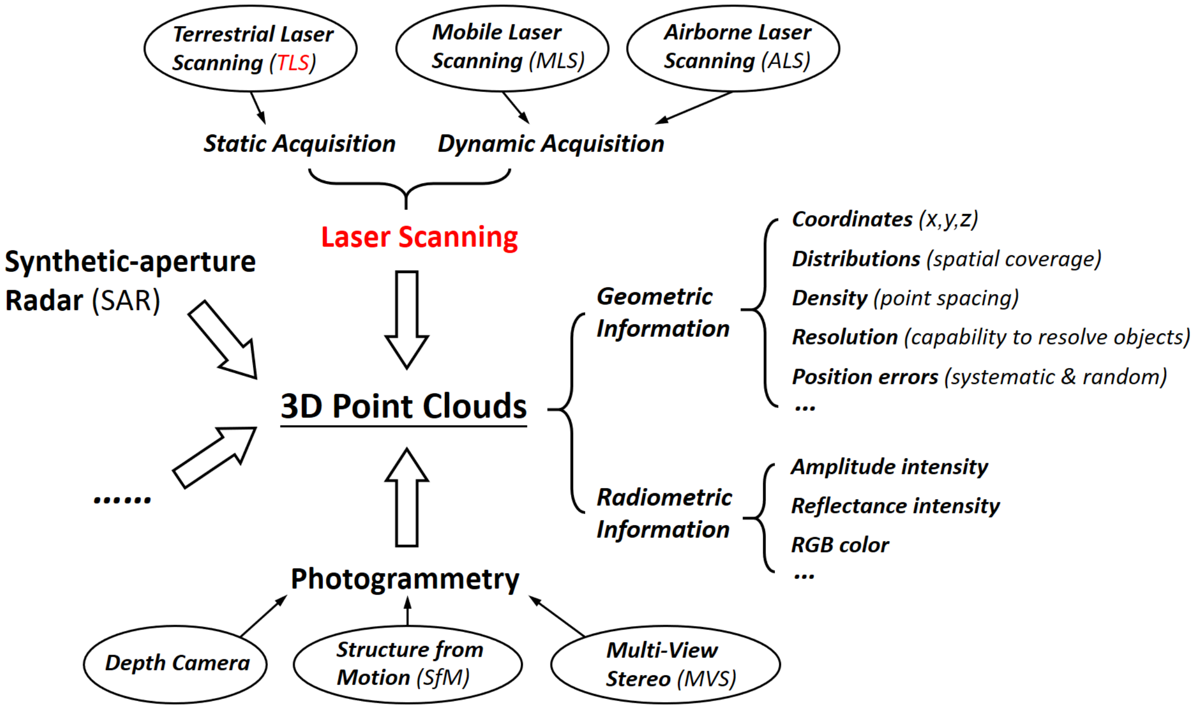


Figure 1.1: Typical acquisition methods of 3D point clouds and their common properties.

On the one hand, point cloud-based change detection and deformation analysis has been rapidly developed in the geodetic as well as geographic, photogrammetric, and remote sensing domains in the last three decades (Vosselman and Maas, 2010; Holst and Kuhlmann, 2016; Qin *et al.*, 2016; Eitel *et al.*, 2016). In the field of computer vision/graphics, on the other hand, advanced and efficient algorithms for 3D point cloud processing, like semantic or instance segmentation and automatic feature detection, have provided more possibilities to cope with the limitations and challenges in the point cloud-based areal geodetic monitoring (Xie *et al.*, 2020; Yang *et al.*, 2016a; Yang *et al.*, 2016b; Poiesi and Boscaini, 2021; Gojcic, 2021; Zahs *et al.*, 2022). In particular, dense point clouds generated by laser scanning or LiDAR (Light Detection And Ranging) technologies, especially by TLS for its high accuracy and resolution, have been widely exploited for high-precision deformation monitoring. The relevant theoretical developments and practical applications have also become the research highlights in engineering geodesy in the recent two decades (Schäfer *et al.*, 2004; Lindenbergh *et al.*, 2005; Tournas and Tsakiri, 2008; Wunderlich *et al.*, 2016; Harmening *et al.*, 2021; Medic *et al.*, 2022). In these contexts, this thesis mainly focuses on TLS point clouds and aims to improve the performance of associated data processing arising from TLS-based deformation monitoring.

1.2 Motivation and Objectives

With the continuous development of sensor performance and data processing strategies, TLS-based deformation monitoring is moving rapidly to a mature state. Its processing chain is depicted in Figure 1.2, where a brief description of each part is given by Harmening (2020). The post-processing procedures after the data acquisition include the two most crucial parts: registration/georeferencing and deformation analysis which are highlighted in red.

For the geodetic monitoring using TLS, at least two-epoch scanning has to be carried out so as to detect and quantify the changes or deformations between two-epoch surfaces represented by point clouds. In addition, more than one scanning viewpoint may be required within an epoch in order to achieve a complete acquisition of the monitored areas. However, each point cloud is generated on the basis of a local coordinate system of the scanner due to different viewpoints and initial scanning orientations. In such cases, the resulting scans are rotated and translated with respect to each other and thus must be transformed into a common local (i.e., registration) or global (i.e., georeferencing) framework (Vosselman and Maas, 2010; Paffenholz, 2012; Wujanz, 2016).

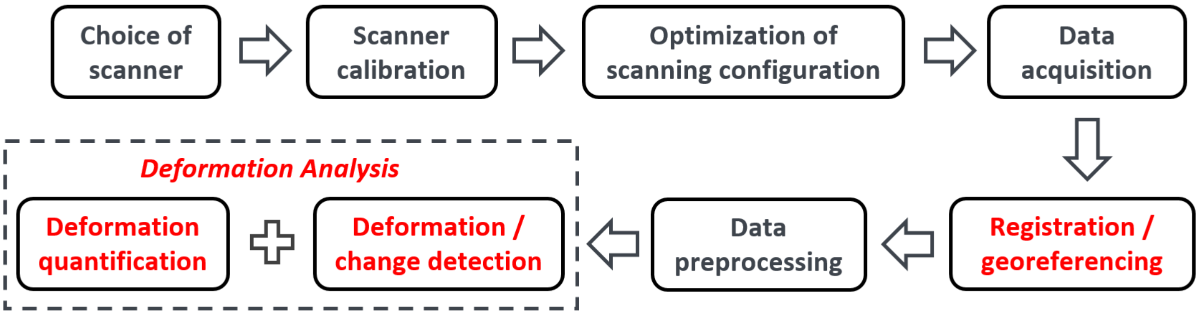

Choice of scanner
Scanner calibration
Optimization of scanning configuration
Data acquisition
Deformation Analysis
Deformation quantification
Deformation / change detection
Data preprocessing
Registration / georeferencing

Figure 1.2: Processing chain of TLS-based deformation monitoring (Harmening, 2020).

Deformation analysis, as the final essential part of the processing chain, consists of change detection and deformation quantification. The former provides a binary result to answer if the monitored object is deformed (or changed) or not (Lindenbergh and Pietrzyk, 2015). From the geodetic perspective, it answers whether the deformation is significant or not by considering the associated stochastic model of estimated deformation values. The latter gives the magnitudes of detected deformations or changes by means of a specified algorithm, and associated directions or signs of deformations can also be output along with the displacement vectors or surface distances.

Numerous methods for the registration and deformation analysis for TLS point clouds have been proposed and successfully applied in man-made or natural scenes from the geodetic domain. Nevertheless, there are still some challenges and room for improvements, such as the targetless registration for deformed point clouds and the detection of small-magnitude deformations under complex topographies. These challenges are often caused by more common issues like the identification of identical points (correspondences) between epochs without semantic information and the establishment of a realistic stochastic model of TLS point clouds (Holst and Kuhlmann, 2016; Wunderlich *et al.*, 2016). Detailed descriptions of these challenges are depicted in Section 2.2.3 and Section 2.3.3.

Even though the deformation process of the natural objects may be inhomogeneous and anisotropic reflected in both deformation magnitudes and directions, on the one hand, the elements (points) within a local area (neighborhood) on the deformed object show high levels of (local) regularity or consistency on stability and deformation behaviors (Gojcic, 2021). On the other hand, although each measurement (laser point) is individually generated by angle and distance observations from the laser scanner, local similarities exist among neighboring points with respect to local orientations, reflectance intensities, standard deviations, etc., and thus a certain level of spatial correlations arise between these neighbors. Besides, some local areas like geometric primitives and individual objects display as rigid bodies in the deformation process, i.e., no distortion occurs within the area. Benefiting from the potential local rigidity and consistency of deformed objects, therefore, spatial division or separation can be performed on TLS point clouds representing the deformed surfaces of the monitored objects, which means the point cloud can be segmented into smaller and assumed-rigid sub-clouds called *patches*. Afterward, stability analysis and deformation analysis can be individually performed on each patch.

Taking into account current challenges in TLS-based deformation monitoring and potential characteristics inherent in TLS point clouds (e.g., local consistency and rigidity), this thesis is dedicated to the development of efficient point cloud segmentation methods for generating rigid patches. The primary objectives of this thesis are **utilizing these segmented patches to enhance the algorithmic performance of (1) targetless registration and (2) deformation analysis for TLS point clouds in challenging contexts** (e.g., point clouds with large-area deformations and complex topographies).

1.3 Main Contributions

This thesis aims to improve two essential parts — targetless registration and deformation analysis of TLS point clouds — in the processing chain of TLS-based deformation monitoring by using patch-based segmentation techniques. To improve the readability of the thesis, the main contributions are briefly summarized as follows:

(1) The concept and assumptions of rigid patches in 3D point clouds are defined (see Section 3.1). Three typical types of rigid patches — small planar patches, geometric primitives, and quasi-rigid areas — can be generated using specified patch-based segmentation strategies (see Section 3.2). Supervoxel-based over-segmentation is adopted to achieve an adaptive spatial division of 3D point clouds, generating small, planar and locally consistent patches. Besides, a simple and efficient coarse-to-fine segmentation framework tailored to extract multiple geometric primitives from point clouds is presented, based on the combination of 3D region growing and Random Sample Consensus (RANSAC). Furthermore, a segmentation-and-validation strategy is proposed for detecting quasi-rigid areas between epochs by exploiting 3D clustering algorithms and correlation estimations between area-wise geometric feature descriptors of associated patches. These generated rigid patches are further employed in the following targetless registration and deformation analysis of TLS point clouds.

(2) A fully automatic targetless registration pipeline is proposed to align two partially deformed point clouds, simultaneously identifying the stable areas during the registration process (see Chapter 4). In this method, coarsely registered point clouds are firstly over-segmented and represented by supervoxels based on the local consistency assumption of deformed objects. A confidence interval based on an approximate assumption of the stochastic model is considered to determine the local minimum detectable deformation for the identification of stable areas. The significantly deformed supervoxels between two scans can be detected progressively by an efficient iterative process, solely retaining the

stable areas to be utilized for the fine registration. The experimental results show that the proposed algorithm exhibits a higher registration accuracy compared with the existing voxel-based method and two robust Iterative Closest Point (ICP) variants.

(3) Based on the segmented small planar patches, a novel M3C2 variant called patch-based M3C2 is proposed (see Section 5.1). This new method constructs prisms from planar patches instead of cylinders in standard M3C2 (Lague *et al.*, 2013) and quantifies surface distances between projected points on the patch plane, avoiding the bias and over-smoothing effect on deformation quantifications in standard M3C2. The definition of *Level of Detection* is also modified by using projected standard deviations along defined deformation directions instead of the local roughness in standard M3C2 and considering the spatial correlations between measurements. Lower uncertainty and higher resolution are achieved to detect small-magnitude deformations under complex surface topographies, thus enabling a better point cloud-based deformation analysis. For segmented geometric primitives and quasi-rigid areas from 3D point clouds, rigorous parameter estimations and local registration are utilized to derive parametric deformations (see Section 5.2) and dense displacement vectors (see Section 5.3) respectively, followed by associated statistical significance tests. The correspondences in the two approaches are established from parametric space or feature space instead of Euclidean space, thus generating real displacements rather than the surface distances by point cloud comparison techniques. All proposed deformation analysis methods are verified and evaluated by simulated and/or real point cloud data, and the choice of three proposed methods for specific scenarios or applications is also provided.

It should be noted that the methodologies presented in this thesis are mainly developed for TLS point clouds considering their extensive use in geodetic monitoring. Nevertheless, the proposed approaches to targetless registration and deformation analysis have the potential to be applied to ubiquitous 3D point clouds captured from other kinds of sensors or platforms (e.g., ALS point clouds or photogrammetric point clouds) under the appropriate parameter or model selections.

1.4 Thesis Outline

This thesis is organized into six chapters, schematically shown in Figure 1.3. Chapter 1 depicts the motivation, objectives, and structure of the thesis and highlights the main contributions of the work. Chapter 2 mainly introduces the theoretical basis and related work of TLS-based deformation monitoring, particularly focusing on point cloud registration and point cloud-based deformation analysis as well as respective current challenges. As the basis of the following two chapters, Chapter 3 systematically describes the concept, implications, and generation methods of rigid patches in 3D point clouds and proposes three typical types of rigid patches that could be segmented from point clouds: Small planar patches, geometric primitives and quasi-rigid areas. In Chapter 4, a supervoxel-based targetless registration method is proposed to align two deformed point clouds, using locally planar and rigid patches to separate the boundaries of surface stabilities. The pipeline is demonstrated on two datasets acquired from an indoor scene and a natural scene. Based on well-registered TLS point clouds, rigorous deformation analyses by means of these segmented rigid patches are presented in Chapter 5. The methodologies of deformation analyses elaborated in Sections 5.1, 5.2, and 5.3 are developed based on the above three types of rigid patches, respectively.

In particular, the proposed registration algorithm in Chapter 4 is further employed in the local registration-based method presented in Section 5.3. All proposed approaches are experimentally validated and evaluated by simulated and/or real point cloud data in related sections. Chapter 6 concludes this thesis by summarizing the connections of contributions and discussing the limitations and future investigations.

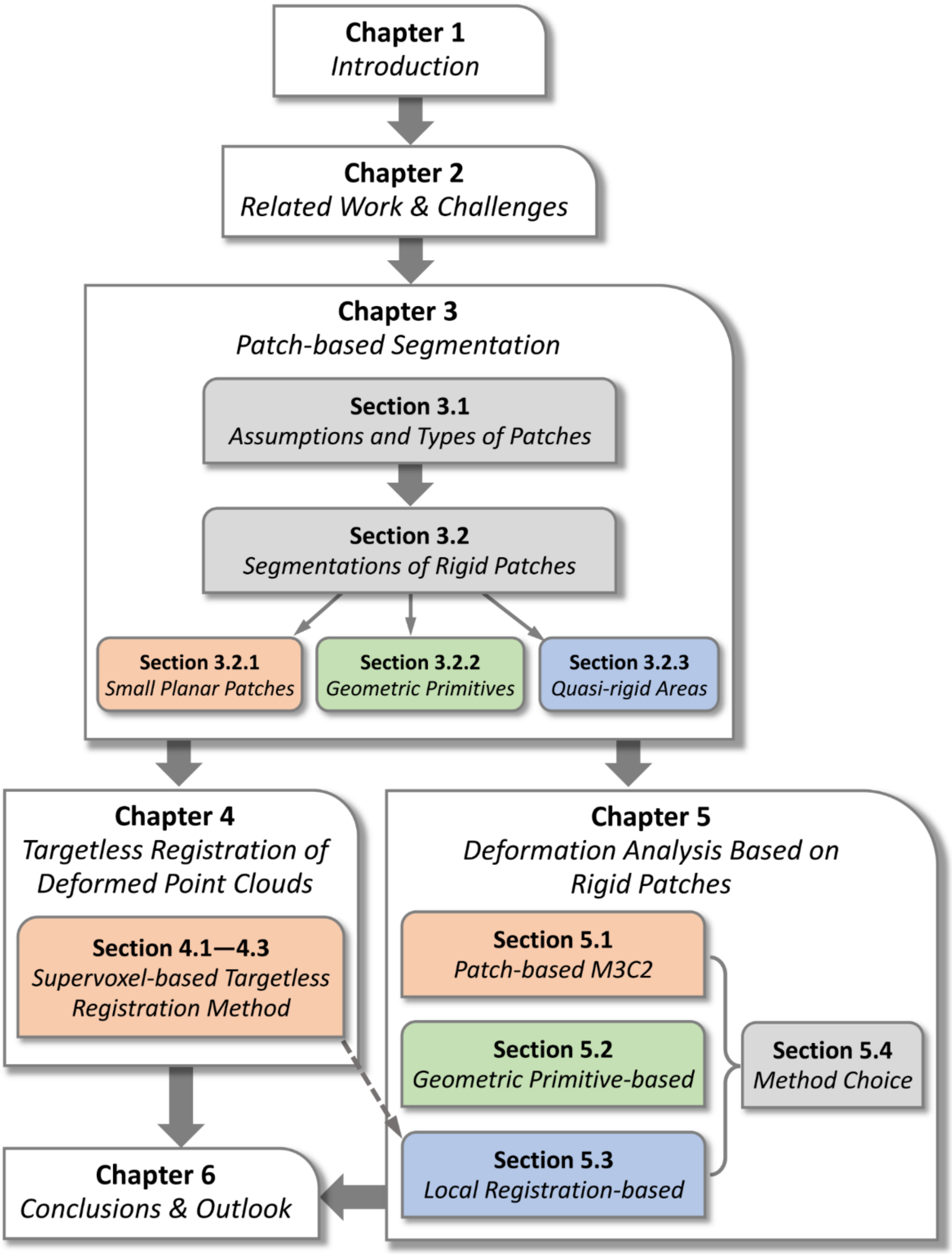


Figure 1.3: The general structure of the thesis (block arrows indicate the recommended reading order; the same colors and dashed arrows indicate potential connections).

2 Fundamentals, Related Work and Challenges of TLS-based Deformation Monitoring

2.1 TLS-based Deformation Monitoring

Terrestrial laser scanning, also referred to as terrestrial LiDAR, acquires 3D coordinates (XYZ) of numerous points on land by emitting laser pulses or carrier waves with known wavelength towards these points and measuring the horizontal and vertical angles and distance from the device (sensor) to the object (Vosselman and Maas, 2010). The acquisition way of scanning could be 2D (e.g., profile scanner) or 3D (e.g., panorama scanner), thus generating point clouds distributed on a 2D profile or 3D surface, respectively. Figure 2.1 demonstrates the basic principle of a TLS system.

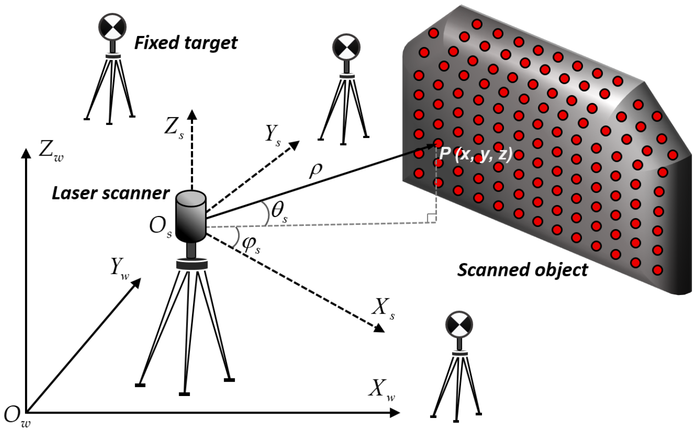


Figure 2.1: The measuring principle of a TLS system.

The measured distance ρ from the scanner center to the object surface can be acquired by time-of-flight (TOF) or phase difference technology (Vosselman and Maas, 2010). The ranging process of the first method can be simplified into the model of Equation (2.1), where c is the speed of electromagnetic wave and T is the TOF of the laser pulse. The phase difference method measures the whole length of modulated carrier wave by computing the phase difference $\Delta\phi_w$ and the number of integer wavelengths n_w, as denoted in Equation (2.2) where λ_w is the wavelength of the modulated signal (Shan and Toth, 2018).

$$\rho_{TOF} = \frac{c \cdot T}{2} \tag{2.1}$$

$$\rho_{phase\_diff} = \frac{\lambda_w}{2}\left(n_w + \frac{\Delta\phi_w}{2\pi}\right) \tag{2.2}$$

The final 3D coordinates (in the scanner-defined local coordinate system $O_s - X_s Y_s Z_s$) measured by the scanner are calculated in Equation (2.3) where θ_s and φ_s are the vertical and horizontal angles. The absolute coordinates in the world coordinate system $O_w - X_w Y_w Z_w$ can be obtained by a georeferencing process using fixed artificial targets mounted on stable areas or by means of external sensors (see Section 2.2.2). Detailed measuring principles of TLS and its applications can be found in Vosselman and Maas (2010) and Shan and Toth (2018).

$$\begin{aligned} x &= \rho \cdot \cos\theta_s \cdot \cos\varphi_s \\ y &= \rho \cdot \cos\theta_s \cdot \sin\varphi_s \\ z &= \rho \cdot \sin\theta_s \end{aligned} \tag{2.3}$$

The modern laser scanners are able to acquire data in a significant higher speed (e.g., up to 1.1 million measurements/second can be acquired by Z+F IMAGER® 5016), compared to conventional surveying instruments (e.g., total station) (Abellán *et al.*, 2011). Moreover, a TLS system can capture the intensity of the reflected pulsed laser and even RGB color information of the object surface with an integrated camera, which allows for a more detailed representation of the monitored objects. These radiometric data are output as relevant attributes along with the 3D coordinates.

In TLS-based deformation monitoring, deformations and changes are detected and analyzed based on the generated 3D point clouds (attached radiometric information may also be utilized) from different epochs. As presented in Figure 1.2, registration (georeferencing can be regarded as a global registration) and deformation analysis are two crucial data processing parts in the pipeline of TLS-based deformation monitoring. Incorrect or inaccurate registration or deformation analysis will directly fail the monitoring task, thus leading them to the most investigated processes. The following two sections present the fundamentals and related work of point cloud registration and deformation analysis, and discuss their main challenges in some application scenarios.

2.2 Point Cloud Registration

Accurate and robust point cloud registration is a prerequisite for accurate deformation analysis in TLS-based deformation monitoring. The registration methods can be basically classified into relative registration and absolute registration, as summarized in Figure 2.2. The former is also called point cloud alignment and rotates and translates one point cloud $\boldsymbol{P} = \{\boldsymbol{p}_i\}_{i=1}^{n}$ (namely the source cloud with n points $\boldsymbol{p}_i$) to the other reference point cloud $\boldsymbol{Q} = \{\boldsymbol{q}_j\}_{j=1}^{m}$ (namely the target cloud with m points $\boldsymbol{q}_j$) for achieving a unification of two local coordinate systems (Holz *et al.*, 2015). The latter is also called georeferencing and transforms the local coordinate systems of point clouds into a global or a superior system (e.g., a national or a world coordinate system). It should be noted that this thesis solely focuses on the rigid registration (i.e., the size and shape of the source point cloud are not changed after the rigid transformation), thus the non-rigid point cloud registration (i.e., the size or shape of the source point cloud may be changed after the non-rigid transformation) is not included in the classification categories. Interested readers are referred to Huang *et al.* (2008) and Dyke *et al.* (2019) for more details about the non-rigid registration of 3D point clouds.

Except for the direct georeferencing way, both absolute and relative registration can be implemented by establishing corresponding elements between two scans and performing a coordinate transformation based on these correspondences (for georeferencing a connection to the superior coordinate system is needed). These correspondences could be identical points or other corresponding geometric features like lines and planes. The goal of point cloud registration is to estimate transformation parameters, including three rotation angles and a translation vector in 3D, that minimize the spatial distances between these correspondences, thus unifying the two coordinate frames (locally or globally) as accurately as possible.

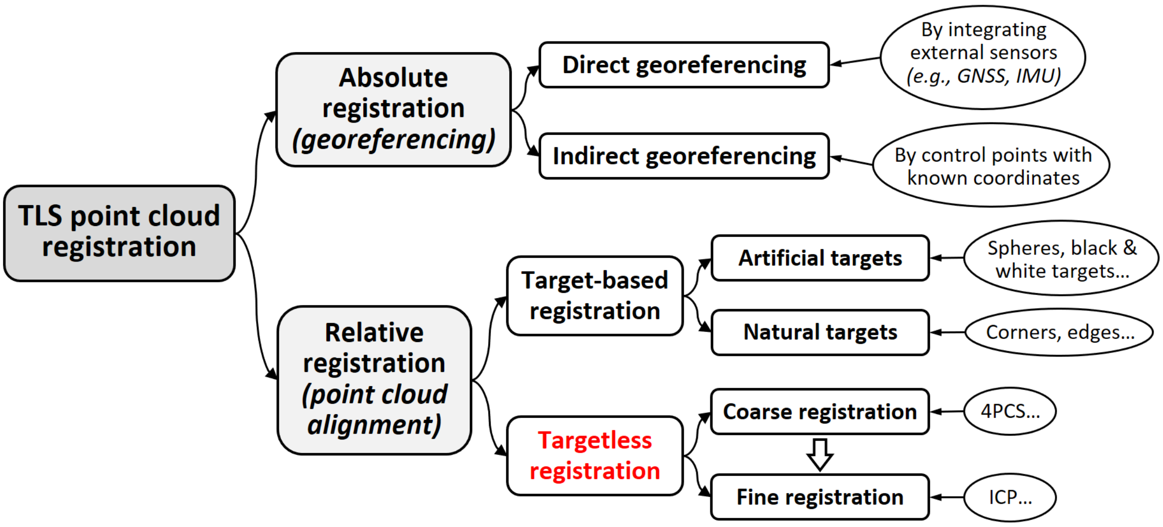

TLS point cloud registration
Absolute registration (georeferencing)
Direct georeferencing
By integrating external sensors (e.g., GNSS, IMU)
Indirect georeferencing
By control points with known coordinates
Relative registration (point cloud alignment)
Target-based registration
Artificial targets
Spheres, black & white targets...
Natural targets
Corners, edges...
Targetless registration
Coarse registration
4PCS...
Fine registration
ICP...

Figure 2.2: Classification of TLS point cloud (rigid) registration.

A transformation matrix $\boldsymbol{T}$ can be constructed by incorporating the rotation matrix $\boldsymbol{R}$ and translation vector $\boldsymbol{t}$. The determination of $\hat{\boldsymbol{T}}$ is normally achieved by an optimal-estimation process based on an objective function defined as (Holz *et al.*, 2015)

$$\underset{4\times4}{\hat{\boldsymbol{T}}} = \begin{bmatrix} \underset{3\times3}{\hat{\boldsymbol{R}}} & \underset{3\times1}{\hat{\boldsymbol{t}}} \\ \underset{1\times3}{\boldsymbol{0}} & 1 \end{bmatrix} = \underset{\boldsymbol{R},\boldsymbol{t}}{\operatorname{argmin}} \sum_{i=1}^{N} \left\| \boldsymbol{R}\boldsymbol{p}_i + \boldsymbol{t} - \phi(\boldsymbol{p}_i, \boldsymbol{Q}) \right\|^2 , \tag{2.4}$$

where $\phi(\boldsymbol{p}_i, \boldsymbol{Q})$ is a correspondence function that maps $\boldsymbol{p}_i$ in the source cloud $\boldsymbol{P}$ to its corresponding point in target cloud $\boldsymbol{Q}$, and N is the number of correspondences.

The translation vector $\boldsymbol{t}$ and the rotation matrix $\boldsymbol{R}$ can be denoted by Equation (2.5) and Equation (2.6), respectively.

$$\boldsymbol{t} = [t_x \ t_y \ t_z]^T \tag{2.5}$$

$$\begin{aligned} \boldsymbol{R} = \boldsymbol{R}_x\boldsymbol{R}_y\boldsymbol{R}_z &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos R_x & -\sin R_x \\ 0 & \sin R_x & \cos R_x \end{bmatrix} \begin{bmatrix} \cos R_y & 0 & \sin R_y \\ 0 & 1 & 0 \\ -\sin R_y & 0 & \cos R_y \end{bmatrix} \begin{bmatrix} \cos R_z & -\sin R_z & 0 \\ \sin R_z & \cos R_z & 0 \\ 0 & 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} \cos R_y \cos R_z & -\cos R_y \sin R_z & \sin R_y \\ \sin R_x \sin R_y \cos R_z + \cos R_x \sin R_z & -\sin R_x \sin R_y \sin R_z + \cos R_x \cos R_z & -\sin R_x \cos R_y \\ -\cos R_x \sin R_y \cos R_z + \sin R_x \sin R_z & \cos R_x \sin R_y \sin R_z + \sin R_x \cos R_z & \cos R_x \cos R_y \end{bmatrix} \\ &= \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \overset{\substack{\text{when } R_x, R_y, R_z \\ \text{are small}}}{\approx} \begin{bmatrix} 1 & -R_z & R_y \\ R_z & 1 & -R_x \\ -R_y & R_x & 1 \end{bmatrix} \end{aligned} \tag{2.6}$$

Correspondences in the form of identical points can be artificial signalized targets mounted within the scanned area (normally used in indirect georeferencing and target-based registration), feature points extracted from point clouds (normally used in coarse registration), or nearest neighboring points (normally used in ICP-based fine registration (Besl and McKay, 1992)), etc. The quantity and quality of correspondences have a significant influence on the accuracy of registration results (Fan *et al.*, 2015; Janßen *et al.*, 2019).

The optimal transformation parameters are typically solved by linear least squares (when rotation angles are small) (Low, 2004), nonlinear least squares (where an iterative process is required) (Gruen and Akca, 2005), or singular value decomposition (SVD) by which the solution can be fast achieved (Kabsch, 1976). Notably, if available, a variance-covariance matrix (VCM) for correspondences could be integrated into linear least squares and nonlinear least squares. The registered point cloud $\boldsymbol{Q}'$ can be readily obtained with the transformation parameters or matrix by

$$\begin{bmatrix} q'_{j\_x} \\ q'_{j\_y} \\ q'_{j\_z} \end{bmatrix} = \hat{\boldsymbol{R}} \begin{bmatrix} q_{j\_x} \\ q_{j\_y} \\ q_{j\_z} \end{bmatrix} + \hat{\boldsymbol{t}} \quad \text{or} \quad \begin{bmatrix} q'_{j\_x} \\ q'_{j\_y} \\ q'_{j\_z} \\ 1 \end{bmatrix} = \hat{\boldsymbol{T}} \begin{bmatrix} q_{j\_x} \\ q_{j\_y} \\ q_{j\_z} \\ 1 \end{bmatrix}. \tag{2.7}$$

The following subsections present detailed principles and related work of relative and absolute registration, respectively.

2.2.1 Relative Registration (Point Cloud Alignment)

In most geodetic monitoring cases where absolute positions are not necessarily required, the deformation magnitudes between epochs are quantified independently of the absolute coordinate system. The derived movements or distortions relative to the reference epoch can also indicate the deformation behaviors on the point cloud surface in the local coordinate system. Therefore, a relative registration (i.e., alignment) of one point cloud to another is basically enough for revealing the expected deformation information in the scene, assuming the monitored surface in the reference epoch is stable.

The typical relative registration is conducted on pairwise point clouds with overlapping areas. These pairwise point clouds can be acquired (1) in the same epoch while from different viewpoints, or (2) from the same viewpoint but in different epochs, or even (3) from different viewpoints and in different epochs. Figure 2.3 shows strategies to register multi-viewpoint and multi-epoch point clouds into a common frame/epoch (i.e., the reference frame/epoch).

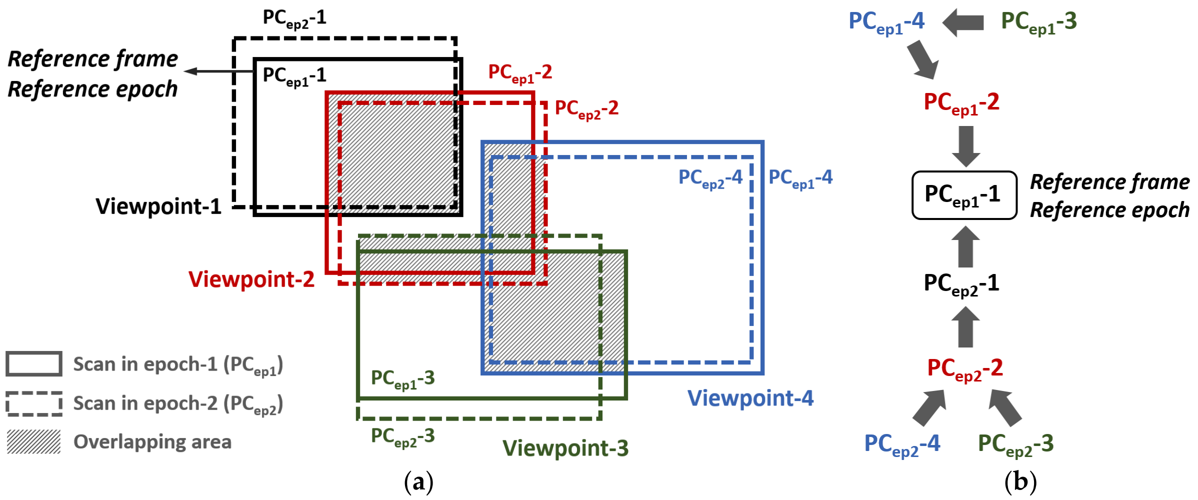


Figure 2.3: Registration strategies for multiple scans from different viewpoints and epochs: (a) Scans acquired from different viewpoints and epochs with overlapping areas; (b) Possible registration sequences of scans to the reference frame/epoch (each arrow indicates a transformation matrix).

For multiple point clouds captured from different viewpoints and epochs, the registration between pairwise point clouds should be performed stepwise based on fixed artificial targets or the stable point cloud surfaces within overlapping areas. For deformed point clouds from monitoring epochs, alignment is normally conducted between spatially adjacent scans (with larger overlaps) first in one epoch and then between epochs. Point clouds are ultimately registered to the reference cloud (frame) after the former step and unified to the reference epoch after the latter step. The choice of two scans for alignment depends on the number and distribution of common fixed targets or the dimension of common stable areas.

Typically, relative registration can be achieved by target-based or targetless methods. Placing a certain number of artificial targets (e.g., signalized planar boards, corner reflectors, or spheres, etc.) evenly distributed within the scanned area is one of the most popular approaches to register multi-station and multi-epoch laser scans in geodetic monitoring tasks. The identical points estimated from these signalized targets directly derive reliable correspondences between two TLS point clouds as long as these targets keep stable. However, several downsides of the target-based registration strategy are evident in spite of its superiorities of high accuracy and high reliability (Janßen *et al.*, 2019; Janßen *et al.*, 2022), such as the necessity to access the monitored areas, which may be cumbersome, dangerous or even impossible (Wujanz, 2016). Besides, the stability of target positions between epochs cannot be guaranteed due to possible movements of their located regions and other unexpected interferences like the wind (Friedli, 2020). Stable and distinct geometric or radiometric features in the scanned scene can also be utilized as so-called natural targets to construct correspondences. However, the usage of these artificial targets needs prior knowledge with respect to their stability and distinction between epochs. Thereby, targetless registration solely based on the point cloud data has been of great research interest since the beginning of the 1990s (Vosselman and Maas, 2010).

There are numerous algorithms aiming to register one point cloud to another without artificial targets in the scanned scene. These methods are generally under a coarse-to-fine procedure (Cheng *et al.*, 2018) and can be roughly divided into three categories including feature-based (Rusu *et al.*, 2009; Yang *et al.*, 2016a), four-points congruent set (4PCS)-based (Aiger *et al.*, 2008; Theiler *et al.*, 2014), and ICP-based methods (Besl and McKay, 1992; Park and Subbarao, 2003). The former two strategies are mainly employed for coarse registration in which only several points or features are utilized, while ICP or its variants are the most commonly used for fine registration owing to its conceptual simplicity and high usability. Nevertheless, the standard ICP framework is only capable of deriving accurate transformation parameters under the condition of no or merely negligible deformations occurring between two scans even if a good initial alignment is given (Wunderlich *et al.*, 2016). Significant deformations may cause incorrect correspondences in the ICP algorithm, resulting in erroneous registration of two scans, ultimately leading to a wrong deformation analysis. Hence, the unstable areas must be rejected from the respective point cloud in the registration process (Wujanz, 2016; Friedli, 2020). It should be noted that the unstable areas incorporating deformations, movements and non-overlapping areas (i.e., the areas are not measured in all epochs) are normally unknown in a point cloud without *a priori* knowledge. Thus, automatic identification of the stable areas in two unregistered point clouds plays a significant role in targetless registration, especially in complex natural scenes where the stable and unstable regions are difficult to be distinguished manually.

2.2.2 Absolute Registration (Georeferencing)

The georeferencing process can be direct or indirect. A direct georeferencing of TLS measurements is typically implemented by using external sensors to acquire the 3D position (e.g., by GNSS or total station) and orientations (e.g., by inertial measurement unit or compass) of the scanner. This approach has been widely applied in MLS and ALS where the scanner keeps a dynamic state when measuring (Meng *et al.*, 2017; Klingbeil *et al.*, 2017). Other efficient direct georeferencing methods are also used in TLS. For example, Paffenholz (2012) utilized two GNSS antennas attached on the laser scanner to estimate its position and orientations, while Pandžić *et al.* (2017) adopted a "station-orientation" procedure by a backsight target assuming that the scanner only rotates by the *Z*-axis. Different to the direct way, indirect georeferencing of TLS point clouds is implemented by a coordinate system transformation based on at least three points (i.e., control points) well-distributed in the scans. The absolute positions of these control points are known in a geodetic network or can be acquired by other sensors (e.g., total station or GNSS). The measurement of target positions can be carried out while acquiring the scans, which ensures a successful registration in cases where targets move between scanning epochs (Glira *et al.*, 2013).

Since this thesis only focuses on the quantification of relative deformations between two TLS point clouds, absolute registration will not be discussed in the following chapters. Detailed background and related work of georeferencing can be found in Vosselman and Maas (2010), Scaioni (2005), Paffenholz (2012) and Pandžić *et al.* (2017).

2.2.3 Challenges of Targetless Registration of Deformed Point Clouds

As stated in Section 2.2.1, only stable (relative to the reference point cloud) areas or points should be involved in the registration of deformed point clouds. To tackle the challenge of targetless registration of deformed point clouds, a variety of solutions have been proposed to avoid employing unstable areas to estimate transformation parameters. Two typical strategies are commonly used: Stable feature-based and robust ICP-based.

The first strategy concentrates on the construction of invariant features between two scans. These features include hand-crafted feature points, lines, planes, etc. (Yang *et al.*, 2015; Yang *et al.*, 2016a; Chen *et al.*, 2020). The RANSAC algorithm considering the geometric constraints could then be used to extract the identical features without movements and deformations so that the wrong and deformed correspondences could be rejected (Theiler *et al.*, 2014). However, these feature-based methods are typically used for coarse registration due to the potential errors of feature extraction and matching, and the distribution of extracted features directly influences the registration accuracy. The stable geometric features are easily extracted from the scenes incorporating man-made objects like buildings, whereas for natural scenes in particular the smooth surfaces lack of corners and edges, there would be few detectable geometric features for establishing correspondences. Besides, respective structures in the scans may fail the right correspondence establishment as well.

The second strategy is the robust variant of standard ICP algorithm. The basic idea of these variants is to reweight (soft rejection) or remove (hard rejection) the correspondences comprised of the points in unstable areas of two scans in the calculation of transformation parameters. The Generalized-ICP (G-ICP) proposed by Segal *et al.* (2009) exhibits better robustness against wrong correspondences by introducing an empirical VCM for each point, assuming that the variance along the surface normal direction is lower than that along the local plane. Accordingly, correspondences where the surface orientations of two points are inconsistent make a much smaller contribution to the transformation estimation. Nevertheless, G-ICP cannot weaken the influence of unstable areas moving along the normal direction, and the assumed stochastic model is unrealistic. Chetverikov *et al.* (2005) proposed a method called Trimmed-ICP (Tr-ICP) which directly rejects a certain percentage of incorrect correspondences in each iteration by setting a specified ratio (also called overlapping parameter). Although this approach is fast and easily implementable, it is hard to determine the overlapping parameter without prior information in a complex scenario.

In the field of engineering geodesy, Wujanz *et al.* (2016) and Friedli and Wieser (2016) have proposed surface-based targetless registration methods by using a voxel-based segmentation strategy. Each segment (i.e., voxel derived from an octree-based spatial index structure) is represented by its centroid or uniform subsampling. Spatially equal-sized voxels that are subject to deformation will not be employed to compute the transformation parameters. The voxel-based partition for point clouds has dramatically improved the computational efficiency compared to a single point-based strategy. However, the equal-sized segments containing deformed parts may still be regarded as stable areas if these deformed points have a tiny influence on the calculated centroid due to their symmetric geometric distribution in one voxel. These retained voxels incorporating outliers will eventually affect the transformation estimation. A uniform subsampling strategy adopted by Wujanz *et al.* (2016) allows for a better description of one segment, but a mixed partition including both stable and unstable points may be classified as an unstable voxel, resulting in insufficient or poorly distributed points for the following transformation estimation. Therefore, the equal-sized spatial partition strategy is unreasonable when coping with the targetless registration of point clouds containing irregularly distributed deformations and changes.

In addition to the above methods, techniques like robust registration by efficient outlier removal strategies (Bustos and Chin, 2017; Yan *et al.*, 2023) and neural network-based registration (Gojcic *et al.*, 2020a; Huang *et al.*, 2022) applied in the field of computer vision show better performance but bring higher complexity and time consumption.

2.3 Point Cloud-based Deformation Analysis

Multitemporal 3D point clouds generated from laser scanning or photogrammetric techniques are widely exploited to reveal geometric surface changes in the real world (Qin *et al.*, 2016; Stilla and Xu, 2023). Two-epoch point cloud comparison under a unified georeferencing frame can directly detect and quantify geometric changes of the monitored surface between monitoring epochs. In the field of engineering geodesy, this process is formulated as "point cloud-based deformation analysis" (Kuhlmann *et al.*, 2014; Holst and Kuhlmann, 2016), where the term "deformation" herein, including both rigid-body movement and shape deformation (or distortion), has equivalent implications to "change" used in the geoscientific domain. The analytical results of the former are typically displayed via displacement vectors (or displacement fields), while the shape deformation could be directly reflected by the calculations of surface distances (for a point) or volume changes (for an area) by point cloud comparison techniques (Girardeau-Montaut *et al.*, 2005). In addition, there is another geometric change: Existing objects disappear, or new objects appear in the scanned scene (in this case, correspondences of these objects do not exist between epochs). This kind of changes can be regarded as the movement out of the scene (i.e., movements with unknown directions and magnitudes), and they could be analyzed by simply calculating the Euclidean distances (EDs) or volume changes between two point clouds. A general category of the geometric changes of object surfaces is summarized in Figure 2.4.

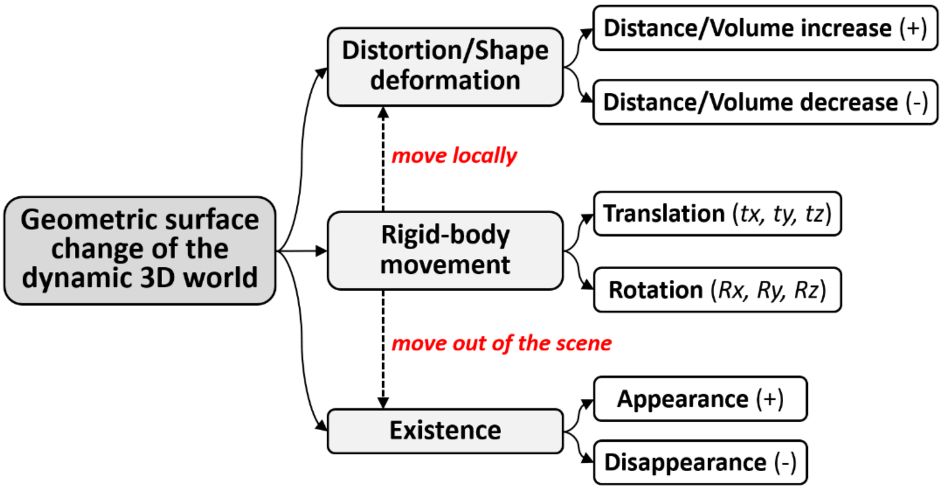


Figure 2.4: General types of geometric surface changes.

The difference between the distance and displacement calculated on a point should be emphasized. Displacement vectors are derived to show the real movement between corresponding elements in two epochs, while surface distances are calculated under specific criteria (e.g., the distance between the nearest points or the distance along a defined direction). Thereby, the distance value (*Dist*) from one epoch to the other may differ from the distance in reverse sequence, while the displacement magnitude (*Disp*) will not be affected by the sequence of comparison, as illustrated in Figure 2.5. However, real displacement vectors can be derived only when realistic corresponding elements exist in point clouds from both epochs. Hence, for the areas with high distortions or object appearance/disappearance, real displacements cannot be calculated in the absence of realistic correspondences. In such cases, surface distances are usually utilized to represent the magnitudes of surface changes.

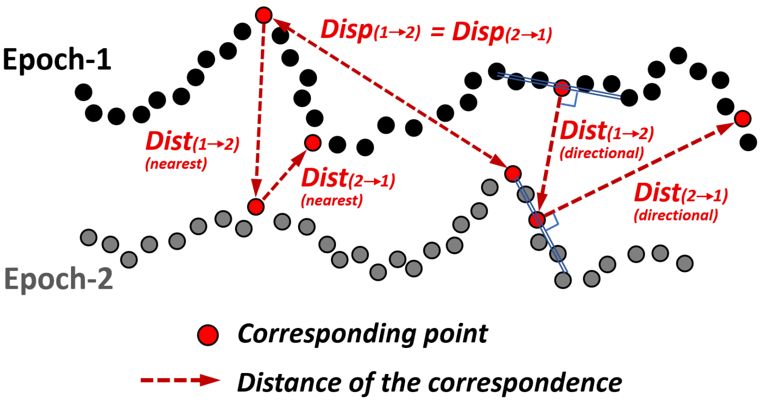


Figure 2.5: The difference between surface distances and real displacements.

To detect and quantify the significant deformations between two point clouds, numerous methods have been proposed and applied for monitoring infrastructural (Kermarrec *et al.*, 2020), industrial (Holst *et al.*, 2019), and environmental objects (Zhong *et al.*, 2021). The adoption of specific algorithms depends on the objectives of deformation analysis (e.g., to obtain 3D displacement vectors or directly measure the spatial distances between two surfaces) and the geometric information of acquired point clouds (e.g., point density, topographical complexity, surface roughness, etc.).

Similar to the point-wise geodetic monitoring, the prerequisites of point cloud-based deformation analysis are (1) registering two scans into one local or global coordinate system and (2) determining the corresponding elements within successive point clouds in explicit or implicit ways (Wunderlich *et al.*, 2016; Gojcic, 2021). Then, the EDs of these correspondences will be calculated as the deformation values. Therefore, the strategies of point cloud-based deformation analysis can be classified according to the types or definitions of these explicit or implicit correspondences between two scans.

2.3.1 Classification of Strategies According to Correspondence Types

The geometric types of correspondences in point cloud-based deformation analysis can be in the form of point-to-point, point-to-surface (point-to-model), or surface-to-surface (model-to-model) (Mukupa *et al.*, 2016), as listed in Table 2.1. The points in one point cloud can be original measured points, extracted feature points, or estimated points from parameterized geometric primitives, while the modeled surfaces of another point cloud are usually derived by meshing or parameterization techniques.

Table 2.1: Three categories of point cloud-based deformation analysis by the correspondence type.

Methods	General principles	Representative algorithms
Point to point-based	Calculating EDs between the corresponding points in two point clouds (PCs).	C2C (Girardeau-Montaut *et al.*, 2005) F2S3 (Gojcic *et al.*, 2020b)
Point to surface-based	Calculating EDs between the points in one PC and the modeled surface of another PC.	C2M (Cignoni *et al.*, 1998) B-spline surface (Harmening, 2020)
Surface to surface-based	Calculating EDs between the generated points from modeled surfaces or estimating parametric changes between the models of two PCs.	M2M (Aspert *et al.*, 2002) M3C2 (Lague *et al.*, 2013) B-spline surface (Harmening, 2020)

The simplest way of point to point-based methods is to directly compute EDs between the individual points of the source point cloud and their respective closest neighbors in the target point cloud. These individual points could be original measured points or the centers/centroids of divided voxels of point clouds (adopted by Girardeau-Montaut *et al.*, 2005). The methods based on the correspondence in the form of point-to-point are generally called cloud-to-cloud (C2C). While C2C exhibits the highest simplicity and computational efficiency, EDs between these nearest points are only sensitive to out-of-plane deformations and not able to denote realistic displacements between corresponding points over epochs. Besides, a large point spacing makes the distance undesirably overestimated. Thereby, distance calculations between constructed (hand-crafted or learned) feature points that are regarded as identical elements become another option, especially for detecting in-plane movements. For example, Gojcic *et al.* (2020b) proposed feature-to-feature supervoxel-based spatial smoothing (F2S3) that describes feature points solely by the point clouds' geometric information and establishes correspondences between epochs in the feature space under a deep learning framework. Wagner *et al.* (2017) utilized the color information from captured images along with the laser scans to detect identical elements in different-epoch point clouds by image-based feature points.

In point-to-surface correspondences, the reference point cloud is approximated by a surface which can be a mesh or a parametric model. The EDs between the points in the other point clouds and this surface are computed as deformations. The distance between a point to a surface implies that an interpolated point may be generated from the surface or model as the corresponding point for obtaining the shortest distance. Thus, a point to surface-based method is finally transferred into the point-to-point way. For example, a cloud-to-mesh (C2M) method takes the closest facet (triangle) or edge in the triangulated mesh of the reference point cloud to compute the point-to-plane distance (Cignoni *et al.*, 1998), or the distances between points to an approximating B-spline surface can be calculated by resampling dense points from the parametric surface (Harmening, 2020).

Similarly, the surface to surface-based methods can be transferred into a point to surface-based or even point to point-based methods by generating points from one or two surfaces. A mesh-to-mesh (M2M) method calculates the distances between vertices of one meshed point cloud to their closest facet (triangle) or edge of the other meshed point cloud (Aspert *et al.*, 2002). Lague *et al.* (2013) proposed the multiscale model-to-model cloud comparison (M3C2) to calculate the distance between two average positions of their local neighborhoods in two point clouds captured by a cylinder along the normal direction. In addition, deriving control points from modeled B-spline surfaces of two point clouds and calculating the distance between corresponding control points have become an efficient approach to detect and quantify deformations between the point clouds with high measurement noise and non-uniform point densities (Harmening *et al.*, 2021).

It should be noted that not all existing approaches can be clearly assigned into these three classes (Harmening, 2020). Therefore, a new classification of point cloud-based deformation analysis according to the definition of correspondences instead of their geometric types may allow for including more existing methods.

2.3.2 Classification of Strategies According to Correspondence Definitions

Different from the categorization by other relevant literatures (Ohlmann-Lauber and Schäfer, 2011; Qin *et al.*, 2016; Neuner *et al.*, 2016; Wunderlich *et al.*, 2016; Harmening, 2020), herein the existing methods of point cloud-based deformation analysis are classified according to their explicit or implicit definitions of correspondences which also indicate the deformation directions, as shown in Table 2.2. The correspondences defined in these methods may be realistic (e.g., feature-based methods), unrealistic (e.g., nearest neighbor-based methods used for highly rough surfaces), or potentially realistic (e.g., defined direction-based methods with appropriate definitions of deformation directions).

Among the above methods, the closest points between two point clouds are regarded as the correspondences in C2C and local ICP, which are the exemplary algorithms of nearest neighbor-based and local registration-based methods, respectively. C2M or M2M are based on the same principle but take the closest facet or edge in the triangulated mesh as the correspondence. The deformations calculated by this kind of correspondence, however, merely represent the minimum distance between two surfaces and may underestimate actual deformation magnitudes in variable surface topographies.

Table 2.2: Five categories of point cloud-based deformation analysis by the correspondence definition.

Methods	General principles	Representative algorithms
Nearest neighbor-based	Calculating EDs of the points in one point cloud (PC) to their nearest neighbors in another PC.	C2C, C2M (Cignoni *et al.*, 1998) M2M (Aspert *et al.*, 2002)
Feature-based	Calculating EDs of feature elements (e.g., points and planes) extracted and matched from two PCs or their converted 2D images.	F2S3 (Gojcic *et al.*, 2020b) CD-PB M3C2 (Zahs *et al.*, 2022) Image-based correlation (Travelletti *et al.*, 2014) Image-based feature points (Holst *et al.*, 2021)
Parameter-based	Calculating changes of parametric elements (e.g., normals of planes or centers of spheres) estimated by the parameterization of PC surfaces.	Geometric primitives (Yang *et al.*, 2021) B-spline surface (Harmening *et al.*, 2021)
Defined direction-based	Calculating EDs of constructed corresponding points from two PCs along defined directions (e.g., gravity direction or surface normal).	DoD (Lane *et al.*, 2003) M3C2 (Lague *et al.*, 2013) M3C2-EP (Winiwarter *et al.*, 2021)
Local registration-based	Performing a registration procedure locally on selected subsets of two PCs to derive the transformation matrix and displacement vectors.	Local ICP (Teza *et al.*, 2007)

The feature-based methods are capable of finding realistic corresponding elements, so as to output actual surface displacement vectors. These algorithms are sensitive to both in-plane and out-of-plane deformations. Nevertheless, the correspondences like identical points within point clouds are either not measured between two scans or cannot be readily identified without prior knowledge (Wunderlich *et al.*, 2016), especially when shape deformations or non-overlapping areas (i.e., the areas are not measured in all epochs) occur during geodetic monitoring. Besides, the uncertainties of these features used for a significance test are still challenging to be estimated. F2S3 introduces learned 3D feature descriptors instead of handcrafted feature points and establishes correspondences in the feature space. This neural network-based technique has dramatically improved the detection and matching of features in 3D point clouds from natural scenes. However, annotated training dataset incorporating ground-truth correspondences is needed, and smooth or highly deformed surfaces and surfaces with lots of repetitive structures are still challenging for this pipeline (Gojcic *et al.*, 2021). Correspondence-driven plane-based M3C2 (CD-PB M3C2) proposed by Zahs *et al.* (2022) extracts individual planar surfaces from point clouds by region growing-based segmentation and establishes corresponding planes between epochs via a random forest classifier based on defined plane parameters (as similarity measures). M3C2 distances are then calculated between corresponding planes, and lower uncertainties can be achieved by benefiting from these segmented planes (Zahs *et al.*, 2022). In addition, constructing correspondences in the converted 2D images from 3D point clouds based on digital image correlation (Travelletti *et al.*, 2014) or feature detection in hillshade images (Holst *et al.*, 2021) has also become an efficient way to derive dense displacement vectors. The converted image solely exploits the 3D coordinates of measured points, and thus its quality entirely relies on the point cloud (regarding surface geometries, point densities, etc.).

The parameter-based methods are able to estimate parametric changes. The types of these parametric changes depend on the definitions of estimated parameters, which can be angle differences between surface orientations in two epochs when parameters are direction vectors, real displacements of geometric primitives when parameters are their 3D positions, or spatial distances between two parameterized surfaces (e.g., when taking the control points in B-spline surfaces as the parameters (Harmening *et al.*, 2021)), etc. The choice of parameters to be estimated should mainly consider the requirements of monitoring tasks and application scenarios. Since the variances (or the variance-covariance matrices) of estimated parameters could be derived from the adjustment process, significance tests can be performed on these parametric changes.

In the defined direction-based methods, correspondences are constructed implicitly along defined directions that are assumed to be consistent with the deformation directions. For example, DEM of difference (DoD) simply calculates the vertical distances of corresponding pixels of the generated DEM from point clouds (Lane *et al.*, 2003). As one of the most commonly used algorithms for point cloud-based change detection, M3C2 adopts multi-scale to estimate the local surface normal and take it as the deformation direction. The correspondences are constructed by averaging the sub-clouds captured by a cylinder the axis of which is along the defined deformation direction (Lague *et al.*, 2013). A variant of standard M3C2 called M3C2-EP extends the estimation of the Level of Detection (i.e., LoDetection defined in Lague *et al.* (2013)) by error propagation that integrates the uncertainties of the measurements themselves and the registration of two scans to achieve a more appropriate LoDetection (Winiwarter *et al.*, 2021). Notably, these defined direction-based methods are developed mainly for quantifying distances between point clouds rather than the real displacements on monitored surfaces. Nonetheless, the derived distance values have the potential to agree with the magnitudes of actual movements, provided reasonable definitions of deformation directions with prior knowledge. Besides, the spatial resolution of quantified deformations (distances) by defined direction-based methods will not be limited by the spatial coverage and distribution of feature-based or parameter-based correspondences, which means the respective deformation calculation can theoretically be carried out for all measured points.

To explain the definitions of these correspondences and their implications in 3D point clouds more clearly, an intuitive schematic diagram is presented in Figure 2.6. Besides, Table 2.3 summarizes the main characteristics, advantages/disadvantages and applicable scenarios of the five types of point cloud-based deformation analysis methods.

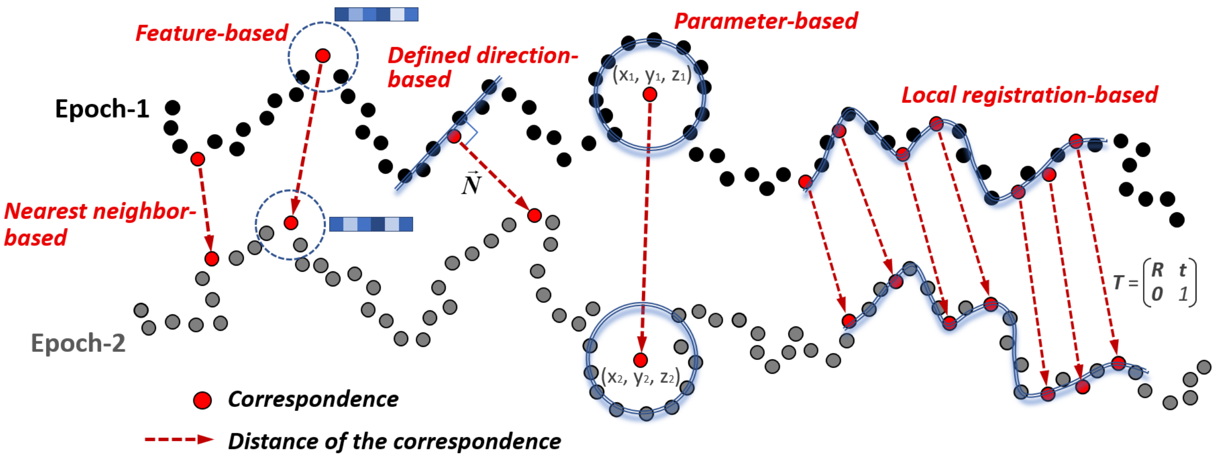


Figure 2.6: A schematic illustration of five kinds of correspondence definitions.

Table 2.3: Main characteristics, pros and cons of five types of point cloud-based deformation analysis classified by the correspondence definition.

Methods	Type of derived deformations	Advantages	Limitations	Applicable scenes
Nearest neighbor-based	Surface distances	Simple, fast and easy to be implemented	Deformation values are sometimes underestimated	Any kind of point clouds
Feature-based	Displacement vectors (fields)	Real displacements vectors can be derived	Low spatial resolution and incorrect correspondences in repetitive structures	Point clouds with distinct and corresponding features
Parameter-based	Parametric changes (e.g., distances, displacements, angles, etc.)	Parameters of interest and their uncertainties can be estimated	Prior knowledge for global or local parameterization is required	Point clouds with geometric primitives, or parametric changes are required
Defined direction-based	Surface distances	Introducing defined directions for deformation quantifications with high resolution	Deformation values are under- or over-estimated if defined directions are unrealistic	Any kind of point clouds with assumptions of deformation directions
Local registration-based	Displacement vectors (fields)	Dense displacement fields can be derived	Extracting corresponding rigid areas without prior knowledge is challenging	Point clouds with corresponding rigid areas

2.3.3 Challenges for Detecting Small Deformations under Complex Topographies and Deriving High-resolution Displacement Vectors

Natural like landslides or rockslides normally display high variability and roughness and incorporate both rigid and non-rigid areas. The acquired multi-temporal 3D point clouds from geomonitoring procedures may contain varying point densities or data missing in non-overlapping areas due to occlusion and changes (Lague *et al.*, 2013). Therefore, deformation analysis between these geomorphic surfaces requires accurate detection and quantification of deformations of different magnitudes under complex surface geometries and high measurement noise. Besides, calculations in non-overlapping areas should be avoided where spurious results may yield by the nearest neighbor-based methods. In addition, in the absence of prior knowledge of the stochastic model of point clouds, statistical significance tests can only be performed by empirically estimating the uncertainties of quantified deformations.

High measurement noise of point clouds will badly influence the calculated deformations using individual measured points or extracted feature points, thus probably producing inaccurate results, particularly for small deformations. In this case, the influence of Gaussian-distributed measurement noise can be reduced by averaging the selected neighboring points so as to decrease the uncertainties of calculated deformations. This skill of local averaging has been widely exploited in current advanced algorithms for point cloud-based deformation analysis like M3C2 (averaging points in the cylinder) and surface modeling-based methods (averaging points by estimating a surface model). However, surface modeling-based methods (e.g., B-spline surface (Harmening, 2020) and Poisson surface reconstruction (Kazhdan *et al.*, 2006)) still remain problems like under-fitting and optimal model selection, especially for complex and highly discontinuous point cloud surfaces.

Among the existing approaches mentioned in Section 2.3.2, only M3C2 and its variants, to the author's best knowledge, could preferably meet the above demands. Nevertheless, the standard M3C2 has a limited capacity to detect small deformations due to its assumption of the local planarity of compared surfaces. For the surfaces with variable geometries and high roughness, the uncertainties of quantified deformations may be overestimated and thus propagated into a higher LoDetection (Zahs *et al.*, 2022). For instance, when monitoring a sandy beach by laser scanning, high-frequency (daily or even hourly) data acquisition should be conducted to adapt to the deformation rate of the surface (Anders *et al.*, 2019). In such applications, the magnitudes of some captured deformations are small compared to associated LoDetection (overestimated by local roughness), resulting in failed detection of these small deformations. As an extension of standard M3C2, M3C2-EP allows lower LoDetection for rough surfaces. However, additional inputs (e.g., sensor accuracy and scan positions) are required, and the cylindrical neighborhood continuously used in M3C2-EP may still estimate inaccurate average positions in rough areas (Winiwarter *et al.*, 2021). Thereby, challenges still exist in M3C2 when detecting small deformations in highly rough surfaces.

The spatial resolution of quantified deformations herein refers not only to their spatial coverage and density but also to the capability to resolve the boundaries of deformation discontinuities. This is a critical quality feature to evaluate the performance of deformation analysis algorithms. Subsampled core points adopted in standard M3C2 to accelerate the computation undesirably decrease the resolution of derived deformations. Another variant tailored for the geomonitoring of a rock glacier, CD-PB M3C2, enables much lower uncertainties and could derive real displacements by using feature planes as correspondences (Zahs *et al.*, 2022). Like feature point-based approaches (e.g., F2S3), however, CD-PB M3C2 only computes deformations (displacements) in (assumed) rigid areas and takes the corresponding planes between epochs as the prerequisite. Therefore, the spatial resolution of quantified deformations by CD-PB M3C2 is reduced in non-planar or highly deformed areas.

The currently proposed M3C2 variants have indeed improved the performance of the distance quantification between two point cloud surfaces. For the deformed objects with predominately in-plane movements like a creeping glacier, however, using M3C2 or its variants can hardly derive realistic movements (i.e., displacement vectors) along the underlying surface. If dense and stable geometric features are available on highly variable surfaces, realistic correspondences across epochs can be established between extracted and matched feature points by hand-crafted feature descriptors, consequently yielding 3D displacement vectors (Gojcic *et al.*, 2020b; Gojcic *et al.*, 2021; Holst *et al.*, 2021). Nevertheless, the resolution of this displacement field totally depends on the density of successfully matched feature points, hence displacement vectors may be unable to be derived in flat or smooth areas, which reduces the deformation resolution in these areas.

To make a brief summary, although numerous versatile or tailored data-driven methods for efficient point cloud-based deformation analysis have been proposed, detecting small deformations under complex topographies and deriving high-resolution displacement vectors in lack of distinct features are still the challenges in TLS-based deformation monitoring.

3 Patch-based Segmentation

As mentioned in Section 1.2, natural objects with globally inhomogeneous and anisotropic deformations normally show local rigidity and a certain level of local consistency concerning stabilities as well as deformation directions in the deformation process. Besides, individual laser scanning measurements display local similarities among neighboring points with respect to orientations, standard deviations, reflectance intensities, etc., when the points are dense enough. Thereby, TLS point clouds capturing deformed surfaces can be spatially segmented into small and assumed-rigid patches benefiting from these physical characteristics, which are further employed to improve targetless registration of deformed point clouds and deformation analysis under complex topographies. In this chapter, assumptions and typical types of these rigid patches will be introduced, and approaches to the generation of each specific type of patch from 3D point clouds will be elaborated as well.

3.1 Implication of Rigid Patches in 3D Point Clouds

3.1.1 Local Rigidity and Local Consistency Assumptions

In most geodetic monitoring of natural scenes, parts of the scanned surface may remain stable while others move or deform over the monitoring epochs (Gojcic *et al.*, 2020b). Discontinuities hence arise at the boundaries between stable and unstable areas. The resulting deformations may present globally anisotropic directions and inhomogeneous magnitudes due to the inconsistencies of the deformation mechanism as well as the surface morphology of deformed objects, but locally show a certain level of regularity or even consistency (Gojcic, 2021). Furthermore, even for the surface with shape deformations, a local area within it can still be regarded as moving like a rigid body[1] as long as it is small enough (in the extreme case, it is an individual point). Herein, it is assumed that the surface of a deformed point cloud is comprised of small and locally rigid areas that do not cross the discontinuities of surface rigidity, and the points within these areas represent a rigid body rather than a significantly deformed part. In addition, when points within a small area show similarities regarding positions, orientations, reflectance intensities, colors, etc., consistencies with respect to deformation directions and measurement uncertainties probably exist within this area. Therefore, in this thesis, these small areas are formulated as *rigid patches* and imposed with local rigidity or/and local consistency assumptions.

Based on the local rigidity assumption, rigid patches can be segmented from the deformed point cloud and be further individually analyzed with respect to their stabilities during the deformation process. Besides, parameter estimation and local point cloud registration are possible to be utilized for these rigid patches to derive parametric deformations or displacement vectors (see Section 5.2 and Section 5.3, respectively). Based on the local consistency

[1] Rigid-body movements can be described by the change of spatial 6-degree of freedom (i.e., three rotation angles and three translation components) which can be integrated into a transformation matrix as shown in Equation (2.4).

assumption of deformed point clouds, calculations of deformation directions and empirical uncertainties of measured points are unified in respective patches, i.e., defined deformation direction and estimated standard deviation of any individual point can be shared within its corresponding patch. It should be noted that the rigid patches herein are all assumed locally rigid but not always assumed locally consistent. For instance, a larger rigid area may not show consistent deformation directions or consistent uncertainties of measured points, while small and planar (locally similar) patches are more likely to show local consistency.

3.1.2 Types of Rigid Patches

In this thesis, the concept of the rigid patch of point clouds is generalized with respect to its spatial scale, not limited to a small rigid area. The rigidity of patches means the relative position (distance) between any two points within the patch will not change during the deformation process, i.e., no shape deformation inside. According to this definition, Figure 3.1 shows five typical kinds of rigid patches in the 3D point clouds.

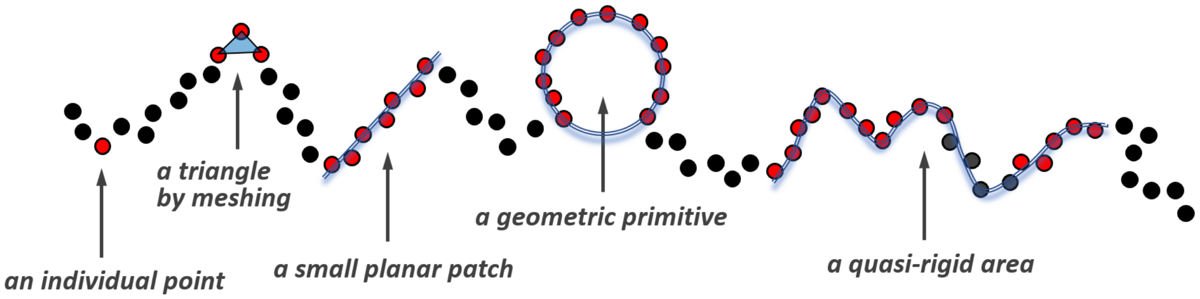


Figure 3.1: Typical types of rigid patches in 3D point clouds.

The extreme case of a patch is an infinitesimal area that converges to an individual point. The change of a point during a deformation process can only be rigid-body movement (only translations but no rotations) rather than shape deformation. Although single point-based deformation analysis is simple and easy to be conducted, correspondences of points between epochs and standard deviations are intractable in the absence of prior semantic and stochastic information of point clouds. In addition, when the point cloud contains high-level noise, the uncertainty of the calculated deformation of an individual point may be larger than the deformation value itself, leading to a failure of small deformation detection.

The minimum unit of rigid patches could be a triangle comprised of three non-collinear points, which can be generated from the triangulated irregular network (TIN)-based meshing process. This kind of patch enables the estimation of its local orientation and the interpolation of the interior of triangles which is not measured by the laser scanner. For the point cloud with high-level noise, however, these triangular patches unsatisfactorily display dramatic changes that worsen the following processing. Besides, associated uncertainties are not able to be empirically estimated from these triangles.

Taking more (> 3) neighboring points with local similarities (e.g., positions, orientations, etc.) together, plenty of small and planar patches can be formed and regarded as rigid areas individually. Discontinuities or boundaries in the point cloud will not appear in these small planar patches. Besides, empirical uncertainties of these patches incorporating more measurements are conveniently estimated by local modeling like plane fitting. This kind of small planar patch can be assumed not only locally rigid but also locally consistent due to similar geometric properties like positions and orientations of points within the patch. One of the efficient ways to generate such small planar patches is supervoxel-based over-segmentation[2] (see Section 3.2.1).

In some scenes, especially artificial scenarios, some simple objects keep rigid and only present translations or rotations when changes occur. These simple objects, such as geometric primitives, can also be regarded as rigid patches if their shapes keep unchanged. Efficient extraction and parameter estimation of geometric primitives (e.g., planes, spheres, cylinders, etc.) allow for a rigorous parameter-based deformation analysis on these patches. Automatic extraction of these geometric primitives can be easily realized by instance segmentation techniques with prior knowledge of their types and approximate parameters (see Section 3.2.2). Furthermore, checking whether the shape is deformed between epochs can be performed based on precisely estimated parameters, ensuring the rigidity prerequisite.

When these rigid patches get larger, they could be complex objects whose shapes are not deformed, or just large areas with arbitrary shapes and variable surfaces but keeping rigid. This local rigidity usually appears in the scene of natural geological phenomena like glaciers or rockslides, where fairly large areas (e.g., boulders or trees) merely show rigid-body movements instead of shape deformations. These large rigid patches with rich variations of surface orientations enable finding area-wise correspondences between epochs and performing a local registration technique to derive displacement vectors (see Section 5.3). Nevertheless, automatic, precise, and complete identification or extraction of these rigid areas with arbitrary shapes is still challenging when no prior knowledge is available. Quasi-rigid areas where the majority of points inside are rigid, herein, are introduced to represent such kind of rigid patch for the convenience of their efficient segmentation from 3D point clouds.

In order to estimate the local geometric and stochastic information of point clouds from rigid patches and utilize them in the following deformation analysis, only three typical patches, i.e., small planar patches, geometric primitives, and quasi-rigid areas, are discussed and analyzed in this thesis. In the following section, specific segmentation methods to obtain these three kinds of patches from 3D point clouds will be presented respectively.

3.2 Segmentation of Rigid Patches from 3D Point Clouds

3.2.1 Small Planar Patch Generation by Supervoxel-based Over-segmentation

As discussed in Section 3.1.2, point clouds representing the surfaces of natural objects can be divided into numerous small and nearly planar patches, of which points inside have similar properties. These locally consistent patches could be generated by means of segmentation or clustering techniques based on the point similarities of spatial distance, local surface orientation, associated radiometric values (e.g., RGB colors and reflectance intensities), etc. Consid-

[2] Over-segmentation is the process by which the objects or regions of interest segmented from the background are themselves segmented or fractured into subcomponents. It increases the chances that boundaries of importance are extracted at the cost of creating many insignificant boundaries.

ering the performance and computational efficiency of relevant algorithms, the supervoxel-based over-segmentation strategy is exploited to partition the 3D point cloud into small and locally planar patches, simultaneously preserving the boundaries and discontinuities of the surface (Gojcic, 2021; Lin *et al.*, 2018).

Supervoxels are the extension of superpixels[3] from 2D images to 3D point clouds (Papon *et al.*, 2013), and their generation considers not only color information (mainly used in superpixels) but also 3D geometric features. Compared to the voxels that partition 3D point clouds into equal-sized 3D grids (cubes), supervoxels could adapt to the surface variations with respect to local orientations and radiometric values, thus showing local consistency within each segment. Two mainstream strategies for the supervoxel generation from 3D point clouds will be introduced and utilized in this thesis.

The first strategy is based on seeding and local iterative clustering by using points' color and geometric features. The representative algorithm is Voxel Cloud Connectivity Segmentation (VCCS) algorithm proposed by Papon *et al.* (2013). VCCS employs a variant of *k*-means clustering for generating its labeling of points, with two important constraints:

(1) The seeding procedure of supervoxel clusters is carried out by partitioning the 3D space, which assures a uniform distribution of generated supervoxels according to the scene's geometry.
(2) The iterative clustering method enforces strict spatial connectivity of occupied voxels when considering points for potential clusters. Thus supervoxels cannot flow across boundaries which are disjoint in 3D space.

The pipeline of VCCS to generate supervoxels can be briefly summarized into four steps as follows, and more detailed processing steps can be referred to Papon *et al.* (2013).

(1) **Adjacency graph**: As a preliminary step, an adjacency graph is constructed for the voxelized point cloud, which can be achieved by searching the *k*d-tree of voxels. As for a given voxel, the centers of all 26-adjacent voxels are contained within $\sqrt{3}R_{voxel}$ (R_{voxel} denotes the voxel resolution used for the segmentation).

(2) **Spatial seeding**: A number of seed points are selected at the beginning and used to initialize supervoxels. As illustrated in Figure 3.2, the space is firstly divided into a voxelized grid with a chosen size R_{seed} that is much higher than R_{voxel}. Then the initial candidates for seeding are chosen by selecting the voxel (i.e., the voxel with a size of R_{voxel}) nearest to the center of each occupied seeding voxel (i.e., the voxel with a size of R_{seed}). To filter out the seeds caused by noise, the candidate seeds which are points isolated in space (outliers) must be removed. VCCS establishes a small search radius R_{search} around each seed, and deletes the seeds not having at least as many voxels as would be occupied by a planar surface intersecting with half of the search volume, which is shown by the green plane in Figure 3.2.

[3] Superpixels group similar pixels in the image into perceptually meaningful and regularly distributed small regions which conform to object boundaries (Papon *et al.*, 2013).

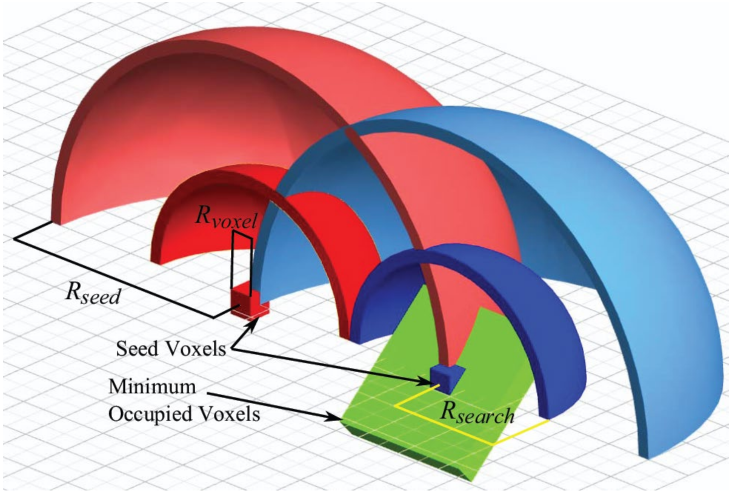


Figure 3.2: Seeding parameters and filtering criteria (Papon *et al.*, 2013). R_{seed} determines the distance between supervoxels, while R_{voxel} determines the resolution to which the cloud is quantized. R_{search} is used to determine if there are a sufficient number of occupied voxels to necessitate a seed.

(3) **Features distance measure**: Supervoxels generated from the over-segmentation of the 3D point cloud can be viewed as the result of voxel clustering based on the similarities of spatial distance, local surface orientation, and other properties. These similarities can be measured by a synthetic feature distance (SFD) that incorporates the normalized spatial distance, the difference of surface orientation, and the normalized color distance in CIELAB space (Luo, 2015) with respective weights. The points with shorter SFD to the seed point will be clustered into the corresponding supervoxel. The SFD is defined as

$$SFD(\boldsymbol{p}_i,\boldsymbol{p}_j) = w_d \frac{\left\|\boldsymbol{p}_i - \boldsymbol{p}_j\right\|_2}{R_{SV}} + w_n(1-\left|\boldsymbol{n}_{p_i} \cdot \boldsymbol{n}_{p_j}\right|) + w_c \frac{\left\|\boldsymbol{c}_{p_i} - \boldsymbol{c}_{p_j}\right\|_2}{m}, \tag{3.1}$$

where $SFD(\boldsymbol{p}_i, \boldsymbol{p}_j)$ is the defined SFD between the seed point $\boldsymbol{p}_i$ of one supervoxel and the point $\boldsymbol{p}_j$ to be clustered, $\boldsymbol{n}_{p_i}$ and $\boldsymbol{n}_{p_j}$ are the normal vectors of the two points, $\boldsymbol{c}_{p_i}$ and $\boldsymbol{c}_{p_j}$ are the CIELAB color values of the two points, w_d, w_n and w_c are the weight factors for spatial distance, orientation difference, and color difference respectively, R_{SV} denotes the approximate radius of the supervoxel and m is a constant for normalizing the CIELAB color distance. Notably, the color values are not always available in 3D point clouds. Thus, only the first two terms in Equation (3.1) are used for SFD calculation (i.e., $w_c = 0$) if radiometric information is not considered in the patch generation.

(4) **Flow-constrained clustering**: A local k-means clustering is used to assign voxels to supervoxels in an iterative way, considering their spatial connectivity. As illustrated in Figure 3.3, beginning at the voxel nearest the cluster center, it flows outward to adjacent voxels and computes the SFD from each of these to the supervoxel center by Equation (3.1). The supervoxel label will be set if the SFD of this voxel is the smallest, and its neighbors are added to the search queue for this label using the adjacency graph. Then it proceeds to the next supervoxel, so that each level outwards from the center is considered simultaneously for all supervoxels. It proceeds iteratively outwards until the edge of the search volume for each supervoxel (or no more neighbors left to be checked) is reached. Once the search of all supervoxel adjacency graphs is finished, centers of each

supervoxel cluster will be updated by taking the mean of all its constituents. This is done iteratively until the cluster centers stabilize or for a fixed number of iterations.

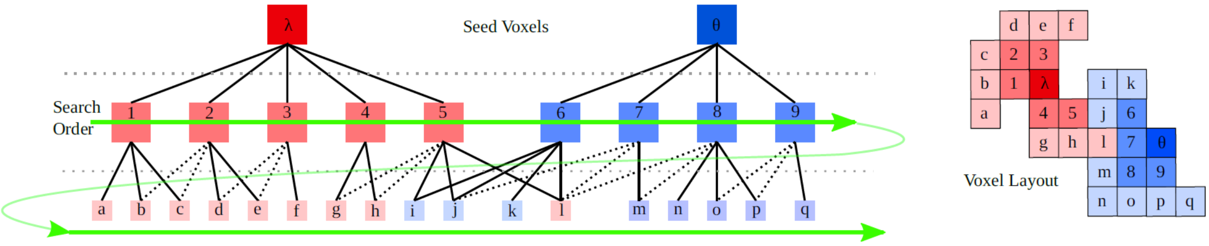


Figure 3.3: Search order for the flow constrained clustering algorithm (shown in 2D for clarity) (Papon *et al.*, 2013). Dotted edges in the adjacency graph are not searched, because the nodes have already been added to the search queue.

The second strategy is based on a fusion—exchange optimization (FEO) framework that formalizes supervoxel segmentation as a subset selection problem (Lin *et al.*, 2018; Xiao *et al.*, 2020). A representative algorithm is the better boundary-preserved supervoxel segmentation method proposed by Lin *et al.* (2018). This FEO-based strategy could produce supervoxels with adaptive resolutions and does not rely on the selection of seed points (while VCCS has an unsatisfactory performance on the point cloud with non-uniform resolutions and small structures). The pipeline of this method is briefly summarized as follows, while more detailed processing steps can be found in Lin *et al.* (2018).

(1) **Problem formulation**: Given a point cloud with N points, the partitioning of the point cloud into K_{rp} supervoxels can be regarded as a mapping from each point to a label of a supervoxel. The goal of the optimization framework is to determine K_{rp} representative points to minimize the sum of the dissimilarity distances between each point and its representative point, which can be formalized as (Lin *et al.*, 2018)

$$\begin{aligned} &\min_{\{z_{ij}\}} \sum_{i=1}^{N}\sum_{j=1}^{N} z_{ij} D(\boldsymbol{p}_i, \boldsymbol{p}_j) \\ &s.t.\ \ z_{ij} = \{0,1\},\ \ \forall\, i, j;\ \ \sum_{i=1}^{N} z_{ij} = 1,\ \ \forall\, j;\ \ C(Z) = K_{rp} \end{aligned}, \tag{3.2}$$

where $D(\boldsymbol{p}_i, \boldsymbol{p}_j)$ is a distance metric to measure the dissimilarity between point $\boldsymbol{p}_i$ and point $\boldsymbol{p}_j$, which can use the expression of SFD in VCCS (i.e., Equation (3.1)), $z_{ij} \in \{0,1\}$ are unknown binary variables and $z_{ij} = 1$ means that 1) $\boldsymbol{p}_i$ is a representative point and 2) $\boldsymbol{p}_j$ is a non-representative point that is represented by $\boldsymbol{p}_i$. Here, $\sum_{i=1}^{N} z_{ij} = 1$ is set to guarantee that each point $\boldsymbol{p}_j$ in the point cloud is represented by exactly one supervoxel. The function $C(\cdot)$ is used to count the number of representative points during the optimization process, and it is defined as (Lin *et al.*, 2018)

$$C(Z) = \sum_{i=1}^{N} I\left(\sum_{j=1}^{N} z_{ij}\right), \tag{3.3}$$

where $I(\cdot)$ is an indicator function defined as

$$I(x) = \begin{cases} 0, & if\ x = 0 \\ 1, & if\ x \neq 0 \end{cases}. \tag{3.4}$$

(2) **Optimization method**: To obtain the optimal segmentation, an efficient energy descent method is employed that leverages local information of the point cloud. This method does not require the initialization of seed points. An energy function $E(Z)$ is constructed considering the following relaxation of Equation (3.2) denoted as (Lin *et al.*, 2018)

$$\begin{aligned} \min\ E(Z) &= \sum_{i=1}^{N}\sum_{j=1}^{N} z_{ij} D(\boldsymbol{p}_i, \boldsymbol{p}_j) + \lambda \left| C(Z) - K_{rp} \right| \\ s.t.\ z_{ij} &= \{0,1\},\ \forall\, i,j;\ \sum_{i=1}^{N} z_{ij} = 1,\ \forall\, j. \end{aligned} \tag{3.5}$$

The first term in Equation (3.5) ensures that the selected representative points can effectively approximate the collection of all points, while the second term constrains the number of representative points to be close to K_{rp}. A regularization parameter, λ ($\lambda > 0$), is introduced to set the trade-off between two terms. Then, the FEO framework including fusion and exchange-based minimization is conducted to solve the energy function.

(3) **Fusion-based minimization**: An adaptive strategy to automatically evaluate the value of λ is adopted. First, λ is set to a small value λ_0, and Equation (3.5) is solved to obtain the initial representative points. Afterward, the value of λ is increased in each step and the representative points are updated through solving Equation (3.5) with the new λ value. Considering that a supervoxel is a local region of the point cloud surface, local information is used to accelerate the optimization algorithm. As illustrated in Figure 3.4, an energy minimization method is adopted based on a bottom-up fusion strategy. More details of this step are described in Lin *et al.* (2018). The intuition behind the optimization is that it preferentially aggregates the points situated in the smooth areas, which facilitates the supervoxels to avoid crossing the boundaries (Lin *et al.*, 2018).

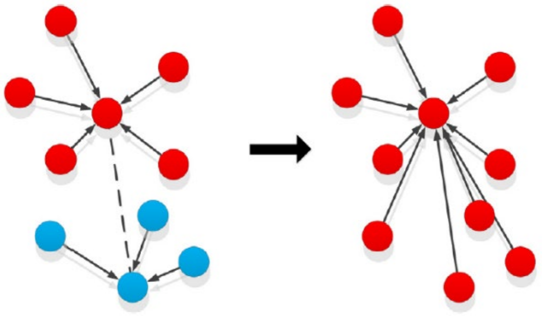
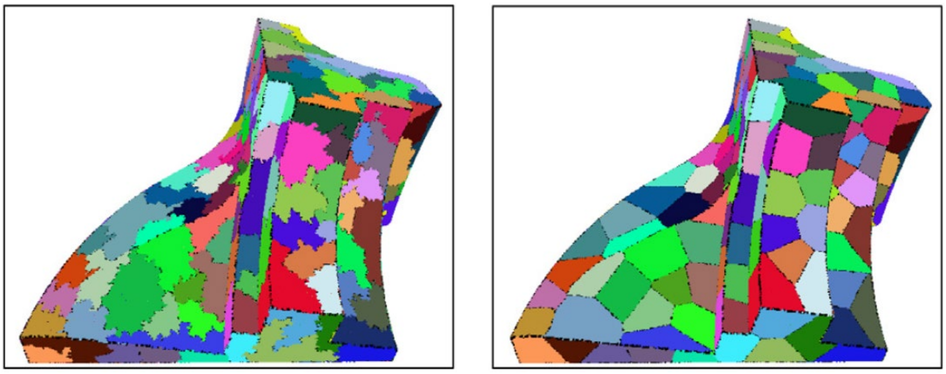


Figure 3.4: An example of the fusion operation for two adjacent representative points (Lin *et al.*, 2018). The merging operation will be performed if the energy function is reduced.

(4) **Exchange-based minimization**: After K_{rp} representative points are generated through the above fusion-based operation, the energy function $E(Z)$ can be continuously optimized by assigning each non-representative point to the representative point with the lowest dissimilarity distance from it. Specifically, for a pair of adjacent points $\boldsymbol{p}_i$ and $\boldsymbol{p}_j$, if the condition $D(\boldsymbol{p}_i, \boldsymbol{r}_j) < D(\boldsymbol{p}_i, \boldsymbol{r}_i)$ is satisfied, where $\boldsymbol{r}_i$ and $\boldsymbol{r}_j$ are the representative

points of $\boldsymbol{p}_i$ and $\boldsymbol{p}_j$, respectively, then the energy function $E(Z)$ can be reduced by assigning $\boldsymbol{p}_i$ to $\boldsymbol{r}_j$. The energy function is constantly reduced until no further improvement can be achieved. Figure 3.5 illustrates that this exchange-based minimization is beneficial for preserving supervoxels' boundaries in a better way and generating more regular shapes. Further elaboration on this step can be found in Lin *et al.* (2018).



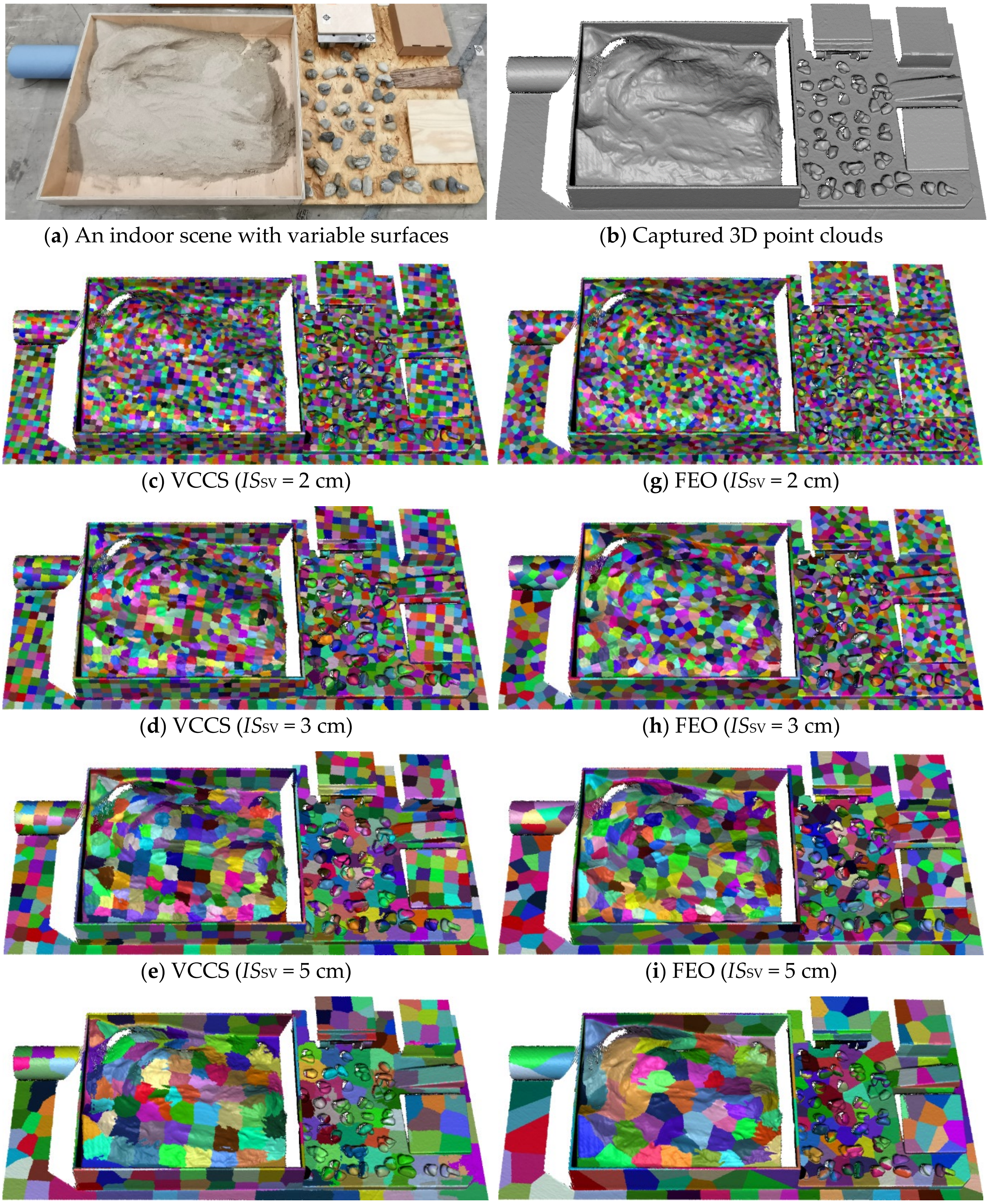
(**a**) Before exchange (**b**) After exchange

Figure 3.5: Illustration of the exchange-based minimization process (Lin *et al.*, 2018).

Utilizing the above two strategies, the size of supervoxels is capable of adapting to the surface variation of point clouds automatically as long as an appropriate initial size (or resolution) of supervoxels (*IS*sv) is set by the user. For VCCS, the parameter R_{seed} can be regarded as *IS*sv, while the number of representative points K_{rp} (i.e., the expected number of supervoxels) in the FEO-based method can be calculated by *IS*sv (Lin *et al.*, 2018).

Figure 3.6 shows the generated supervoxels of a 3D point cloud (representing an indoor scene with high surface variations and structures of different sizes) under different settings of *IS*sv by VCCS (Papon *et al.*, 2013) and the FEO-based method (Lin *et al.*, 2018). In Figure 3.6, the boundaries between objects are basically preserved by both methods when *IS*sv < 5 cm. However, too small *IS*sv may cause inefficient points within the patch (supervoxel), thus affecting the following processing based on patch points. In contrast, too large *IS*sv may fail to separate the boundaries or discontinuities for small structures (see the areas with small stones in Figure 3.6(f) and (j) when *IS*sv = 8 cm). Generally, FEO performs better than VCCS on boundary preservation, especially near the areas with noise and high roughness. In contrast, VCCS enables more homogeneous sizes and shapes of generated supervoxels (even close to equal-sized voxels in planar areas). Nevertheless, VCCS is more sensitive to noise and varying point densities (Lin *et al.*, 2018).

To achieve a trade-off that could guarantee the point quantity in patches as well as separate boundaries and discontinuities well, two empirical ways to determine the initial resolution of supervoxels are suggested: (1) Using twenty times (or larger for planar surfaces) the average point spacing of the point cloud to ensure enough points within each patch; (2) Taking the double size of the smallest structures in the scene provided sufficient points (i.e., high spatial resolution) in these areas. The final size of generated supervoxels may be smaller than the initial resolution in rough areas or larger in smooth areas (Lin *et al.*, 2018). For the indoor scene in Figure 3.6, an *IS*sv around 3 cm (i.e., twenty times the average point spacing) can perceptually produce a better patch segmentation result (see Figure 3.6(d) and (h)).



(**a**) An indoor scene with variable surfaces

(**b**) Captured 3D point clouds

(**c**) VCCS (IS_{SV} = 2 cm)

(**g**) FEO (IS_{SV} = 2 cm)

(**d**) VCCS (IS_{SV} = 3 cm)

(**h**) FEO (IS_{SV} = 3 cm)

(**e**) VCCS (IS_{SV} = 5 cm)

(**i**) FEO (IS_{SV} = 5 cm)

(**f**) VCCS (IS_{SV} = 8 cm)

(**j**) FEO (IS_{SV} = 8 cm)

Figure 3.6: Generated supervoxels under different initial sizes (IS_{SV}) by two strategies.

The additional advantage of introducing supervoxel to laser scanner-based deformation analysis is that the points in one supervoxel are correlated to some degree due to their short spatial distances, subequal incident angles, and similar surface properties. Hence, the correlation in point clouds from the laser scanning process could be considered in the local modeling of small planar patches. It should be noted that the surface properties can be partially reflected on the radiometric values (e.g., reflectance intensity or RGB color) along with surface measurements. This radiometric information could also be considered in the generation of supervoxels, especially for the surface comprised of various materials. This thesis merely concerns geometric information for patch generation, making the proposed methods more general and applicable to point clouds acquired from different sensors.

If the correlation between the points within one patch (supervoxel) is considered in the uncertainty estimation of patch models (e.g., centroids or fitted planes), the number of points n_{SV} in the patch should be replaced by the effective number of measurements n_{eff} in relevant calculations, which can be defined as (Taubenheim, 1969)

$$n_{eff} = \frac{n_{SV}}{1+2\cdot\sum_{i=1}^{n_{SV}-1}\left(\frac{n_{SV}-i}{n_{SV}}\cdot K(i)\right)}, \tag{3.6}$$

where *K(i)* describes the correlation coefficient between two measurements in a supervoxel. To simplify the computation of n_{eff}, a constant empirical value of spatial correlations K for TLS point clouds could be adopted to replace *K(i)* in Equation (3.6), resulting in a simplified and approximate n'_{eff} as

$$n'_{eff} = \frac{n_{SV}}{1+K\cdot(n_{SV}-1)}. \tag{3.7}$$

This empirical K can be seen as an approximate average of *K(i)* in a local area. The choice of K ($K \in [0,1]$) for a patch could roughly take into account the point spacing, the measuring distance, the size of the patch (supervoxel), etc. (Kauker *et al.*, 2016; Kauker and Schwieger, 2017). For example, a higher K can be considered for a high point density or a small patch size. If no correlation is assumed in the point clouds (i.e., *K(i)* = 0), the effective number of measurements equals the number of all points involved in the patch modeling, which agrees with the theoretical fact. Figure 3.7 illustrates the relationship between n'_{eff} and n_{SV} based on Equation (3.7) with different K settings.

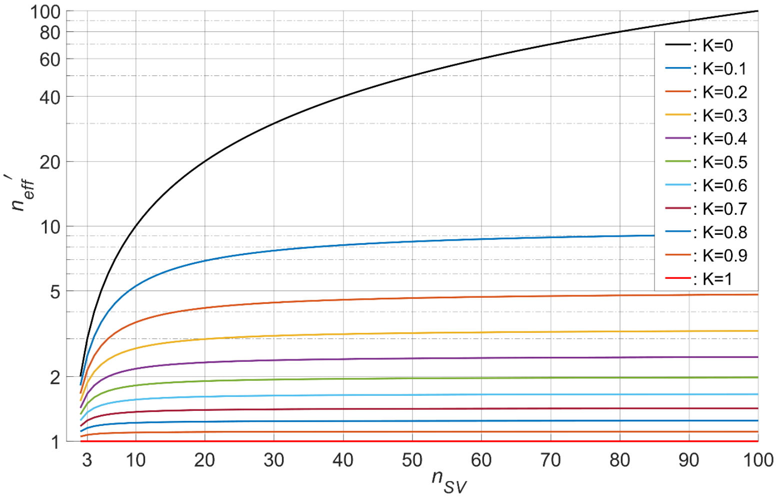


Figure 3.7: The relationship between n'_{eff} and n_{SV} with different K within the patch. For a clearer comparison, the vertical axis is on a logarithmic scale.

As shown in Figure 3.7, the higher the correlation between points in a patch, the less the effective number of measurements, leading to higher uncertainties of relevant estimates.

Furthermore, the effective number of measurements always equals one when points in the patch are fully correlated (i.e., $K(i) = 1$). Additionally, when n_{SV} is larger than a certain value, n'_{eff} ($K > 0$) tends to converge without much increase (e.g., $n_{SV} > 50$ when $K > 0.1$). Therefore, high correlations between measurements significantly influence the uncertainty estimations.

3.2.2 Geometric Primitive Extraction by Region Growing and RANSAC

A geometric primitive is a basic object (e.g., a line, a surface, or a volumetric shape) whose position, orientation, dimension, and shape can be described by a small set of parameters. Table 3.1 lists the typical types of geometric primitives that may exist in the real world and their possible parametric changes when they keep rigid in the deformation process. Provided the prior knowledge (including types, number, approximate parameters, etc.) of geometric primitives within the scene captured by laser scanning, an instance segmentation can be performed on the point cloud and extract the point sets lying on the surfaces of existing geometric primitives.

Table 3.1: Typical geometric primitives and their key geometric parameters.

Geometric primitives	Equations (simple forms, *a*, *b*, *c*, *d* are coefficients)	Key geometric parameters	Possible parametric changes (only representing rigid-body movements)
Plane	$ax + by + cz = d$	Normal vector	Orientation change of the normal vector
Cylinder	$x^2 + y^2 = a^2$ (when the cylinder axis along Z-axis)	Orientation of the cylinder axis and radius	Orientation change of the cylinder axis
Sphere	$(x-a)^2 + (y-b)^2 + (z-c)^2 = d^2$	Sphere center and radius	Movement of the sphere center
Ellipsoid	$\frac{x^2}{a^2} + \frac{y^2}{b^2} + \frac{z^2}{c^2} = 1$	Ellipsoid center and long/middle/short semi-axes	Movement of the ellipsoid center and orientation change of three semi-axes
Cone	$\frac{x^2 + y^2}{a^2} = z^2$ (when the cone axis along Z-axis)	Orientation of the cone axis, cone angle and cone apex	Movement of the cone apex and orientation change of the cone axis
Hyperboloid	$\frac{x^2}{a^2} + \frac{y^2}{b^2} - \frac{z^2}{c^2} = \pm 1$	Orientation of the rotation axis and the focus	Movement of the focus and orientation change of the rotation axis
Paraboloid	$\frac{x^2}{a^2} \pm \frac{y^2}{b^2} = z$	Orientation of the rotation axis and the focus	Movement of the focus and orientation change of the rotation axis

There are numerous tailored methods of instance segmentation aiming to segment different objects with semantic labels from 3D point clouds. Especially with the development of computer vision and computer graphics, plenty of geometry-based or neural networks-based instance segmentation algorithms for extracting geometric primitives from 3D point clouds have been proposed and applied in recent years (Xu *et al.*, 2017; Tran *et al.*, 2018; Honti *et al.*, 2022; Li *et al.*, 2019; Xia *et al.*, 2020). Based on the prior knowledge of geometric primitives, especially the types and approximate parameters, a simple and effective coarse-to-fine segmentation framework tailored to extract multiple geometric primitives is proposed in this thesis. This framework solely utilizes the geometric information of point clouds and mainly consists of 3D region growing (for coarse segmentation) and RANSAC (for fine segmentation), as shown in Figure 3.8.

In this framework, the preprocessed point cloud (after subsampling and outlier removal) is first segmented into different clusters by 3D region growing (see Appendix A.1 for algorithm details). The geometric primitives can be included in these clusters under an appropriate parameter tuning. Afterwards, a screening procedure is conducted based on the calculated geometric features or parameters of these clusters, including spatial dimensions, the distribution of point curvatures, linearity, planarity, scattering, anisotropy, omnivariance, etc. (cf. Weinmann *et al.* (2015) for details). The clusters meeting the conditions (from the prior knowledge) of specific geometric primitives will be preserved and further segmented finely by RANSAC (see Appendix A.2 for algorithm details) with respective model and parameter settings. The points belonging to specific primitives are finally extracted and used for the following rigorous parameter estimation (see Section 5.2.1).

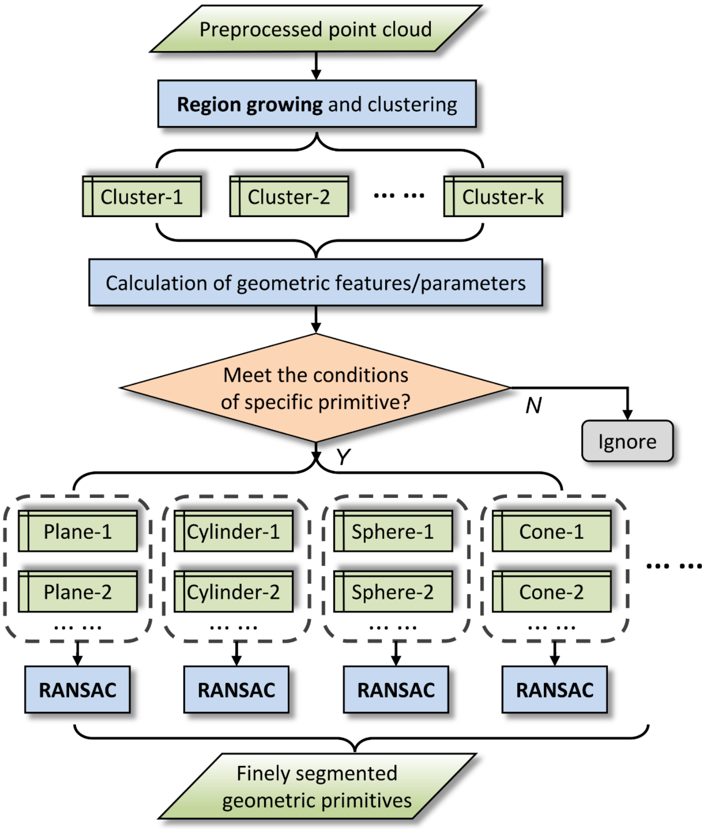

Preprocessed point cloud
Region growing and clustering
Cluster-1
Cluster-2
……
Cluster-k
Calculation of geometric features/parameters
Meet the conditions of specific primitive?
N
Ignore
Y
Plane-1
Plane-2
Cylinder-1
Cylinder-2
Sphere-1
Sphere-2
Cone-1
Cone-2
……
RANSAC
RANSAC
RANSAC
RANSAC
Finely segmented geometric primitives

Figure 3.8: Proposed framework for extracting multiple geometric primitives from 3D point clouds.

It should be noted that the initial coarse segmentation by region growing works well only on separating the primitive with smooth surfaces and discontinuous boundaries, like planes, spheres, cylinders, etc. Geometric primitives with sharp edges, like cuboids, will be segmented into several individual parts (planes). For such primitives, more specifically tailored segmentation methods should be developed (e.g., combining the initially segmented parts into a primitive randomly in a local neighborhood and finding the most satisfactory one). In particular, the patches generated via supervoxel-based over-segmentation (see Section 3.2.1) can partially or entirely occupy the existing planes in the point cloud with relatively larger IS_{SV} settings, thus regarded as plane primitives. However, in variable scenes with significantly different-sized planes, a large plane (as one geometric primitive) may be split into two or more parts by generated patches due to the constraint of supervoxel size in the generation process. In contrast, 3D region growing will not be influenced by the size of planes and thus enables the completeness of extracted geometric primitives.

Unlike small planar patches that are assumed rigid in all monitoring epochs, the rigidity of segmented geometric primitives should be further validated by calculating and comparing associated shape parameters (e.g., the radius of spheres or cylinders) between the correspondences (if exist) across different epochs. If the difference of associated parameters is within a small threshold, the extracted primitives can be regarded as rigid. This estimation procedure of primitive parameters will be given in Section 5.2.1.

3.2.3 Quasi-rigid Area Generation by a Segmentation-and-validation Strategy

In complex natural scenarios, precisely and automatically detecting the irregular boundaries between rigid and non-rigid areas from the deformed point clouds with few apparent discontinuities is quite challenging. Consequently, extracting complete rigid areas directly from two-epoch point clouds without prior knowledge is still difficult. These larger rigid areas typically exhibit irregular shapes and distributions, making parameterization processes like geometric primitives impractical for identifying corresponding areas in another epoch and verifying whether shape deformations occur inside (through parameter comparison). Nevertheless, when these locally rigid patches exist in the form of individual objects (e.g., boulders and trees on a landslide, or vehicles on the road), it is possible to extract them completely as individual patches by means of instance segmentation techniques (Walicka and Pfeifer, 2022; Himmelsbach *et al.*, 2010). Besides, assumed quasi-rigid areas could be manually or semi-automatically segmented with prior knowledge with respect to their locations and properties (e.g., the location and boundary of a rocky surface in a deformed landslide are known in point clouds). Then hand-crafted or learned geometric features of these segmented quasi-rigid areas (patches) can be computed and compared to those in other epochs so as to find their corresponding areas (patches) and assess their rigidity. This thesis mainly takes individual objects as an example of larger rigid areas to illustrate the concept and workflow of the point cloud-based deformation analysis based on rigid patches with irregular shapes.

As mentioned in Section 3.2.2, numerous instance segmentation methods have been proposed and specifically tailored to extract individual objects from point clouds. These geometry-based or neural networks-based segmentation strategies could be also employed to obtain assumed rigid areas from deformed point clouds. Since the investigation of advanced instance segmentation methods is not the focus of this thesis, a segmentation-and-validation strategy in a fast and low complexity manner for detecting quasi-rigid areas between epochs is presented herein. The segmentation part allows for utilizing flexible combinations of classical segmentation and 3D clustering algorithms (e.g., *k*-means clustering (Lavoué *et al.*, 2005), Euclidean clustering (Rusu, 2009), Density-Based Spatial Clustering of Applications with Noise (DBSCAN, Ester *et al.*, 1996), etc.) to extract the points lying on individual objects which are potentially (quasi-) rigid. The validation part is realized by computing the correlation between the constructed area-wise geometric feature descriptors of segmented patches in different epochs. If the correlation is beyond a certain threshold, the segmented patch can be regarded as quasi-rigid (i.e., the major parts of the patch are rigid). Thereby, the rigidity validation of patches with arbitrary shapes is conducted in a feature space instead of a parametric space adopted by geometric primitives. Inspired by the Point Feature Histogram (PFH) proposed by Rusu *et al.* (2008), in this thesis, an area-wise geometric feature descriptor of the segmented patch is constructed by using a supervoxel-based PFH.

Although representative geometric features like surface normals and curvature estimates are extremely fast and easy to compute, they cannot capture too much detail of the geometry around a specific point or of a selected patch. As a result, most scenes may contain many points with the same or quite similar feature values, thus reducing their informative characteristics (Rusu, 2009). The standard PFH descriptor is to encode the geometric features of a query point's *k*-neighborhood merely using the points in the defined neighborhood and their estimated surface normals. Feature values (spatial relationships) between all pairs of points in the neighborhood are computed and represented by a multi-dimensional histogram which is invariant to the 6 degrees of freedom (positions and orientations) of the underlying surface (Rusu *et al.*, 2008). It should be emphasized that the standard PFH is utilized to describe 3D geometric features of a single query point, thus only deriving point-wise feature descriptions. In this thesis, an area-wise geometric feature descriptor based on the calculation of standard PFH is proposed and used for (1) searching the corresponding patches between epochs and (2) validating the rigidity of segmented patches qualitatively.

To cope with the influence of non-uniform point densities as well as speed up the feature computation, PFH calculations are modified to be performed on the centroids of over-segmented supervoxels (generated by the methods in Section 3.2.1) of larger segmented patches (i.e., potentially quasi-rigid areas). The normal vector of each centroid is estimated by the points within the associated supervoxel. It should be emphasized that the segmented supervoxels of these larger patches are merely employed to calculate their feature descriptors, not involved in their segmentation from the point cloud.

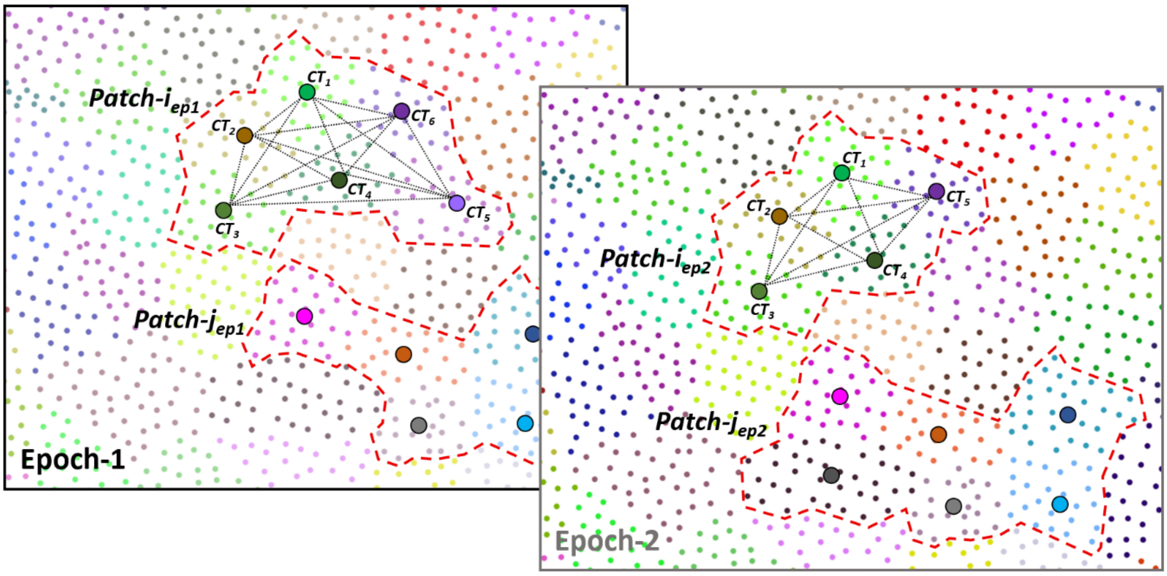


Figure 3.9: The construction of area-wise geometric feature descriptors for segmented patches based on the centroids of over-segmented supervoxels.

As demonstrated in Figure 3.9, potentially (quasi-) rigid patches (boundaries are highlighted by red dashed lines) could be extracted from two-epoch point clouds by automatic segmentation/clustering or manual operations with prior knowledge, and these areas are only represented by the centroids ($\boldsymbol{CT}_i$, $i = 1, 2, \ldots, k$) of generated supervoxels (colored points). Then the areal PFH descriptor of each patch is computed as a histogram of geometric relationships between all pairs of centroids within the patch (highlighted by dotted lines between centroids), thus yielding a pair number of $\frac{k(k-1)}{2}$ with k supervoxels in a patch.

The spatial relationship between a pair of centroids is computed based on a hand-crafted local frame. As illustrated in Figure 3.10, a local orthogonal coordinate frame ***CT-uvw*** at one of the centroids is defined based on the positions (***CT***$_i$ and ***CT***$_j$) and normals ($\boldsymbol{n}_i$ and $\boldsymbol{n}_j$) of two centroids. Three axes (***u***, ***v***, and ***w***) of ***CT***$_i$ are calculated by Equation (3.8) (Rusu, 2009).

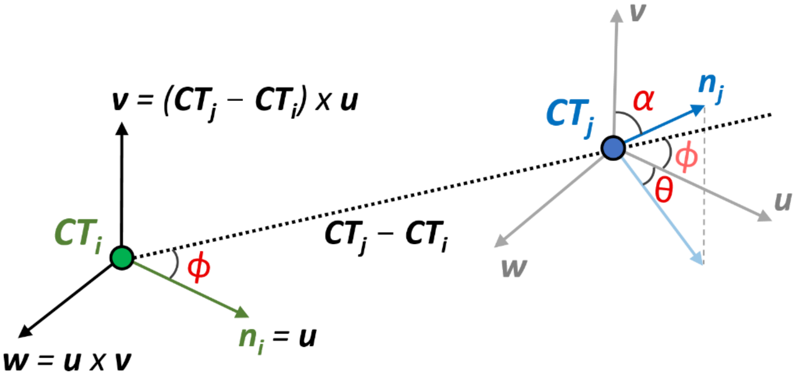


Figure 3.10: The computation of the spatial relationship between two centroids of two supervoxels within a patch (Rusu, 2009).

$$\begin{aligned} \boldsymbol{u} &= \boldsymbol{n}_i \\ \boldsymbol{v} &= \boldsymbol{u} \times \frac{\boldsymbol{CT}_j - \boldsymbol{CT}_i}{\left\| \boldsymbol{CT}_j - \boldsymbol{CT} \right\|_2} \\ \boldsymbol{w} &= \boldsymbol{u} \times \boldsymbol{v} \end{aligned} \tag{3.8}$$

Using the above ***CT-uvw*** frame, the relative difference between any two centroids and their associated normal vectors is depicted as three angular features by (Rusu, 2009)

$$\begin{aligned} \alpha &= \arccos(\boldsymbol{v} \cdot \boldsymbol{n}_j), \\ \phi &= \arccos\left(\boldsymbol{u} \cdot \frac{\boldsymbol{CT}_j - \boldsymbol{CT}_i}{\left\| \boldsymbol{CT}_j - \boldsymbol{CT} \right\|_2} \right), \\ \theta &= \arctan\left(\frac{\boldsymbol{w} \cdot \boldsymbol{n}_j}{\boldsymbol{u} \cdot \boldsymbol{n}_j} \right). \end{aligned} \tag{3.9}$$

The derived α, ϕ and θ are angles between vectors and are normalized to the same interval on the trigonometric circle. These angular features could not only capture the surface variations by taking into account all the interactions between estimated normals, but also cope well with different point densities and noise levels presented in the patch (Rusu, 2009). To create the final PFH descriptor for the segmented patch, the set of normalized angular features (α, ϕ, θ) of all pairs is individually binned into a 3-dimensional feature space. Each dimension (representing one angular feature) of the space is divided into b equal subintervals, thus resulting in b^3 subspaces to be binned. Then the filled 3D feature space is converted into a one-dimensional histogram that takes the number of feature occurrences in each subspace as the frequency. The horizontal axis of the histogram is the ordered arrangement of the subspace index, thus creating a histogram with b^3 intervals in a fully correlated space, as presented in Figure 3.11 with $b = 5$. The frequency sequence of the histogram is used as the areal PFH descriptor for a segmented patch in the point cloud.

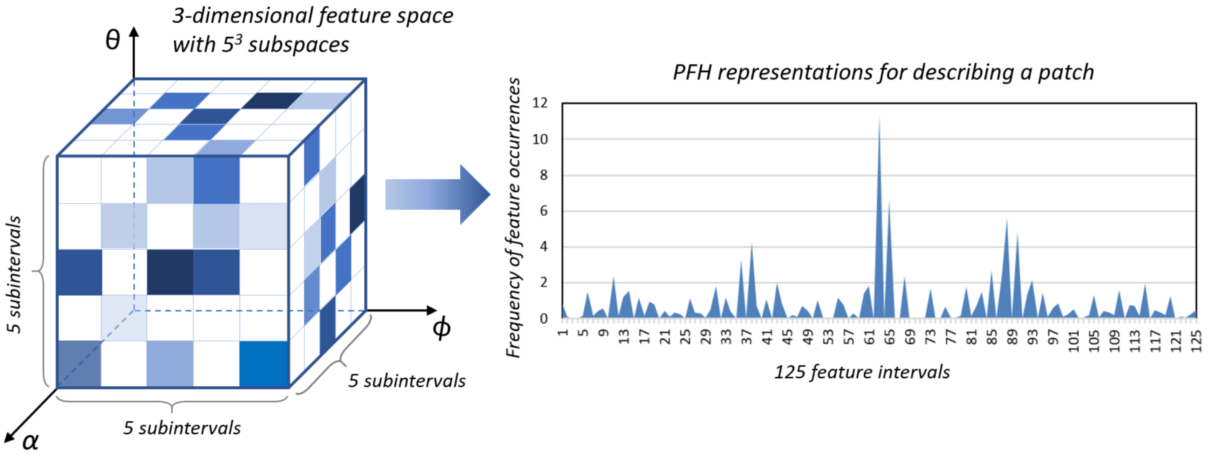


Figure 3.11: Creating an areal PFH descriptor from a 5×5×5 feature space.

Based on the constructed feature descriptors of all segmented patches that are potentially rigid, correspondences across epochs can be established by

(1) calculating the correlation coefficients of descriptor sequences $\rho(PFH)$ (by

$$\rho(PFH) = \frac{\sum_{i=1}^{b^3}\left(F_{epoch\_1,i} - \overline{F}_{epoch\_1}\right)\left(F_{epoch\_2,i} - \overline{F}_{epoch\_2}\right)}{\sqrt{\sum_{i=1}^{b^3}\left(F_{epoch\_1,i} - \overline{F}_{epoch\_1}\right)^2} \cdot \sqrt{\sum_{i=1}^{b^3}\left(F_{epoch\_2,i} - \overline{F}_{epoch\_2}\right)^2}}, \qquad (3.10)$$

where F_{epoch_1} and F_{epoch_2} are areal PFH descriptors of segmented patches in two epochs, respectively) between all possible combinations in two epochs and

(2) finding the patch pair with the highest correlation via brute-force search.

(3) If the highest correlation coefficient is larger than a defined threshold (e.g., > 0.8), these two patches will be regarded as correspondence as well as quasi-rigid across epochs.

In order to generalize the application of the above areal geometric feature-based validation strategy applied for identifying arbitrary rigid areas in point clouds, a manual and careful segmentation considering reliable prior knowledge with respect to surface rigidity is recommended to obtain larger quasi-rigid areas. Then the following validation procedure is performed on these manually segmented patches. This manual operation is prone to achieve a trade-off between surface variation and rigidity (i.e., patches should be rigid and as variable as possible), which is more beneficial to employ local registration techniques for deriving displacement vectors (see Section 5.3).

4 Targetless Registration of Deformed Point Clouds

Based on the segmented small planar patches, a fully automatic supervoxel-based registration method for point clouds with partially unstable areas is proposed to cope with the challenges discussed in Section 2.2.3. This approach does not require artificial targets or extracted feature points and is easy to be implemented due to the absence of surface meshing or complicated modeling. The content of this chapter is mainly based on the author's publication: Yang Y, Schwieger V. **Supervoxel-based targetless registration and identification of stable areas for deformed point clouds** in the *Journal of Applied Geodesy*, 2022 (Yang and Schwieger, 2022).

Section 4.1 and Section 4.2 present the methodology and experimental evaluations of the proposed method, respectively. Performance comparisons with other registration algorithms are given as well. Section 4.3 discusses the main issues of the pipeline as well as its possible investigations for further improvements.

4.1 Supervoxel-based Targetless Registration Method

The global flowchart of the proposed targetless registration algorithm is shown in Figure 4.1. Starting with coarsely registering the preprocessed two-epoch point clouds using the 4PCS-based approaches (Aiger *et al.*, 2008; Theiler *et al.*, 2014), supervoxel-based over-segmentation (see Section 3.2.1) is employed to divide both the reference point cloud in epoch-1 (PC-1) and the deformed point cloud in epoch-2 (PC-2) into small and locally planar patches, namely supervoxels. The centroid and boundary points of each supervoxel are then calculated and used to construct the point-to-plane correspondences. The significantly deformed supervoxel of which the distance of the correspondence exceeds the distance threshold (DT) in the current iteration will be removed, merely remaining the stable supervoxels to perform the point-to-plane ICP (Park and Subbarao, 2003). The entire process above is conducted iteratively until the DT decreases to the local minimum detectable deformation (LMDD) determined by a confidence interval based on an empirical stochastic model of the point cloud. A four-stage update of DT in each iteration is triggered by the corner change of the bounding box of the transformed PC-2. Details of the pipeline are presented in the following subsections.

4.1.1 Point Cloud Preprocessing and Coarse Registration

The outliers in point clouds caused by occlusions or sensor imperfections do not contribute to the correspondence generation between two scans. Thus, these spurious points are supposed to be eliminated in the preprocessing step. Herein, statistical outlier removal (SOR) filter is adopted to clean the raw data (Rusu *et al.*, 2008; Balta *et al.*, 2018). In addition, to avoid the local optimization (i.e., local minimum in the registration problem) of ICP, two-epoch scans in arbitrary coordinate systems need to be coarsely aligned, which can be achieved by the 4PCS-based algorithms. The quality of coarse registration may affect the accuracy of the final estimation, as the approximate coarse registration error can be used as the initial DT (see Section 4.1.4).

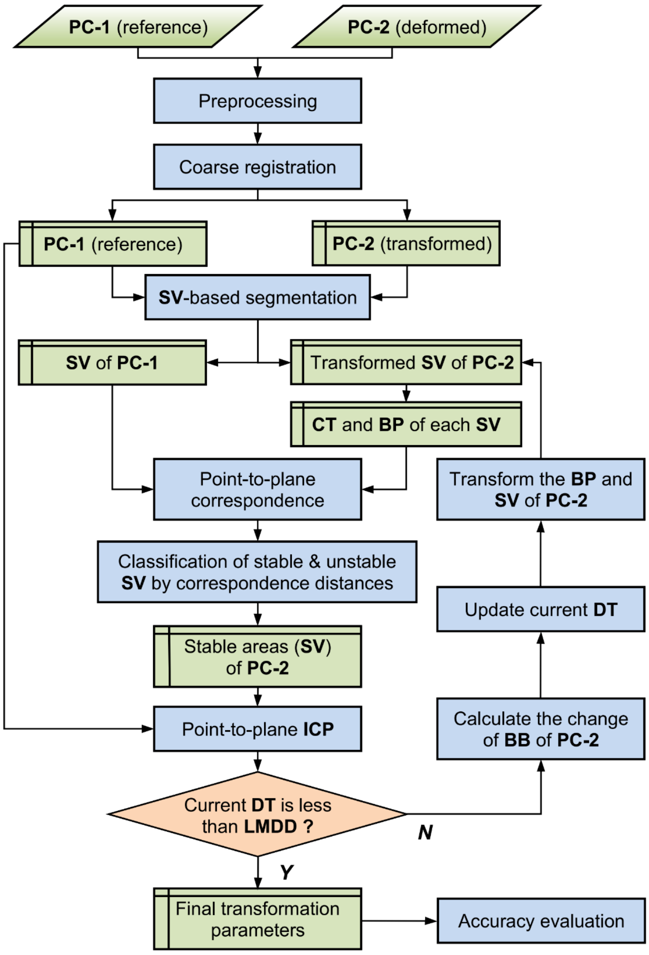

PC-1 (reference)
PC-2 (deformed)
Preprocessing
Coarse registration
PC-1 (reference)
PC-2 (transformed)
SV-based segmentation
SV of PC-1
Transformed SV of PC-2
CT and BP of each SV
Point-to-plane correspondence
Transform the BP and SV of PC-2
Classification of stable & unstable SV by correspondence distances
Update current DT
Stable areas (SV) of PC-2
Point-to-plane ICP
Calculate the change of BB of PC-2
Current DT is less than LMDD ?
N
Y
Final transformation parameters
Accuracy evaluation

Figure 4.1: Flowchart of the proposed targetless registration method (Yang and Schwieger, 2022). Abbreviations in the figure: *SV: supervoxel, CT: centroid, BP: boundary point, BB: bounding box*.

4.1.2 Supervoxel-based Over-segmentation

Based on the assumptions in Section 3.1.1, the deformed point cloud can be divided into small and locally rigid patches by segmentation techniques. Then the derived patches could be respectively analyzed for the classification of stable and unstable areas. As depicted in Section 2.2.3, the commonly used voxel-based segmentation fails to distinguish the boundary between stable and unstable areas in one segment (voxel). Therefore, the supervoxel-based over-segmentation strategy is exploited herein to partition the point clouds into small planar patches while simultaneously preserving the boundaries (see Section 3.2.1). Compared to the single point-based strategy (i.e., identifying the stability of each measured point), this patch-based division strategy is able to considerably improve the registration speed for large-scale point clouds and suppress the misclassification of unstable points as stable. For example, a point whose correspondence distance is within the DT may be erroneously classified as stable if it is located in an area with in-plane movements.

To ensure the computational efficiency of the pipeline, as few representative points as possible should be selected from each supervoxel to construct correspondences. Herein, the boundary points of one supervoxel are adopted considering the approximately planar property of a single supervoxel, as shown in Figure 4.2. Boundary points mean the points of maximal and minimal coordinates in three orthogonal directions. It is assumed that all points in one supervoxel are regarded as stable (i.e., this supervoxel is located in stable areas) as long as the distances of all correspondences constructed by all boundary points in this supervoxel are less than a DT. In addition to the boundary points, the centroid of supervoxel is also calculated to be used for searching the nearest supervoxel (see Section 4.1.3). It should be emphasized that the best-fit plane of the supervoxel estimated by least squares is only exploited to obtain the normal vector for calculating correspondence distances and will not be regarded as a geometric primitive for relevant deformation analysis.

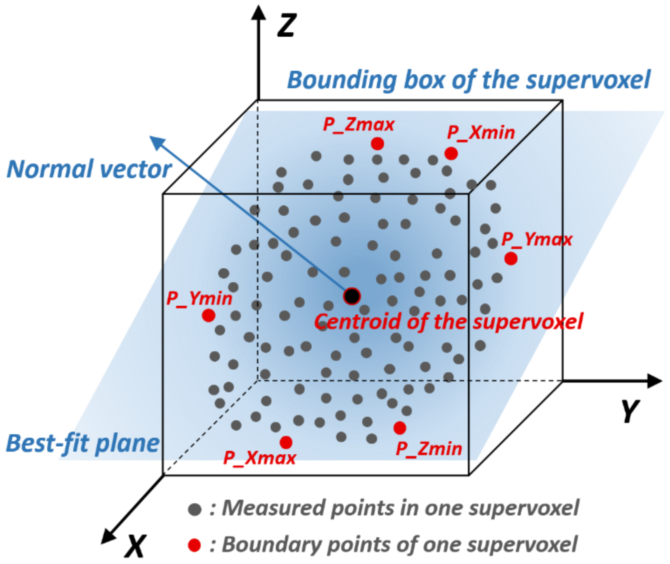

Z
Bounding box of the supervoxel
P_Zmax
P_Xmin
Normal vector
P_Ymax
P_Ymin
Centroid of the supervoxel
Y
Best-fit plane
P_Xmax
P_Zmin
X
● : Measured points in one supervoxel
● : Boundary points of one supervoxel

Figure 4.2: Boundary points of the supervoxel (Yang and Schwieger, 2022).

4.1.3 Correspondence Construction and Classification of Stable and Unstable Supervoxels

In voxel-based methods proposed by Friedli and Wieser (2016), correspondences are directly built between the centroids of nearest voxels in two scans, as depicted in Figure 4.3(a). The centroids of equal-sized voxels in PC-2 (*V'1*, *V'2*, … *V'n*) are calculated and connected to the nearest centroids of voxels in PC-1 (*V1*, *V2*, … *Vm*) as the point-to-point correspondences. For example, *V3* and *V'3* are classified as stable for a short correspondence distance, but unstable parts still exist in *V'3*, thus leading to an inaccurate estimation of transformation parameters. Although voxels incorporating mixed stable and unstable points can be avoided by reducing the voxel size, false matchings of centroids that generate incorrect correspondences may still arise if the voxel size is set too small (Friedli and Wieser, 2016; Friedli, 2020).

Herein, a boundary point-to-supervoxel plane (BP-to-SVP) correspondence is proposed to cope with the problems in the voxel-based method. As illustrated in Figure 4.3(b), both point clouds are over-segmented into locally planar but irregular-sized supervoxels (*SV1*, *SV2*, … *SVm* in PC-1 and *SV'1*, *SV'2*, … *SV'n* in PC-2) that are assumed rigid individually, allowing for an adaptation to surface variations. The normal vector of each supervoxel in PC-1 results from the best-fit plane of the points inside. The projection of the distance between the boundary point ($BP_{\_i}$) in PC-2 and its nearest centroid in PC-1 on the normal vector of this centroid (i.e., the supervoxel's orientation) is defined as the distance of BP-to-SVP correspondence, i.e., D_{cor_i} of the $BP_{\_i}$ that can be calculated by

$$D_{cor\_i} = \left| n_x(x_{BP\_i} - x_{CT}) + n_y(y_{BP\_i} - y_{CT}) + n_z(z_{BP\_i} - z_{CT}) \right|, \tag{4.1}$$

where $(x_{BP\_i}, y_{BP\_i}, z_{BP\_i})$ and (x_{CT}, y_{CT}, z_{CT}) are the coordinates of the boundary point $BP_{\_i}$ in PC-2 and its nearest supervoxel centroid in PC-1, (n_x, n_y, n_z) is the normalized normal vector of the centroid of the nearest supervoxel centroid in PC-1. Detailed illustrations of the constructed BP-to-SVP correspondences are shown in Figure 4.4.

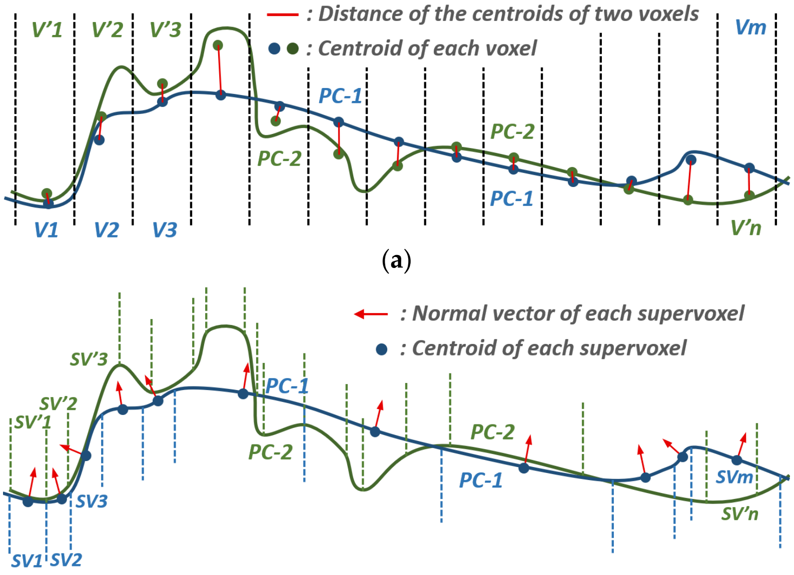


Figure 4.3: Spatial division of point clouds by voxel-based (a) and supervoxel-based (b) methods (Yang and Schwieger, 2022). Dashed lines denote the boundaries of segmented voxels or supervoxels.

The supervoxel will be classified as stable only when the D_{cor_i} of all BP-to-SVP correspondences in one supervoxel are less than a specific DT (see Section 4.1.4). Otherwise, the supervoxel will be removed from the input points of PC-2 for the following ICP-based registration. As shown in Figure 4.4, *SV(i)* is classified as unstable due to two correspondences beyond the DT, while *SV(i+1)* is classified as stable since the distances of all BP-to-SVP correspondences are within the DT.

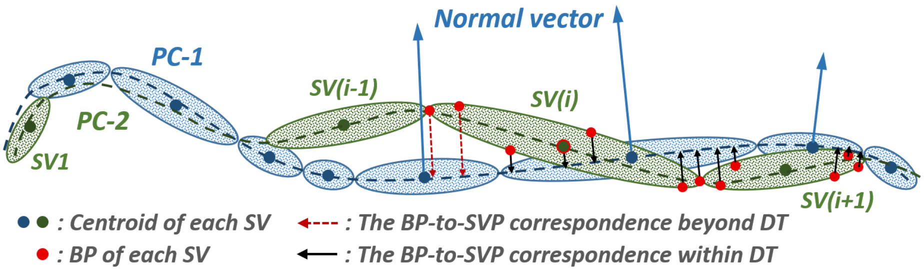

Normal vector
PC-1
PC-2
SV1
SV(i-1)
SV(i)
SV(i+1)
: Centroid of each SV
: BP of each SV
: The BP-to-SVP correspondence beyond DT
: The BP-to-SVP correspondence within DT

Figure 4.4: Classification of stable and unstable supervoxel (SV) by comparing BP-to-SVP correspondences with the distance threshold (Yang and Schwieger, 2022).

Since the distance of a BP-to-SVP correspondence is calculated from a point to a plane, the point-to-plane ICP is adopted to register the PC-2 containing only stable supervoxels to PC-1 in each iteration. This point-to-plane registration strategy exhibits faster convergence and better robustness to the case of varying point densities compared to the standard point-to-point ICP (Park and Subbarao, 2003). Algorithm details of point-to-plane ICP can be found in

Appendix A.3. These point-to-plane correspondences will be reconstructed and updated after each iteration in the inner loop of ICP as well as the outer loop involving DT updating.

4.1.4 Piece-wise Distance Threshold

As presented in Section 4.1.3, the classification of stable and unstable areas (supervoxels) is achieved by comparing the distances of BP-to-SVP correspondences to a piece-wise DT in each outer-loop iteration. Due to the error from coarse registration and no prior knowledge of the magnitude of deformations, the initial value of DT should not be too small in case of insufficient retained points that may lead to a local optimization (minimum) in ICP. Herein, a four-stage piece-wise DT in the k-th iteration is defined as

$$DT_k = \begin{cases} \overline{D}_{k,\,entire} + \sigma_{k,\,entire}\ , & stage\ 1 \\ \overline{D}_{k,\,stable} + \sigma_{k,\,stable}\ , & stage\ 2 \\ \beta \cdot DT_{k-1}, & stage\ 3 \\ LMDD\ , & stage\ 4 \end{cases}, \tag{4.2}$$

where $\bar{D}_{k,entire}$ and $\sigma_{k,entire}$ are the mean and standard deviation of distances of all BP-to-SVP correspondences, $\bar{D}_{k,stable}$ and $\sigma_{k,stable}$ are merely for the correspondences derived from stable supervoxels classified in the last iteration. In stage-3, DT will decrease linearly with a scaling factor $\beta \in (0,1)$ to the LMDD that indicates the end of the outer iteration. The change from one stage to the next stage is triggered when the change of corner points of the bounding box comes to convergence. The update of DT between the four stages is monotonically decreasing, resulting in a progressive identification of all stable areas. It is worth noting that, for the scans of which the majority are deformed but have already been well initially aligned (i.e., good coarse registration), an approximate coarse registration error can be used as the initial DT and the user can start the procedure from stage-2 directly (skip stage-1).

Theoretically, LMDD should be zero since no deformation occurs between the corresponding stable areas within PC-1 and PC-2, but uncertainties associated with each measurement in the laser scanning process imply that the LMDD ought to be determined by the boundary of a confidence interval involving the stochastic model of both point clouds. Herein, LMDD is computed parametrically based on the assumption of independent Gaussian distributions of the boundary points and the estimated centroid of associated supervoxels, thus deriving a Gaussian distribution of D_{cor_i} as well. Hence, LMDD is defined as

$$LMDD = y_P \cdot \sqrt{\sigma^2_{BP\_i,PC-2} + \frac{\sigma^2_{CT,PC-1}}{n_{eff}}}, \tag{4.3}$$

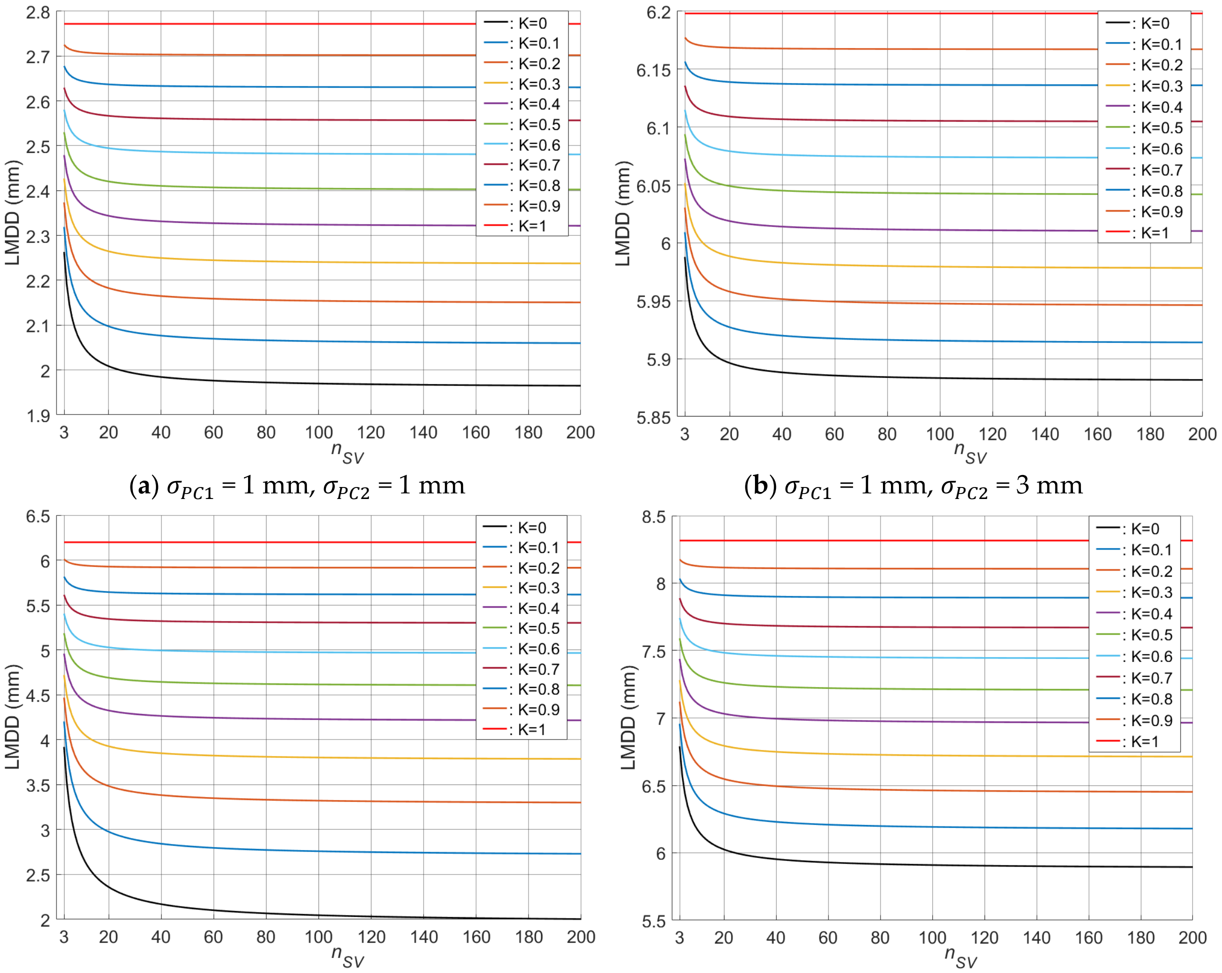
where y_P is the quantile of the standard normal distribution at a confidence level P (e.g., y_P equals 1.96 when P = 95%), $\sigma^2_{BP\_i,PC-2}$ is the variance of the boundary point in PC-2 and $\sigma^2_{CT,PC-1}$ is the nominal variance of the centroid of supervoxel which should be divided by the effective number of measurements n_{eff} considering the potential correlation between the points within one supervoxel (explained in Section 3.2.1). $\sigma^2_{CT,PC-1}$ can be obtained by

$$\sigma^2_{CT,PC-1} = \frac{\sum_{i=1}^{n_{SV}} \sigma_i^2}{n_{SV} - 1}, \tag{4.4}$$

where σ_i^2 is the variance of each single point in the supervoxel and n_{SV} is the number of points in this supervoxel. The n_{eff} here can be calculated by Equation (3.6) or its simplified approximation by Equation (3.7).

Since the determination of the stochastic model of point clouds is one of the most challenging issues in the TLS domain, the variances (or standard deviations) of measured points and boundary points herein are both derived based on an approximate assumption of the stochastic model by empirically and predominately considering the scanner's specifications from the manufacturer and the actual measuring distance.

To determine an empirical LMDD based on the approximate standard deviations of 3D positions of measured points, the relationship between LMDD (y_P = 1.96) and the point quantity in a patch (n_{SV}) with different approximate averages of correlation coefficients (i.e., K calculated by Equation (3.7)) is investigated and presented in Figure 4.5 (σ_{PC1} and σ_{PC2} are approximate standard deviations of measured points in PC-1 and PC-2, respectively. Here four combinations of σ_{PC1} and σ_{PC2} are presented as examples).



(**a**) σ_{PC1} = 1 mm, σ_{PC2} = 1 mm

(**b**) σ_{PC1} = 1 mm, σ_{PC2} = 3 mm

(**c**) σ_{PC1} = 3 mm, σ_{PC2} = 1 mm

(**d**) σ_{PC1} = 3 mm, σ_{PC2} = 3 mm

Figure 4.5: The relationship between LMDD and n_{SV} with different K and σ_{PC} within the patch.

Similar to Figure 3.7, when n_{SV} is larger than a certain value, LMDD also tends to converge without much decrease, especially when $K > 0$. From Figure 4.5, it can be seen that LMDD is significantly influenced by K (a higher K yields a higher LMDD). Obviously, higher standard deviations (σ_{PC1} and σ_{PC2}) lead to higher LMDD. Besides, comparing Figure 4.5(b) and (c) to Figure 4.5(a) and (d), LMDD is more sensitive to the change of σ_{PC2}, because the influence of the increase or decrease of σ_{PC1} will be weakened by the effective number of measurements as indicated in Equation (4.3).

4.1.5 Iteration Condition and Calculation of Final Transformation Parameters

Different from Tr-ICP, the proposed algorithm performs a complete point-to-plane ICP after the identification of stable areas in each iteration of the outer loop. Therefore, the stopping criterion of the outer iteration is only when DT decreases to LMDD instead of the convergence of the change of the bounding box's corners (or transformation parameters) which is adopted from Friedli and Wieser (2016). Obviously, the smaller β is set, the less iterations are needed. However, too small β makes the rejection of outliers more radical and possibly leads to the local optimization of ICP. Herein, 0.8 is empirically suggested for setting the scaling factor in the third stage of DT.

The final transformation matrix of $\boldsymbol{T}_{final}$ incorporating six transformation parameters can be generated by Equation (4.5) where $\boldsymbol{T}_{coarse}$ and $\boldsymbol{T}_{iteration}$ are the transformation matrices derived from a coarse registration and each iteration (in the outer loop) of the proposed method, respectively.

$$\boldsymbol{T}_{final} = \begin{bmatrix} \underset{3\times3}{\boldsymbol{R}} & \underset{3\times1}{\boldsymbol{t}} \\ 0 & 1 \end{bmatrix} = \boldsymbol{T}_{iteration\_k} \cdot \ldots \cdot \boldsymbol{T}_{iteration\_1} \cdot \boldsymbol{T}_{coarse\_0} = \prod_{i=0}^{k} \boldsymbol{T}_i \tag{4.5}$$

4.2 Experimental Evaluation of the Proposed Registration Method

Two experiments representing the indoor environment (short scanning distance) and natural environment (long scanning distance) were carried out to verify and evaluate the proposed registration method by comparing it with other robust algorithms. The derived two datasets in both experiments consist of two-epoch scans, where significant deformations and changes occurred in the second epoch.

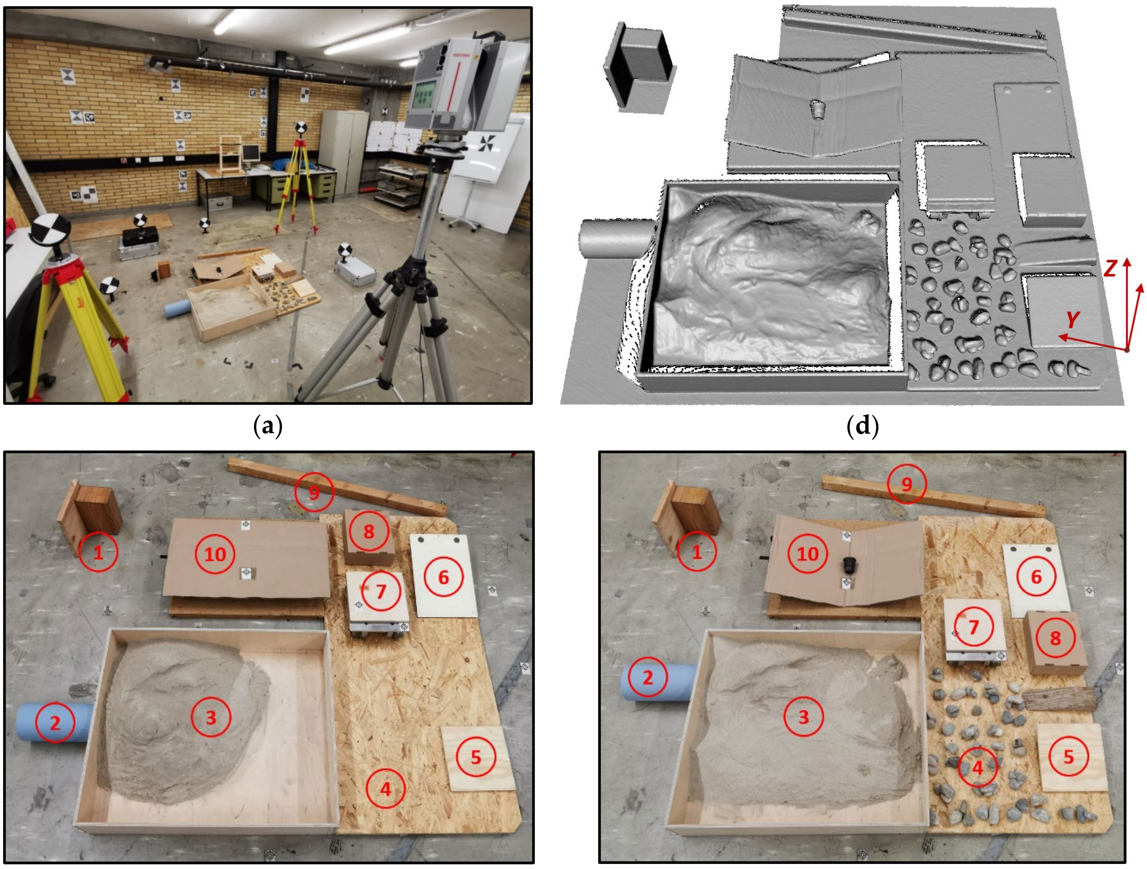
4.2.1 Evaluation of Registration Accuracy

The absolute registration accuracy herein is evaluated based on the comparison with the reference transformation parameters obtained by means of reliable artificial targets, including three rotation angles (R_x, R_y, R_z) and three translation components (t_x, t_y, t_z). Then the absolute value of the difference between the transformation parameter are calculated as registration errors (i.e., ΔR_x, ΔR_y, ΔR_z, Δt_x, Δt_y, Δt_z).

Besides the comparison of estimated transformation parameters, the distribution of distances between two registered surfaces within the stable areas can also reflect the absolute registration accuracy provided the prior knowledge of the location of stable areas. In this way, the registration accuracy of specific directions can be indicated by the calculated distance between the surfaces with corresponding orientations. To avoid a spurious distance calculation between non-overlapping areas, M3C2 distance could be employed to generate a distance distribution histogram (Lague *et al.*, 2013).

4.2.2 Indoor Scene with Simulated Changes (Dataset 1)

The first dataset was acquired from an indoor scene with different moved or deformed objects arranged on the ground. A Leica HDS7000 laser scanner was set up at two different stations in two respective epochs to capture the investigated area as well as six black and white planar targets fixed around this area. As depicted in Figure 4.6, various kinds of rigid-body movements (in-plane and out-of-plane) and shape deformations were simulated by manually changing the positions or shapes of the objects of different geometries. This indoor experiment represents a highly controlled situation with known ground truth and negligible environmental influences. The transformation parameters derived from the six artificial targets are considered as a reference to evaluate the registration accuracy. The (local) coordinate system of the scanner is illustrated in Figure 4.6(d).



(a) (d)

(b) (c)

Figure 4.6: Experimental setup of the indoor simulated scene: (b) and (c) are scenes in epoch-1 and epoch-2 respectively; (d) is the acquired point cloud in epoch-2. Objects with numbers are placed or changed as follows: (1) is a stationary wooden block whose surfaces are used to visually present and evaluate the registration result, but not involved in the registration dataset; (2) is a cylinder moved backward; (3) is sand that is reshaped partially; (4) are gravels added in epoch-2; (5) is a wooden board lifted on one side; (6) is a wooden board laid down on one side; (7) is a wooden board moved horizontally forward; (8) is a cuboid that is moved to another position; (9) is a wooden strip rotated around the right end; (10) is a cardboard that sinks by placing a weight (Yang and Schwieger, 2022).

To assess the robustness of the proposed registration method concerning varying point densities in the two scans (a situation sometimes encountered in geodetic monitoring data), the two-epoch point clouds are downsampled with a point spacing of 6 mm (PC-1) and 3 mm (PC-2) respectively, to illustrate scenarios where data resolutions differ between epochs.

The weight factor w_n is set to four times w_d empirically in Equation (3.1) when performing a VCCS-based supervoxel generation. The segmentation results of the point cloud based on voxel and supervoxel are illustrated in Figure 4.7. It can be seen that the boundaries between different objects are explicitly distinguished in supervoxel-based over-segmentation rather than incorporated in one patch from the voxel-based segmentation. The normal vector indicates the local surface orientation of each supervoxel.

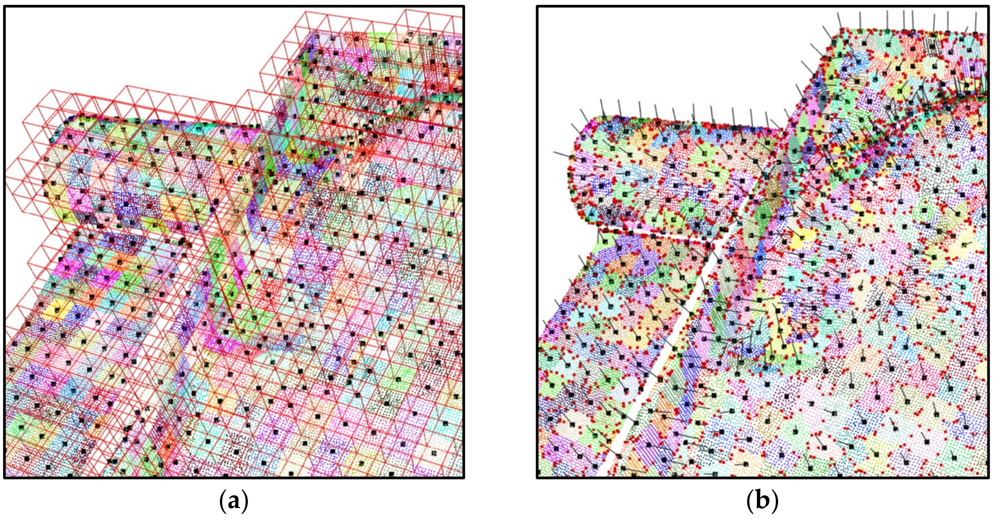


Figure 4.7: Voxel-based (a) and supervoxel-based (b) over-segmentation of PC-2 in dataset-1 (partial view of the cylinder and the wooden box with sand). Patches with random colors are segmented voxels (a) and supervoxels (b), black points are centroids of patches, red cubes (a) are octree-based voxel leaves, short black lines (b) represent the directions of the normal vectors of each supervoxel (Yang and Schwieger, 2022).

As analyzed in Section 4.1.4 regarding Figure 4.5, LMDD is predominately determined by the standard deviations of two point clouds and the correlation coefficients within the patch. Herein, empirical approximations of σ_{PC1} and σ_{PC2} are set as 0.9 mm considering the scanner's specification and measuring distances, while an empirical K of 0.65 is adopted from Kauker and Schwieger (2017) for the point cloud data in this indoor scene. In dataset-1, thus, LMDD is set to 2.3 mm by Equation (4.3) at a 95% confidence level. The initial DT before the iteration process is defined from the result of coarse registration. Herein, the mean distance of all BP-to-SVP correspondences plus double standard deviation is used as the initial DT which is 54.8 mm in dataset-1. The update of DT during the iteration is shown in Figure 4.8. It starts from Stage-1 and shows a significant decline of DT in Stage-2, then gradually tends to LMDD, ending at Stage-4.

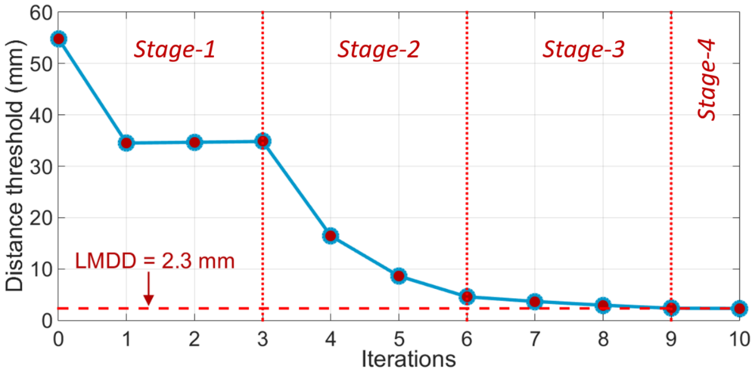
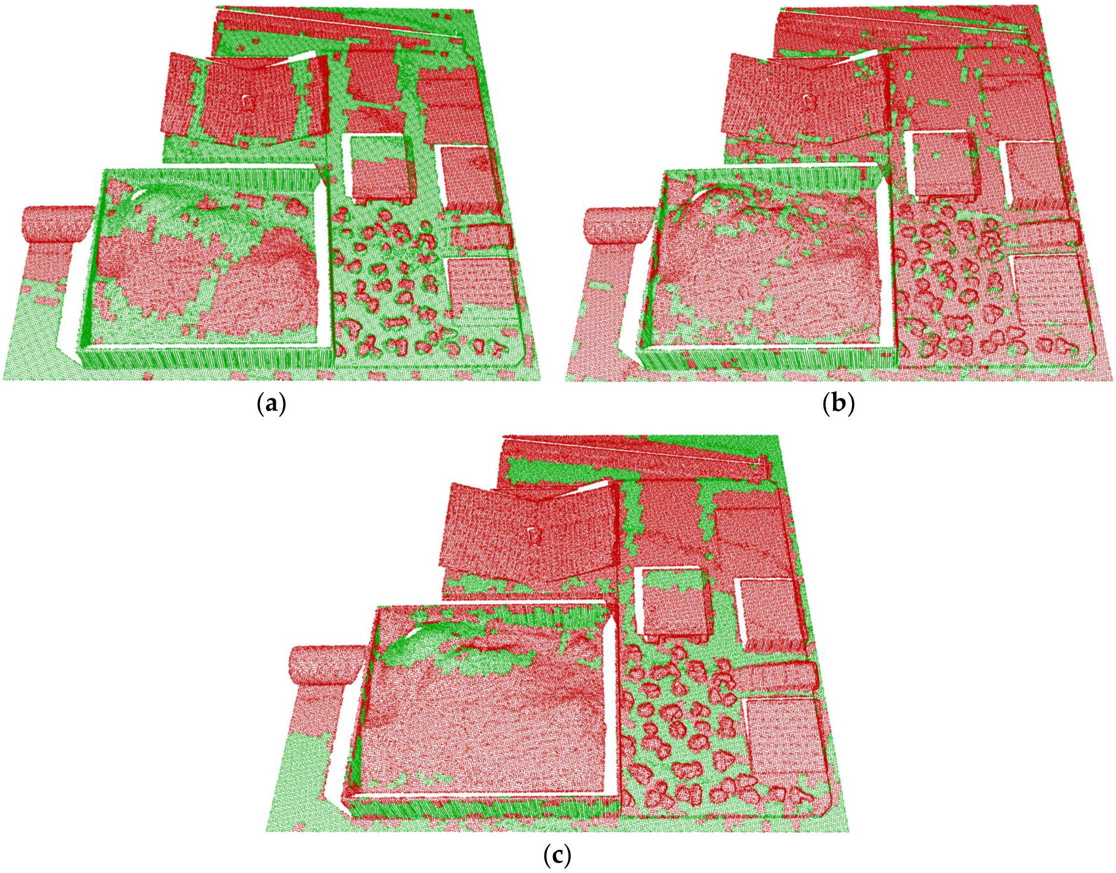


Figure 4.8: The change of DT with iterations in dataset-1 (Yang and Schwieger, 2022).

As the DT decreases to LMDD, the identification result of stable areas of PC-2 is presented as the green parts in Figure 4.9(c), leaving the red parts regarded as unstable areas. These red parts not only incorporate the actually deformed or moved areas but also contain the non-overlapping areas caused by different scanning views from two stations, while these areas may keep unchanged between two epochs. Some overlapping as well as stable areas are still classified as unstable (e.g., some ground areas between gravels) because of the error from supervoxel-based over-segmentation (two or more local planes are not segmented individually). Nevertheless, these misclassified unstable areas will hardly influence the registration result as long as the identified stable areas exhibit a good distribution. The identification results by the voxel-based method are presented as well. The correspondences are constructed by not only the nearest centroids adopted from Friedli and Wieser (2016), but also the proposed BP-to-SVP correspondence for a fair comparison. Due to the ambiguous separation of point stabilities in the equal-sized voxels, the unstable areas are either under-identified (i.e., the unstable areas are mistakenly identified as stable areas, see Figure 4.9(a)) or over-identified (i.e., the stable areas are mistakenly identified as unstable areas, see Figure 4.9(b)) with the same DT setting, and both unsatisfactory identification will finally result in an inaccurate registration.


(**a**) (**b**)

(**c**)

Figure 4.9: Classification of stable (green) and unstable (red) areas of PC-2 in dataset-1: (a) the voxel-based method using nearest centroids; (b) the voxel-based method using BP-to-SVP correspondences; (c) the proposed supervoxel-based method (Yang and Schwieger, 2022).

The absolute registration accuracy (see Section 4.2.1) is evaluated by comparing the transformation parameters (including rotation angles and translation components) estimated by the proposed method (SV-M), standard point-to-plane ICP (P2P-ICP), Tr-ICP, G-ICP, and the voxel-based method (V-M) proposed by Friedli and Wieser (2016) with the reference values derived from six fixed artificial targets, as listed in Table 4.1.

It can be seen that the proposed method displays a considerably better accuracy in dealing with the targetless registration of two TLS point clouds with different point densities and complex deformations than the other four methods. Submillimeter-level registration accuracy with respect to the translation can be achieved in this indoor scene.

Table 4.1: Comparison of registration errors in dataset-1 (Yang and Schwieger, 2022).

Method	Rotation angle (°)			Translation (mm)		
	ΔR_x	ΔR_y	ΔR_z	Δt_x	Δt_y	Δt_z
P2P-ICP	0.813	0.582	2.065	20.09	67.37	28.61
Tr-ICP	0.093	0.026	4.353	75.89	75.24	2.25
G-ICP	0.584	0.143	0.398	4.24	26.63	15.39
V-M	0.144	0.144	0.047	5.18	4.86	6.14
SV-M	0.006	0.003	0.009	0.13	0.36	0.26

The accuracy evaluation is also conducted based on the registered point clouds on the stationary wooden blocks (object-1 in Figure 4.6) that have orthogonal planes in three directions and keep stable throughout the experiment. Theoretically, all distances between the stable areas of PC-1 and registered PC-2 should be zero if the alignment has been perfectly performed. Figure 4.10 presents the registered two point clouds within the stable areas by the above five methods.

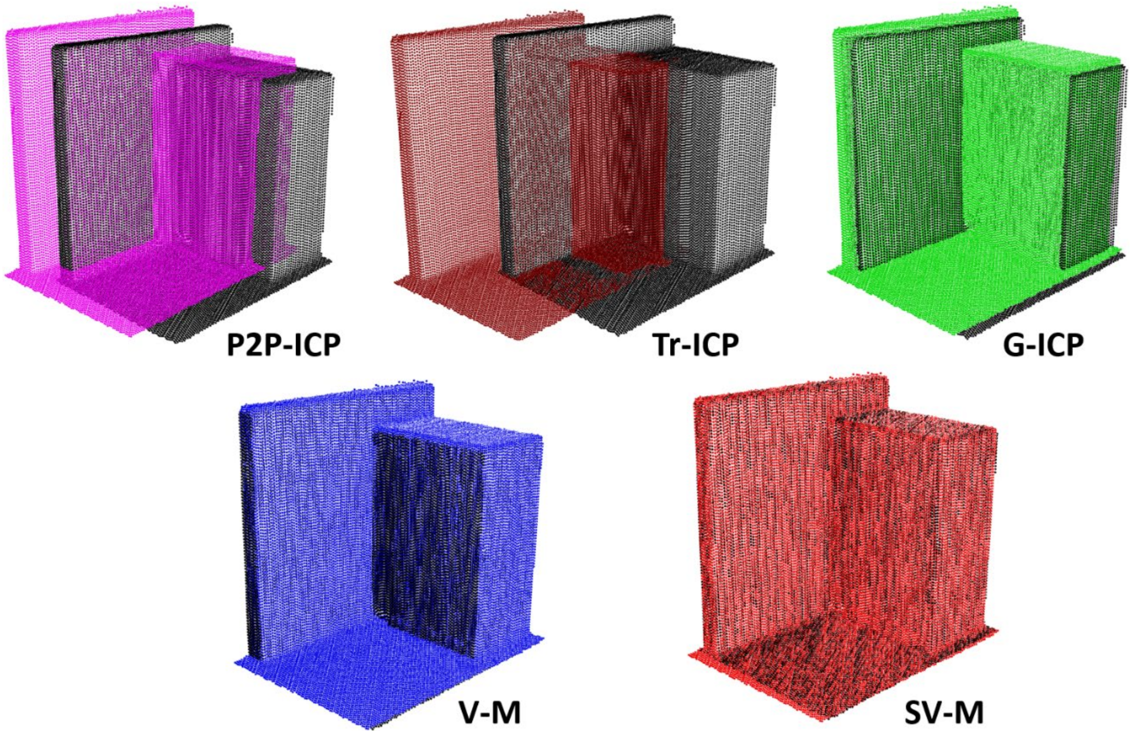


Figure 4.10: Comparison of registered two point clouds in stable areas by five registration methods.

From the visual registration results, the voxel-based and the supervoxel-based methods output much better alignment than other ICP variants, which agree well with the registration errors of transformation parameters in Table 4.1. To quantify the difference between registered surfaces, M3C2 distances are calculated between stable areas, and their distribution histograms are displayed accordingly, as shown in Figure 4.11. It can be seen that the proposed method yields a more accurate alignment in all directions, where the distances between all registered surfaces are within 1 mm (this agrees with the submillimeter-level accuracy of estimated translation parameters).

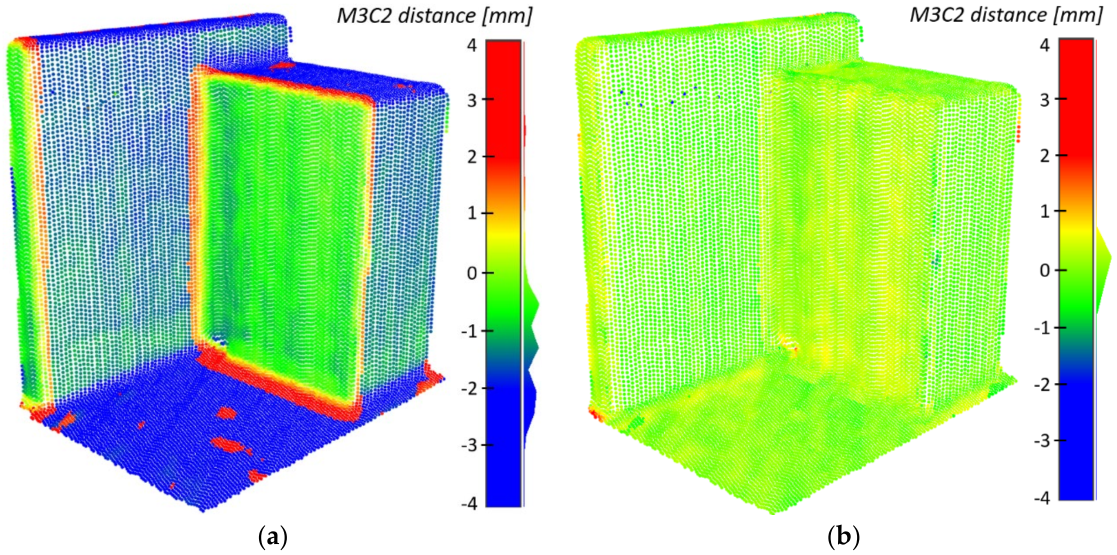


Figure 4.11: Distribution of the M3C2 distance between registered two point clouds within the stable areas by the voxel-based method (a) and the proposed method (b).

4.2.3 Nesslrinna Landslide (Dataset 2)

The Nesslrinna landslide is located on an eastern slope between Zwieselstein and Obergurgl in Austria, as shown in Figure 4.12(a). Two-epoch point clouds were acquired during the Innsbruck Summer School of Alpine Research using different TLS sensors (Pfeiffer *et al.*, 2019). The first scan was acquired by Leica ScanStation C10 in July 2015, and the second scan was obtained by Z+F IMAGER5016 in July 2017. Significant deformations and changes occurred due to the landslide activity and specific environmental processes between each acquisition epoch.

In this thesis, only the landslide areas are selected from the original point cloud datasets to evaluate the registration algorithms. The measuring distances in both epochs range from approximately 50 m to 200 m. The two-epoch scans were already registered by using artificial targets installed on the stable areas and further improved by manually selecting well-distributed stable areas to perform an ICP-based fine registration. Hence, self-defined transformation parameters are performed on PC-2 to generate an unregistered dataset. Thus, these defined transformation parameters are used as the reference for the following accuracy evaluation. Two-epoch TLS point clouds and the simulated unregistered scan are shown in Figure 4.12(b). The coordinate system of the point clouds is also illustrated in Figure 4.12(b).

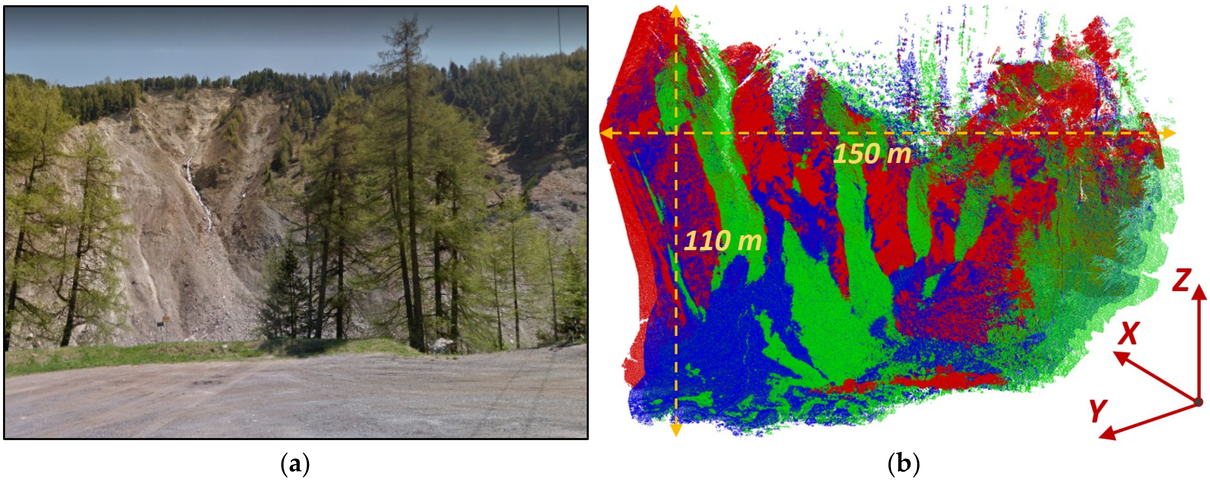


Figure 4.12: Nesslrinna landslide (a) and acquired TLS point clouds (b) (red: PC-1; blue: the registered PC-2; green: the unregistered PC-2 generated by defined transformation parameters).

Taking into account the longer scanning distance and the greater complexity of surface conditions, dataset-2 exhibits higher measurement uncertainties compared to the indoor scene. Consequently, it yields a higher LMDD of 5.4 cm, calculated using Equation (4.3). Furthermore, the increased magnitude of deformations necessitates more iterations, as shown in Figure 4.13. Interestingly, there is a consistent downward trend in the change of DT similar to dataset-1, except for Stage-1, where it experiences a very slight increase.

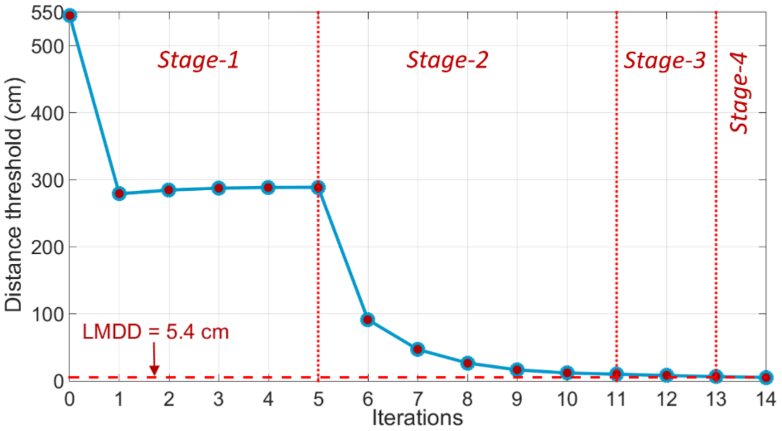


Figure 4.13: The change of DT with iterations in dataset-2 (Yang and Schwieger, 2022).

To present the difference between voxel-based and supervoxel-based methods, the results of stable area identification by both methods are offered in Figure 4.14. The distribution of stable voxels is generally similar to that of stable supervoxels, mainly located on the left and right sides of the upper area of the landslide. However, some deformed areas around the soil-stone accumulation in the front of the landslide are misclassified as stable in the voxel-based method, which has contaminated the employed point set to estimate transformation parameters. Besides, the voxel of regular size is difficult to represent the locally stable unit of landslide surface when complex deformations occur.

The comparison between the accuracy of five targetless registration strategies is given in Table 4.2, showing that the proposed supervoxel-based method performs much better than other ICP variants and the voxel-based method.

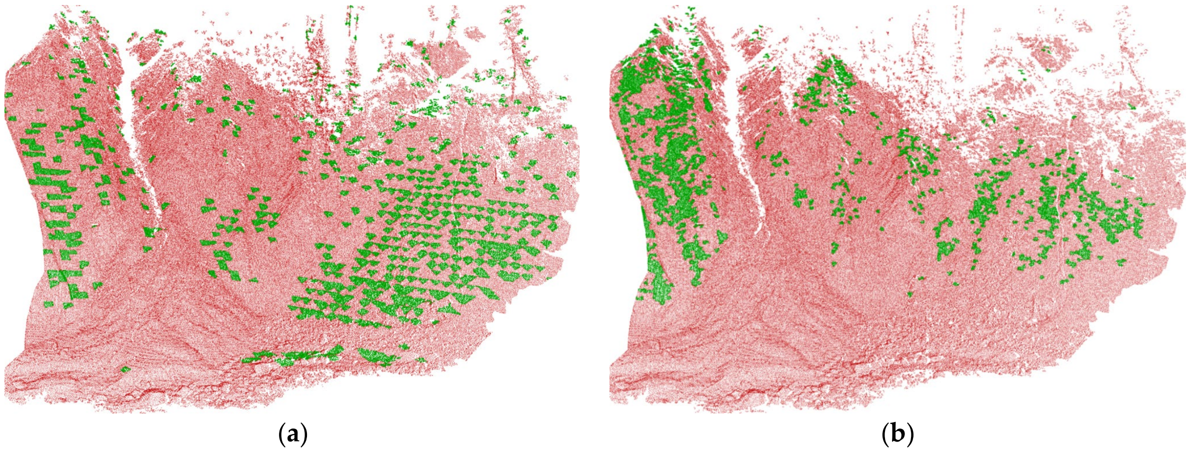


Figure 4.14: Classification of stable (green) and unstable (red) areas of PC-2 in dataset-2: (a) the voxel-based method; (b) the proposed supervoxel-based method (Yang and Schwieger, 2022).

Table 4.2: Comparison of registration errors in dataset-2 (Yang and Schwieger, 2022).

Method	Rotation angle (°)			Translation (cm)		
	ΔR_x	ΔR_y	ΔR_z	Δt_x	Δt_y	Δt_z
P2P-ICP	0.282	1.599	1.411	354.34	349.78	239.15
Tr-ICP	0.170	0.663	2.203	10.71	10.48	120.69
G-ICP	0.250	0.267	0.035	0.80	13.52	108.35
V-M	0.126	0.080	0.070	2.71	19.88	29.67
SV-M	0.029	0.017	0.034	1.70	9.79	3.34

4.3 Discussion

The experimental results of two datasets show that standard ICP and its two robust variants display a low registration accuracy, as certain correspondences are constructed between unstable areas. When these incorrect correspondences reach a certain amount, the registered point cloud deviates to the deformed areas. The voxel-based method classifies the voxels merely according to the distance between the nearest centroids, ignoring the mixing of stable and unstable points in one voxel whose centroid is still within the DT. The proposed supervoxel-based method enables accurate registration even for a high ratio of unstable areas in the scan (e.g., more than 60% in dataset-1 and more than 80% in dataset-2 seen from the identification results of the experiments). In addition to the presented pipeline and associated experimental results above, some key influence factors or alternative steps are discussed and suggested in this section.

4.3.1 Influence of Generated Supervoxels

The over-segmentation of point clouds in both experiments presented in Section 4.2 is based on VCCS (Papon *et al.*, 2013), where the generated supervoxels have more homogeneous sizes and shapes. To investigate the influence of different supervoxel-based over-segmentation approaches, the FEO-based method (Lin *et al.*, 2018) is also used to identify stable areas and derive transformation parameters. Figure 4.15 compares the results of stable-area identification by different supervoxel generation methods. The black dashed frames highlight incorrect identifications (stable areas are erroneously classified as unstable) compared to the other method. Although both methods yield wrong classifications in certain areas, the FEO-based method generally displays a slightly better separation of stable areas from the scene than the VCCS-based method.

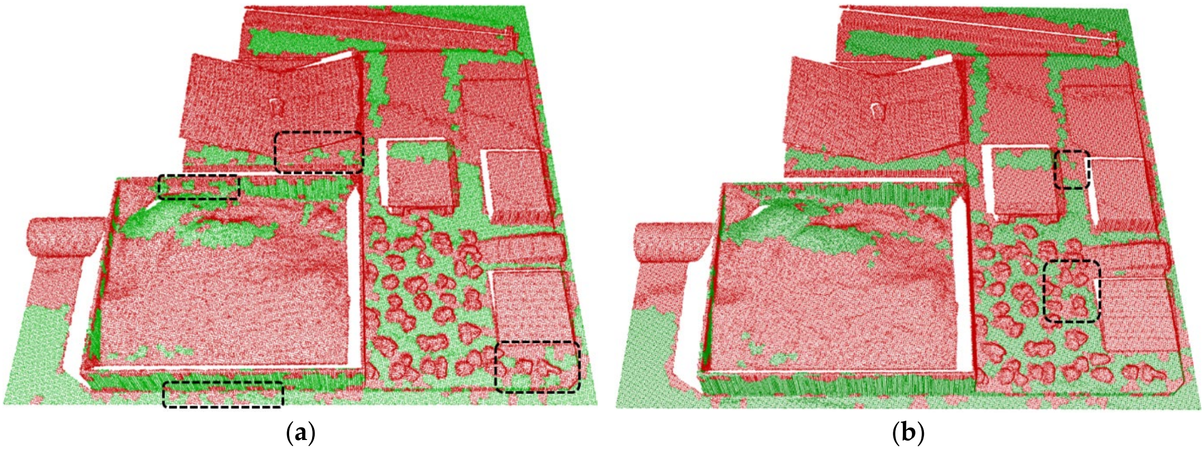


Figure 4.15: Comparison of the classification results of stable (green) and unstable (red) areas of PC-2 in dataset-1 by (a) the VCCS-based and (b) FEO-based supervoxel generation methods.

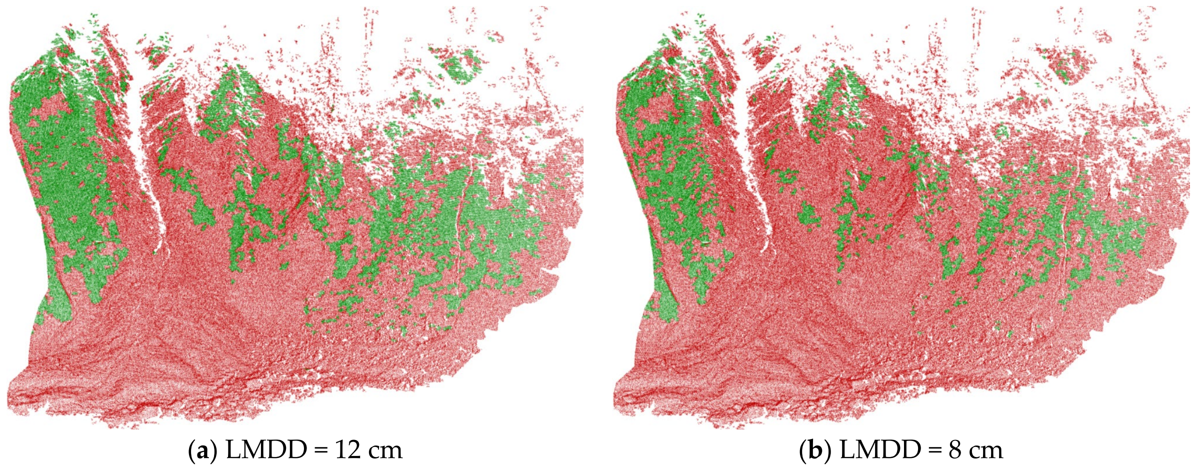
Table 4.3 compares the absolute registration errors by the proposed pipeline under two supervoxel generation approaches (other parameter settings remain the same) in two experiments. It can be seen that better segmentation (i.e., better classification of stable and unstable areas) by the FEO-based method can improve the registration accuracy to a certain extent. This can be explained by its better performance on boundary preservation compared to the VCCS-based over-segmentation.

Table 4.3: Comparison of registration errors in two datasets by different supervoxel generations.

Dataset	Method	Rotation angle (°)			Translation (cm)		
		ΔR_x	ΔR_y	ΔR_z	Δt_x	Δt_y	Δt_z
Dataset 1	*SV-M (VCCS)*	0.006	0.003	0.009	0.13	0.36	0.26
	SV-M (FEO)	0.005	0.001	0.005	0.28	0.33	0.20
Dataset 2	*SV-M (VCCS)*	0.029	0.017	0.034	1.70	9.79	3.34
	SV-M (FEO)	0.024	0.014	0.025	2.04	6.87	2.33

4.3.2 Influence of LMDD

The final registration accuracy depends on the uncertainty as well as the distribution of identified stable points, while the calculated LMDD will directly affect the error of identified stable areas and their spatial distribution, as presented in Figure 4.16 with two larger LMDDs in dataset-2. Thereby, a trade-off problem arises: A larger LMDD enables more points retained to estimate the transformation parameters, but with higher uncertainties in some areas; a smaller LMDD makes the remained stable points more accurate (i.e., shorter distances of the BP-to-SVP correspondences), but with a worse spatial distribution that may lead to a local optimization of the registration. The relationship between the LMDD and the quality of identified stable areas should be investigated quantitatively in future work.


(**a**) LMDD = 12 cm

(**b**) LMDD = 8 cm

Figure 4.16: Classification of stable (green) and unstable (red) areas in dataset-2 with different LMDD.

In the proposed method, an appropriate LMDD setting can guarantee the correctness and accuracy of identified stable areas. However, certain stable points located in unstable supervoxels are still rejected, resulting in the identified stable areas being smaller than the actual situation. The registration result will not significantly deteriorate by a small amount of this kind of misclassification (i.e., false positive), since the proposed algorithm sacrifices the number of identified stable points to ensure their correctness and accuracy.

4.3.3 Other Definitions of Correspondence Distances

In addition to calculating the distances of BP-to-SVP correspondences adopted in this contribution for classifying the stable and unstable supervoxels in PC-2, there are other alternatives with pros and cons:

(1) Simply calculating the distance from the supervoxel centroid in PC-2 to the supervoxel plane in PC-1: The computing efficiency can be dramatically improved but the accuracy will be reduced;

(2) Performing a local ICP for each supervoxel in PC-2 to the PC-1 surface individually and analyzing the magnitudes of the derived rotation or/and translation: The accuracy is better than the boundary points or centroid-based ways, but the computing time will be much longer.

The choice of the above ways depends on the requirements for registration performance from the user. In any case, the proposed BP-to-SVP correspondence enables a compromise between classification accuracy and computational efficiency.

5 Deformation Analysis Based on Rigid Patches

As the final part in the processing chain of TLS-based deformation monitoring, point cloud-based deformation analysis produces quantified deformation magnitudes, directions, and statistical significance by specific methods performed on registered or georeferenced point clouds. Based on three typical rigid patches segmented from TLS point clouds, three associated methods — patch-based M3C2 (Section 5.1), geometric primitive-based (Section 5.2), and local registration-based (Section 5.3) deformation analysis — are proposed in this thesis to deal with the current challenges discussed in Section 2.3.3. This chapter elaborates on the methodology of each approach and demonstrates them on simulated and/or real point cloud data. It should be noted that most deformations of the point clouds in this chapter are simulated in consideration of presenting diverse deformations and evaluating the absolute accuracies when no reference is available. At the end of this chapter (Section 5.4), suggestions on the method selection for specific scenarios and applications are given.

5.1 Patch-based M3C2

As depicted in Section 2.3.3, small-magnitude deformations under complex topographies may fail to be detected by classical point cloud-based deformation analysis methods. Aiming to deal with these challenges, this thesis, based on the framework of the existing standard M3C2 algorithm (Lague *et al.*, 2013), presents a novel method called patch-based M3C2 (PB-M3C2) to quantify 3D distances and detect geometric changes between two point clouds in the context of complex surface topographies. This new M3C2 variant can drastically reduce the uncertainties of detected deformations as well as preserve the original spatial resolution of measurements for a better representation of calculated deformations. The content of this section is mainly based on the author's publication: Yang Y, Schwieger V. **Patch-based M3C2: Towards lower-uncertainty and higher-resolution deformation analysis of 3D point clouds** in the *International Journal of Applied Earth Observation and Geoinformation*, 2023 (Yang and Schwieger, 2023).

5.1.1 Methodology of Patch-based M3C2

The general workflow of the proposed point cloud-based deformation analysis method is schematically shown in Figure 5.1. Two-epoch point clouds (PC-1 and PC-2 are the point clouds captured from monitoring epoch 1 and epoch 2, respectively) are firstly divided into locally planar patches by supervoxel-based over-segmentation. Then the standard deviations (i.e., *Std.* that will be defined in Section 5.1.1.1) of individual points are empirically estimated by fitting a local plane of the corresponding patch. The *Std.* in the direction of patch normals are projected along the associated deformation directions defined by the user. These projected standard deviations (*projected Std.*) of two point clouds are further used to determine the LoDetection, considering a confidence interval and registration uncertainties. The deformations between two point clouds are quantified by PB-M3C2 distances based on the constructed prisms from patches instead of the cylinders used in standard M3C2. Finally, the statistically significant deformations are detected by comparing PB-M3C2 distances with the associated LoDetection. Details of the PB-M3C2 are presented in the following subsections.

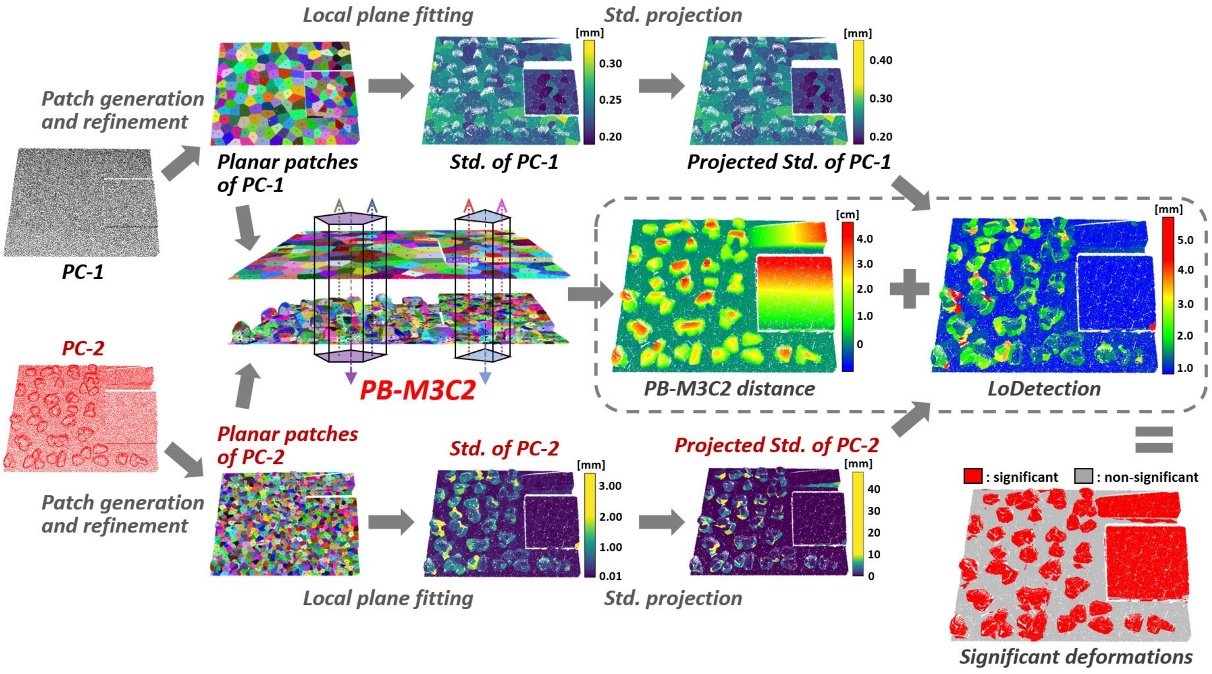


Figure 5.1: The workflow of patch-based M3C2 (PB-M3C2) (Yang and Schwieger, 2023).

5.1.1.1 Patch Generation, Modeling and Refinement

As assumed in Section 3.1.1, the local consistency of deformed point clouds can be kept within a specific neighborhood where calculations of deformation directions and empirical uncertainties of measured points are unified. This assumption is the prerequisite for deriving PB-M3C2 distances and associated uncertainties. These locally consistent patches can be generated by means of segmentation or clustering techniques based on the point similarities of spatial distance, local surface orientation, associated radiometric values (e.g., RGB colors and reflectance intensities), etc. Again, the supervoxel-based over-segmentation is exploited to partition the 3D point cloud into locally planar patches while preserving the boundaries (see Section 3.2.1 for details). The size of supervoxels is capable of adapting to the surface variation of point clouds automatically as long as the user sets an appropriate initial size (resolution) of supervoxels (Lin *et al.*, 2018). Besides the two empirical ways to determine the initial size suggested in Section 3.2.1, taking the same size (diameter) of the cylinder in standard M3C2 is also optional for a fair performance comparison.

The simplest model for the generated patches is the best-fitting planes by means of principal component analysis (PCA) or RANSAC. To approximate the uncertainties of individual points, an empirical standard deviation ($\hat{\sigma}(patch)$) of all points in a patch is defined by the root mean square of the orthogonal distances (*Dist*) from points in the patch (number of points: n_{patch}) to their best-fitting plane, as denoted in Equation (5.1). It should be emphasized that the standard deviation defined herein is only an approximation of the measurement uncertainties and is further used for determining the LoDetection.

$$\hat{\sigma}(patch) = \sqrt{\frac{\sum_{i=1}^{n_{patch}} Dist^2}{n_{patch} - 1}} \tag{5.1}$$

For the PCA-based plane fitting, points too far away from the fitted patch plane (e.g., $Dist > 3\hat{\sigma}(patch)$) could be eliminated to reduce the influence of imperfect segments (e.g., the points on two planes near the boundary are clustered into one patch). For the RANSAC-based plane fitting, such outliers are directly removed in the modeling process by setting a threshold (indicating the maximum distance from points to the modeled plane). This optional patch refinement procedure sacrifices the non-planar points to achieve better accuracy of the patch modeling for surfaces with high roughness and noise. Besides, the patches with too few points (e.g., < 10) or too scattered distribution should also be discarded.

5.1.1.2 Definition of Deformation Directions

In defined direction-based methods, the direction of deformation can be flexibly defined according to the prior knowledge (e.g., causes of deformation like gravity) or actual demands for geomonitoring tasks (e.g., the direction of interest like surface normal). Assuming that local patches deform along consistent directions respectively (as depicted in Section 3.1.1), a deformation direction can be defined or calculated for the entire single patch and imposed on each point within the patch. These deformation directions of all patches can be denoted by a set of direction vectors:

$$\underset{N_{patch}\times 3}{\vec{\boldsymbol{N}}} = [\boldsymbol{d}_{patch\_1} \ \cdots \ \boldsymbol{d}_{patch\_i} \ \cdots \ \boldsymbol{d}_{patch\_n}]^T, \tag{5.2}$$

where $\boldsymbol{d}_{patch\_i} = (d_{x\_i}, d_{y\_i}, d_{z\_i})$ denotes the deformation direction for a patch (i.e., for all points within this patch), and N_{patch} is the number of generated patches. The subsequent calculations of PB-M3C2 distances and associated *projected Std.* of each patch will be performed along these direction vectors, respectively.

Herein, the definition of deformation directions in PB-M3C2 is generalized and extended from that in standard M3C2 which is specialized to surface normals (with a fixed scale or multi-scale) or horizontal/vertical directions (Lague *et al.*, 2013). Figure 5.2 illustrates three ways to define the deformation direction for each patch in PB-M3C2: Pre-defined consistent direction, multi-scale normal direction, and patch-based normal direction.

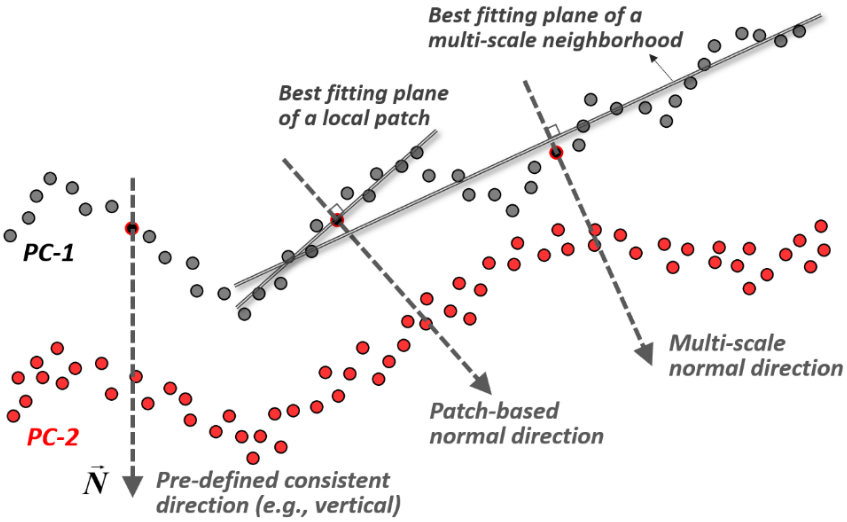

Best fitting plane of a multi-scale neighborhood
Best fitting plane of a local patch
PC-1
PC-2
Multi-scale normal direction
Patch-based normal direction
$\vec{N}$
Pre-defined consistent direction (e.g., vertical)

Figure 5.2: Three definitions of the deformation direction in PB-M3C2 (Yang and Schwieger, 2023).

Pre-defined Consistent Direction

The simplest way is to define a specific deformation direction imposed on the entire point cloud for the distance calculation of all points. The following aspects list some typical ways to define a consistent deformation direction:

- Vertical direction: when gravity is regarded as the cause of deformation, the motion of the monitored object is dominantly along the vertical direction. In this case, the direction vectors will be $(0, 0, 1)$ for all patches;
- Line-of-sight direction: when two-epoch point clouds are acquired by terrestrial laser scanner at the same station, deformations along the ranging direction of the scanner are more sensitive to be detected than that along the lateral direction. In this case, the direction vector is along the line between the measured point and the viewpoint (scanner center), and it can be denoted by $(\frac{x_i}{R}, \frac{y_i}{R}, \frac{z_i}{R})$, where (x_i, y_i, z_i) is the coordinate of the original measurement and $R = \sqrt{x_i^2 + y_i^2 + z_i^2}$ is the ranging distance;
- Direction of interest: the user-defined direction considering some prior knowledge of the deformation direction or demands of specific monitoring tasks. For example, the radial direction of a tunnel for monitoring its convergence, or a uniform direction along the global surface orientation for monitoring its out-of-plane movement.

Multi-scale Normal Direction

In the workflow of standard M3C2, assumed deformation directions are defined by local surface normals in 3D. A simple way is to define a fixed scale (radius) or *k*-nearest neighbors to yield neighboring points that are used for estimating a local surface normal as the orientation of the cylinder for each core point. It is optional to estimate these normals on PC-1, PC-2, or the average of both normal directions depending on the application. In geodetic monitoring, normals for distance calculations generally are estimated on PC-1 concerning the fact that geomorphic changes are more likely caused from the initial epoch to the subsequent epochs (Lague *et al.*, 2013).

However, the normal estimation by a fixed scale is not adaptive to the surface with high variation and roughness, leading to a potential overestimation of orthogonal distances between two surfaces. Therefore, a multi-scale normal estimation is proposed in standard M3C2 to prevent surface normal calculations from being affected by noise and roughness: The optimal normal scale is determined by choosing the scale with the smallest curvature over a range of scales defined by the user (Lague *et al.*, 2013).

Patch-based Normal Direction

The deformation directions defined by multi-scale normal estimation are smoother and more robust against measurement noise and high roughness than that by a fixed-scale normal estimation. The computing time, nevertheless, is considerably longer for its brute-force curvature computation. Besides, small deformations may be overestimated when the estimated multi-scale normal deviates too much from the actual local surface orientation.

Benefiting from the local planarity of generated patches, local surface normals can be directly derived based on the orientation of each patch by plane fitting. This method is desirable and sensitive for small out-of-plane deformations between two surfaces. In addition, the computational efficiency can be greatly improved owing to the absence of a neighboring-point search for normal estimation.

5.1.1.3 Deformation Quantification between Two Point Clouds

Similar to standard M3C2, the deformation quantification between two surfaces in PB-M3C2 is also achieved by calculating spatial distances along defined deformation directions (see Section 5.1.1.2). This section elaborates on the main differences between PB-M3C2 and standard M3C2 in distance calculation.

Using Prisms Instead of the Cylinders in M3C2

Figure 5.3(a) illustrates that a set of core points is generated by uniformly subsampling PC-1 in standard M3C2, then a cylinder is constructed by extending a spherical neighborhood of the core point to capture local neighbors of two point clouds. The cylindrical axis goes through each core point and is oriented to the defined normal direction. The distance between the average positions of two neighborhoods within the cylinder is regarded as the deformation value at the core point. Compared to C2C and C2M, the way of averaging local neighbors from the cylinder can effectively weaken the influence of measurement noise on the distance calculation, and the estimated local roughness allows a statistical significance test of the deformation empirically (Lague *et al.*, 2013). Besides, the distances between non-overlapping areas will not be calculated, as the related cylinder captures insufficient points. This can avoid yielding spurious deformations in the non-overlapping areas.

However, the cylinder will be highly overlapping when sampling high-resolution core points, resulting in an over-smoothing effect of calculated distances. For the areas with non-planar, discontinuous, or sharp structures (i.e., high roughness), the average position in the cylinder may deviate from the actual surface, leading to an incorrect deformation quantification. In addition, the empirical standard deviation approximated by local surface roughness is potentially overestimated, because the actual surface variation and measurement uncertainties are confused by the cylinder.

To solve the above limitations in standard M3C2, the proposed PB-M3C2 utilizes segmented planar patches to separate measurement uncertainties and real surface variations. In this case, cylinders with a constant scale are no longer suitable to capture the corresponding neighbors in the other point cloud due to the irregular shapes of generated planar patches (supervoxels). As presented in Figure 5.3(b), in PB-M3C2, polygons are generated based on the constructed convex hulls of planar patches from PC-1. Then a prism is constructed by extending the polygon along the defined deformation direction. Similar to the construction method of the cylinder in M3C2, the prism extends progressively towards PC-2 from the associated patch in PC-1. In PB-M3C2, the prism is extended in alternating directions until it incorporates the PC-2 surface, ensuring that only the closest area of PC-2 is captured. The maximum length of the prism should be set by the user to avoid endless extension by considering the assumed maximum deformation magnitude.

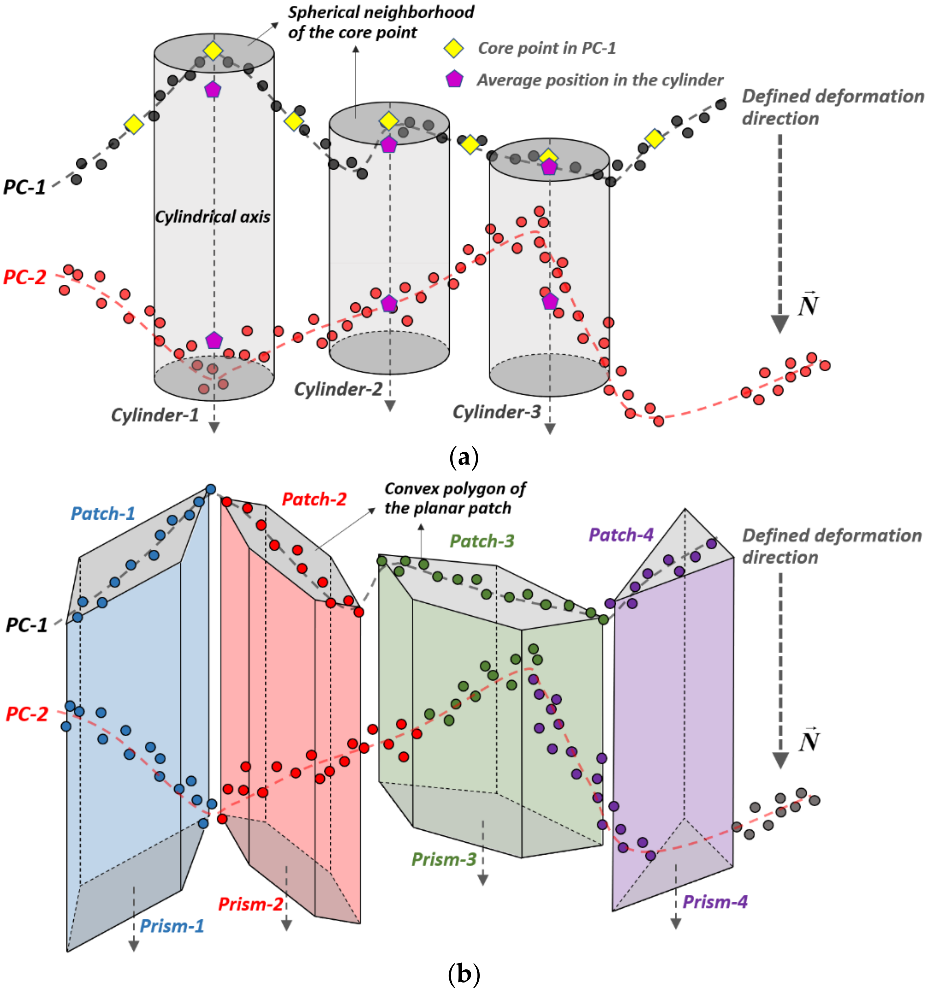

Spherical neighborhood of the core point
Core point in PC-1
Average position in the cylinder
Defined deformation direction
PC-1
Cylindrical axis
PC-2
$\vec{N}$
Cylinder-1
Cylinder-2
Cylinder-3
(a)
Patch-1
Patch-2
Convex polygon of the planar patch
Patch-3
Patch-4
Defined deformation direction
PC-1
PC-2
$\vec{N}$
Prism-1
Prism-2
Prism-3
Prism-4
(b)

Figure 5.3: The construction ways of cylinders in standard M3C2 (a) and prisms in PB-M3C2 (b) (Yang and Schwieger, 2023).

The construction process of polygons of generated patches is shown in Figure 5.4: Measured points are firstly projected on the best-fitting plane of the corresponding patch, then a 2D convex hull is constructed based on the projected points as a convex polygon that will be extended to form a prism. This process is performed on each patch. Hence gaps or overlaps may arise between these polygons, and inconsistent deformation calculations potentially exist at the same position in these overlapping areas. This issue and its possible solutions will be discussed in Section 5.1.3.2.

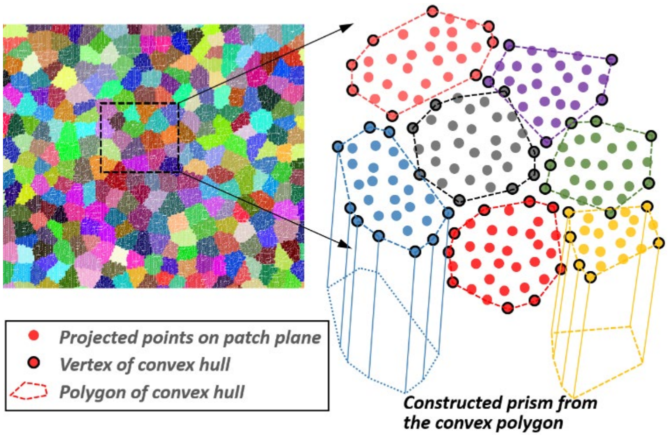

Projected points on patch plane
Vertex of convex hull
Polygon of convex hull
Constructed prism from the convex polygon

Figure 5.4: The construction of convex polygons of segmented patches (Yang and Schwieger, 2023).

Projecting Standard Deviations Along Defined Deformation Directions and Projecting Points onto Local Patch Planes

In standard M3C2, two subsets of points are intercepted by the defined cylinder of radius $d/2$ (d is called projection scale) and then projected onto the cylindrical axis, yielding two distributions on the axis. The means and standard deviations of the distributions will be used respectively as the average positions and local roughness ($\hat{\sigma}_1(d)$ and $\hat{\sigma}_2(d)$) of these two subsets of points. The distance of two average positions is calculated as the deformation magnitude at the associated core point (*M3C2 distance*), as shown in Figure 5.5(a).

In PB-M3C2, all points in the planar patch are firstly projected onto a locally fitted patch plane. As defined in Equation (5.1), the standard deviation ($\hat{\sigma}(patch)$) from plane fitting is used to approximate the local uncertainties for each patch ($\hat{\sigma}_1(patch\_i)$ and $\hat{\sigma}_2(patch\_j)$) instead of the local roughness employed in standard M3C2, as shown in Figure 5.5(b). Since these standard deviations only reflect the uncertainties in the direction of respective patch normals, they are projected along the defined deformation directions for the following calculation of LoDetection. The projected standard deviations ($\hat{\sigma}'(patch)$) can be derived by

$$\hat{\sigma}'(patch) = \frac{\hat{\sigma}(patch)}{\left|\cos\left\langle \boldsymbol{n}_{patch}, \boldsymbol{d}_{patch} \right\rangle\right|}, \tag{5.3}$$

where $\boldsymbol{n}_{patch}$ and $\boldsymbol{d}_{patch}$ are the normal vector of the patch and the defined deformation direction imposed on this patch (see Equation (5.2)), respectively. Particularly when using the patch-based normal direction as the deformation direction of each patch (i.e., the angle ($\theta_{\boldsymbol{d},\boldsymbol{n}} = \arccos\left(\frac{\boldsymbol{n}_{patch} \cdot \boldsymbol{d}_{patch}}{\left|\boldsymbol{n}_{patch}\right|\left|\boldsymbol{d}_{patch}\right|}\right)$) between $\boldsymbol{d}_{patch}$ and $\boldsymbol{n}_{patch}$ is 0° or 180°), $\hat{\sigma}'(patch) = \hat{\sigma}(patch)$. When $\theta_{\boldsymbol{d},\boldsymbol{n}}$ is close to 90°, $\hat{\sigma}'(patch)$ will become extremely large (increasing in multiples of $\cos^{-1}\theta_{\boldsymbol{d},\boldsymbol{n}}$) and deformations are nearly in-plane movements which PB-M3C2 cannot detect. Thereby, only deformations for the patches with $\theta_{\boldsymbol{d},\boldsymbol{n}} \leq 88°$ are calculated in PB-M3C2 ($\hat{\sigma}'(patch) \approx 30\hat{\sigma}(patch)$ when $\theta_{\boldsymbol{d},\boldsymbol{n}} = 88°$). The way of estimating $\hat{\sigma}(patch)$ and $\hat{\sigma}'(patch)$ for all points within a patch exactly corresponds to the local consistency assumption for measurement uncertainties in Section 3.1.1.

As the projected points captured by the prism in PC-2 may be located on different patch planes, these projected points are firstly used as average positions of PC-2, and further projected onto the corresponding patch plane in PC-1 along the reversed defined deformation direction. As shown in Figure 5.5(b), the distance between two corresponding projected points within the prism is calculated as the deformation magnitude at the associated projected point (*PB-M3C2 distance*). Again, the way of projecting the points in PC-2 captured by the prism onto the patch plane in PC-1 to calculate surface distances exactly corresponds to the local consistency assumption of deformation directions in Section 3.1.1. Since the actual surface variations are separated by generated planar patches, the projected points used for distance calculation are closer to the real surface, which is free from the influence of surface variability. Furthermore, the resolution of derived deformations will keep in line with the resolution of the original PC-2 owing to the projection of original measurements instead of the subsampled core points in standard M3C2, and resampling on the modeled patch planes could further improve the resolution of deformation analysis.

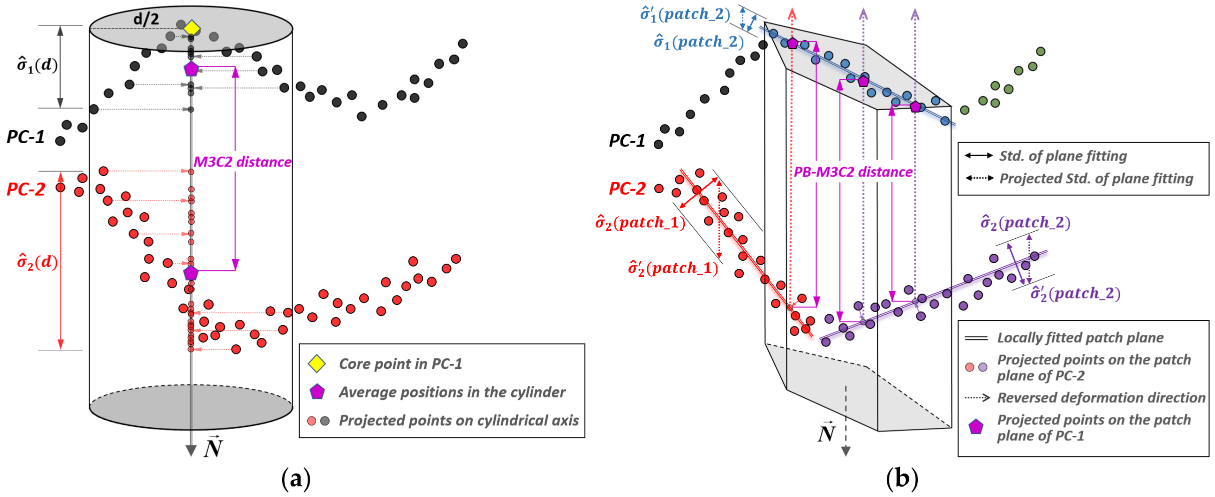

d/2
$\hat\sigma_1(d)$
PC-1
PC-2
$\hat\sigma_2(d)$
M3C2 distance
$\vec N$
Core point in PC-1
Average positions in the cylinder
Projected points on cylindrical axis
(a)
$\hat\sigma'_1(patch\_2)$
$\hat\sigma_1(patch\_2)$
PC-1
PC-2
$\hat\sigma_2(patch\_1)$
$\hat\sigma'_2(patch\_1)$
PB-M3C2 distance
Std. of plane fitting
Projected Std. of plane fitting
$\hat\sigma_2(patch\_2)$
$\hat\sigma'_2(patch\_2)$
Locally fitted patch plane
Projected points on the patch plane of PC-2
Reversed deformation direction
Projected points on the patch plane of PC-1
$\vec N$
(b)

Figure 5.5: The distance calculation between two surfaces by standard M3C2 (a) and PB-M3C2 (b) (Yang and Schwieger, 2023).

5.1.1.4 *Level of Detection with Lower Uncertainty*

In standard M3C2, the local roughness ($\hat\sigma_1(d)$ and $\hat\sigma_2(d)$) and number of points (n_1 and n_2) within the cylinder are used to estimate local uncertainties with respect to the calculated deformation (*M3C2 distance*). Based on a statistical *t*-test and the assumption of Gaussian distributed errors, a spatially variable confidence interval at a specified confidence level of P is defined as the LoDetection by Equation (5.4), incorporating a registration uncertainty $\hat\sigma_{reg}$ that is simply assumed spatially uniform. t_P is the quantile of the two-tailed *t*-distribution at the confidence level P with the degrees of freedom $f = n_1 + n_2 - 2$ (e.g., t_P equals 1.98 when $P = 95\%$ and $f = 100$).

$$LoDetection_P(d) = \pm t_P\left(\sqrt{\frac{\hat\sigma_1(d)^2}{n_1}+\frac{\hat\sigma_2(d)^2}{n_2}}+\hat\sigma_{reg}\right) \tag{5.4}$$

However, the estimated local roughness from standard M3C2 is comprised of not only measurement uncertainties but also the variation of real surface geometries. The latter (local surfaces that are non-planar or non-perpendicular to the cylinder's axis) will lead to a higher $\hat\sigma(d)$, thus increasing the LoDetection derived from Equation (5.4). The trade-off between accurate local average and low roughness by selecting the optimal projection scale d in M3C2 has been investigated by Lague *et al.* (2013) and DiFrancesco *et al.* (2020), while an adaptive cylinder scale is still challenging for surfaces with complex geometries and high roughness.

To decrease the uncertainties of calculated distances, the standard deviations from patch plane fitting are employed and projected along associated deformation directions thus deriving a more realistic LoDetection in PB-M3C2 as

$$LoDetection_P(patch\_i) = \pm t'_P\left(\sqrt{\frac{\hat\sigma'_1(patch\_i)^2}{n'_1}+\frac{\hat\sigma'_2(patch\_j)^2}{n'_2}}+\hat\sigma_{reg}\right), \tag{5.5}$$

where $\hat{\sigma}_1'(patch\_i)$ and $\hat{\sigma}_2'(patch\_j)$ are the projected standard deviations (calculated by Equation (5.3)) of associated patches in PC-1 and PC-2 respectively, n_1' and n_2' are the respective number of points in each patch, and t_P' is the quantile with the degrees of freedom $n_1' + n_2' - 2$. Deformations that are larger than LoDetection are regarded as significant.

If the correlation between the points within one patch (supervoxel) (as explained in Section 3.2.1) is considered in the patch modeling, n' in Equation (5.5) should be replaced by the effective number of measurements n_{eff} in Equation (3.6) or its simplified approximation n_{eff}' in Equation (3.7), where n_{SV} is replaced by n'. Besides, the effective degrees of freedom ($f_{eff} = n_{eff\_1}' + n_{eff\_2}' - 2$) should be considered to determine t_P'. If no correlation is assumed in the point clouds, the effective number of measurements equals the number of all points involved in the plane fitting. Obviously, the higher the correlation between points in a patch, the less the effective number of measurements, leading to a higher LoDetection.

5.1.2 Experimental Evaluation

To verify and evaluate the proposed PB-M3C2 algorithm with reference values with respect to deformation magnitudes, directions and deformation significance, three datasets are employed herein:

(1) Synthetic point clouds with well-defined geometries, noise, and deformations;
(2) Two-epoch scans of an indoor scene with simulated changes (short scanning distance);
(3) One scan of a rock cliff façade with defined deformations (long scanning distance), complemented by a simulated scan representing the second epoch.

In this comprehensive evaluation, the deformation analysis results by PB-M3C2 are systematically compared with the standard M3C2 algorithm. Notably, the datasets presented in Section 4.2 are not fully utilized in this section for the evaluation of PB-M3C2 due to the absence of reference deformation values in Dataset 1 and Dataset 2.

5.1.2.1 Synthetic Point Clouds with Defined Deformations (Dataset 3)

In this experiment, two-epoch synthetic point cloud surfaces are numerically generated, composed of small planes with specifically defined geometries and independent Gaussian noise (standard deviations are 5 mm and 2 mm for PC-1 and PC-2, respectively). As shown in Figure 5.6, five parts are constructed along the *Y*-axis to simulate different surface changes. Both surfaces are 0.5 m in width and 4 m in length, with an average point spacing of 1 cm in the horizontal direction. Since the reference deformations (ground truth) are derived by calculating the distances between two corresponding planes from their equations, absolute accuracies of both the quantified deformations and the estimated local uncertainties can be evaluated for this dataset. The point densities changing by the slope of sub-planes at different parts are intentionally made to validate the robustness of PB-M3C2 against such non-uniform point spacing. For the comparison with standard M3C2 under the same configurations, the deformation direction is defined to be uniformly vertical (along the *Z*-axis) for both methods. Hence the reference deformations are calculated along the vertical direction as well.

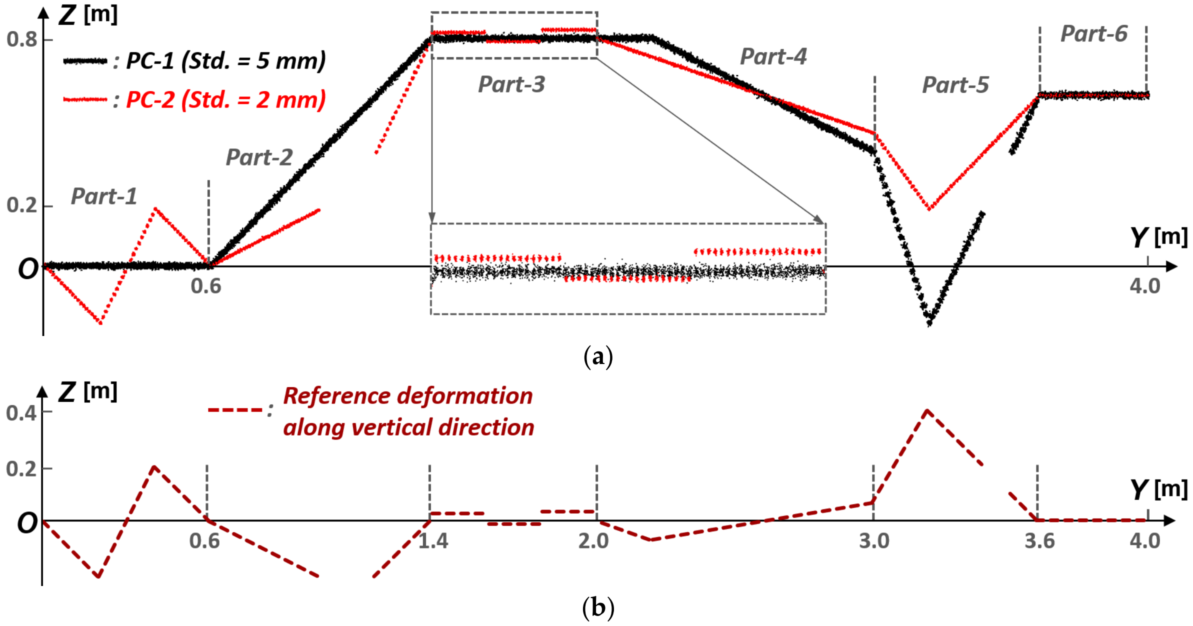


Figure 5.6: Synthetic two-epoch point clouds and related reference deformation (front view): (a) Two surfaces with defined deformations along the vertical direction (Part-1 contains sharp structures in PC-2; Part-2 includes a non-overlapping area in PC-2; Part-3 consists of three deformed planes with small vertical movements of +2 cm, -1 cm, and +3 cm; Part-4 represents gradual deformations by two intersecting planes; Part-5 contains a non-overlapping area in PC-1 and different sharp structures in both clouds; No deformation occurs in Part-6); (b) The magnitudes of reference deformations along the vertical direction are derived from plane equations (Yang and Schwieger, 2023).

Before evaluating the performance of deformation quantification by PB-M3C2 and standard M3C2, point cloud comparison results by C2C and C2M (the mesh is constructed by Poisson surface reconstruction (Kazhdan *et al.*, 2006)) are presented in Figure 5.7. Since both methods compute the minimum distances from one point cloud to another, the definition of the reference epoch (PC-1 or PC-2) may derive different results as shown in Figure 5.7. Hence, the known or assumed deformation directions cannot be considered in deformation calculations. Since the deformation direction in this dataset is defined to be uniformly vertical, distances derived by both C2C and C2M are underestimated in areas where surface orientations differ significantly in two epochs (see Part-1 and Part-5). Besides, statistical significance tests are unavailable for C2C and C2M due to the absence of uncertainties estimated for the distances. Compared to C2C, C2M can derive signed distances relative to the constructed mesh and display less noise owing to the average effects of meshing and free of the influence of point spacing (see Part-3 and Part-4).

Figure 5.8 presents the generated patches of two point clouds with good boundary separation by the better boundary preservation supervoxel segmentation (i.e., FEO-based) method (Lin *et al.*, 2018). The size of supervoxels is initially set as 30 times the average point spacing. After the segmentation process, however, these patches can adapt to the surface variations, resulting in irregular shapes and distributions. The sizes of these patches exhibit significant variability, ranging approximately from 0.1 m to 0.6 m.

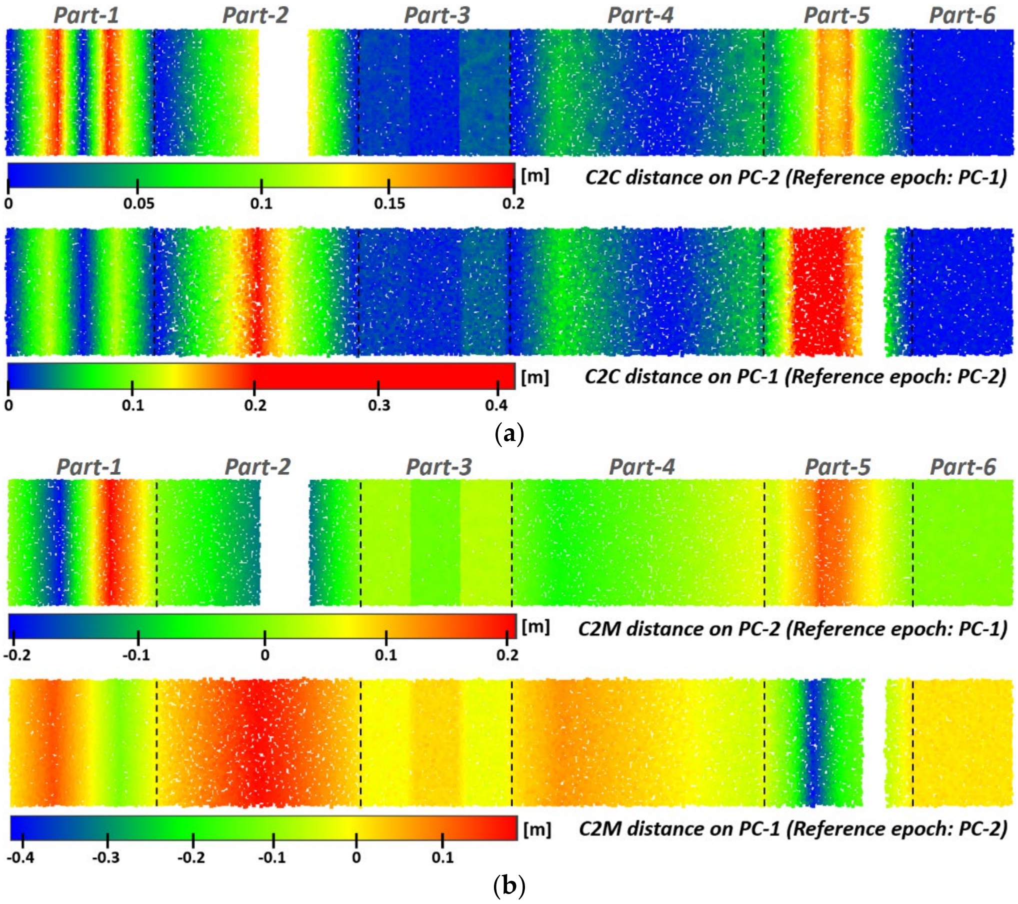


Figure 5.7: The quantified deformations (minimum distances) by C2C (a) and C2M (b).

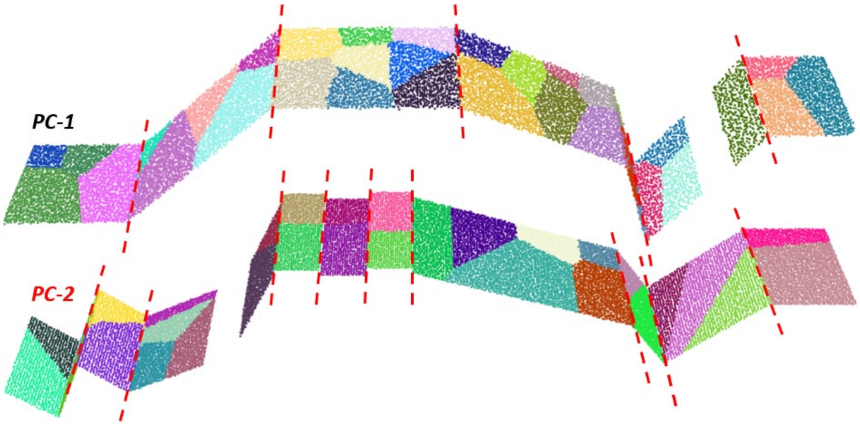


Figure 5.8: Generated patches by supervoxel-based over-segmentation (boundaries of the planes are highlighted by red dashed lines) (Yang and Schwieger, 2023).

Considering the advantages of M3C2 (depicted in Section 2.3.3), the performance of PB-M3C2 is compared solely with standard M3C2 in this dataset. Two diameters of cylinders (d = 10 cm and d = 20 cm) are used in the standard M3C2. Figure 5.9 gives the calculated vertical distances between two surfaces and associated absolute accuracy of two algorithms (all results are displayed on PC-1 for clarity). Generally, PB-M3C2 exhibits better accuracy than standard M3C2, particularly in areas with discontinuous boundaries and sharp structures (see Part-1~3 and Part-5).

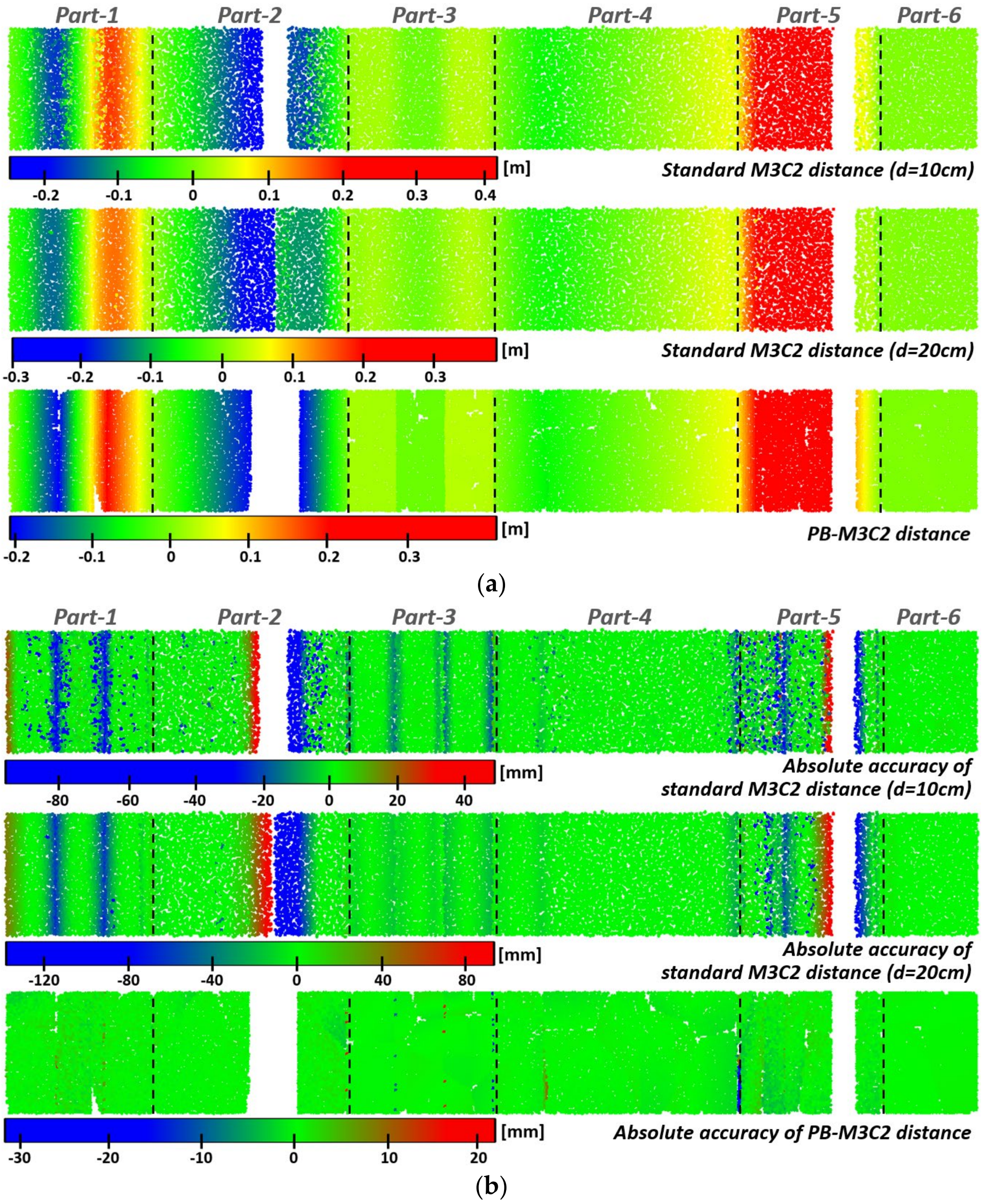


Figure 5.9: The quantified deformations by standard M3C2 and PB-M3C2 and their absolute accuracy: (a) Calculated distances between two-epoch point clouds; (b) Absolute accuracy of detected deformations by comparing with the reference (Yang and Schwieger, 2023).

It can be seen that in standard M3C2 a small cylinder enables an improvement in the distance accuracy near discontinuous areas but may yield noise in the distance calculation (see Part-1) due to insufficient points in the cylinder. Nevertheless, a larger cylinder diameter can cause incorrect over-smooth deformations (see the discontinuous boundaries in Part-3 in Figure 5.9(a)). Besides, the non-overlapping area in Part-2 is incorrectly involved in the distance calculation under a larger cylinder diameter. In contrast, these areas in PB-M3C2 are well preserved as the regions where deformations cannot be calculated, for these areas are not measured in both epochs.

Figure 5.10 shows the empirical uncertainties (i.e., *Std.* from local modeling and their projection along the vertical direction in PB-M3C2) of two-epoch point clouds estimated by two methods. The discontinuous and rough areas show higher uncertainties than planar areas in standard M3C2, and a larger diameter of the cylinder may yield higher and inconsistent uncertainties near the rough areas (see Figure 5.10(a) and (b)). The uncertainties estimated by PB-M3C2 are generally much lower than that by standard M3C2, and basically agree with the standard deviations of the defined Gaussian noise of respective point clouds (see Figure 5.10(c) and (d)). The slight difference in *Std.* between these patches in Figure 5.10(c) is mainly caused by their imperfect segmentation, while the variation of *projected Std.* in Figure 5.10(d) is due to the varying angles between the deformation direction (vertical in this case) and respective patch normals.

Due to the independently distributed Gaussian noise imposed on each point, there is no correlation within the patch (i.e., $K = 0$). Thus, significant deformations at a confidence level of 95%, as shown in Figure 5.11, can be detected by comparing to the LoDetection estimated by Equation (5.4) and Equation (5.5) in standard M3C2 and PB-M3C2, respectively. The reference of significant deformations is also given to evaluate the results of significance test from two methods. Again, with lower uncertainties, the test result of PB-M3C2 agrees better with the reference, even for the small deformations near the rough and discontinuous areas (see Part-3 in Figure 5.11).

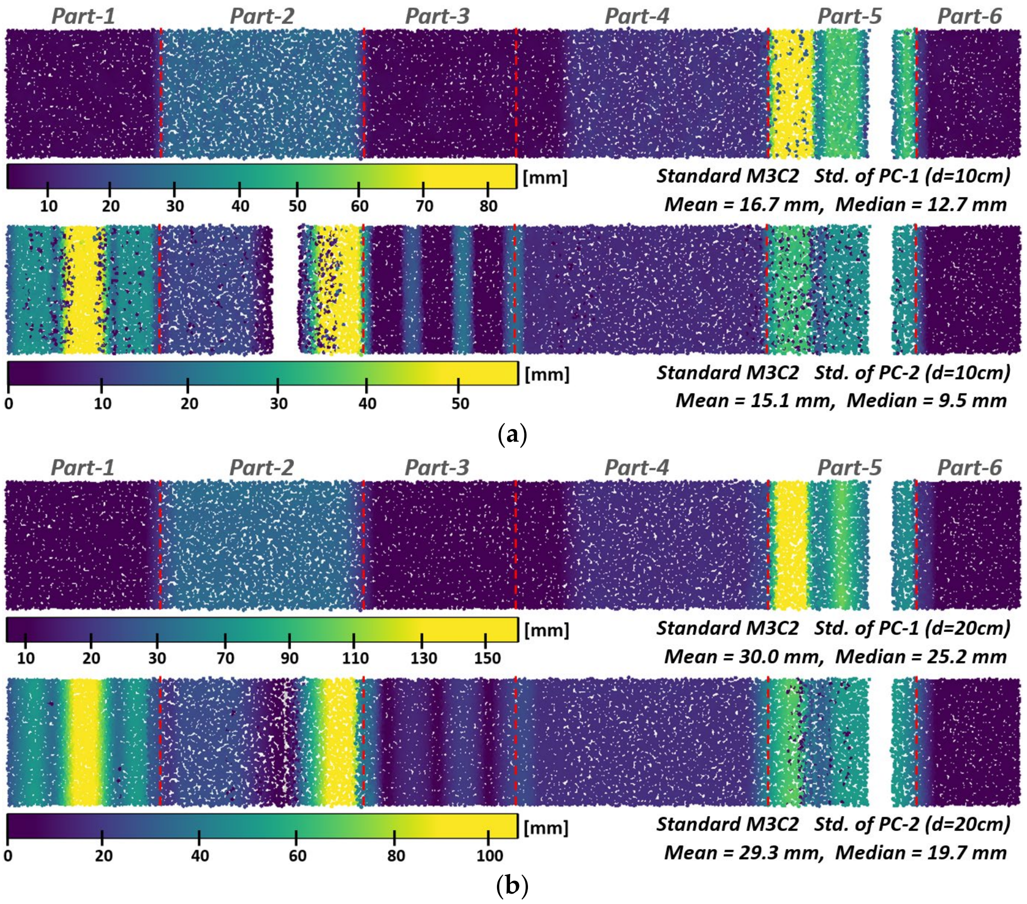


Figure 5.10: The estimated standard deviations of two-epoch point clouds by standard M3C2 (a, b) and PB-M3C2 (c, d) (Yang and Schwieger, 2023).

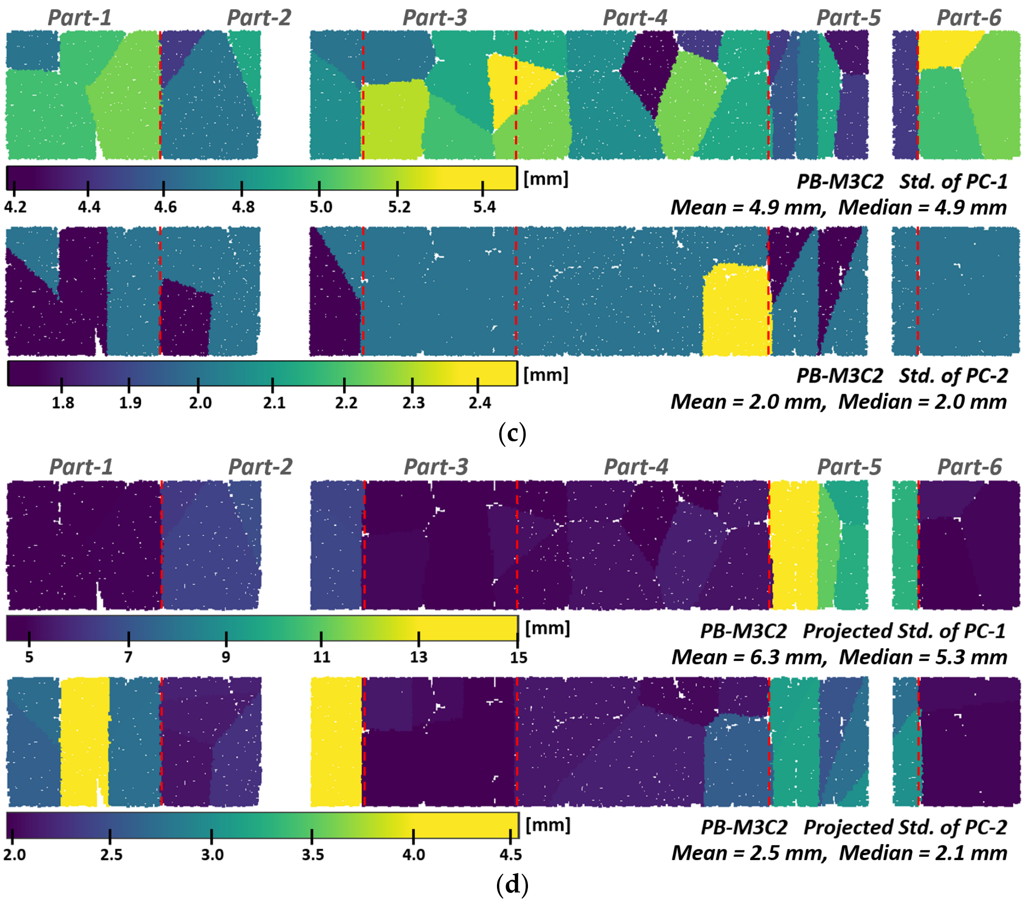


Figure 5.10: The estimated standard deviations of two-epoch point clouds by standard M3C2 (a, b) and PB-M3C2 (c, d) (Yang and Schwieger, 2023).

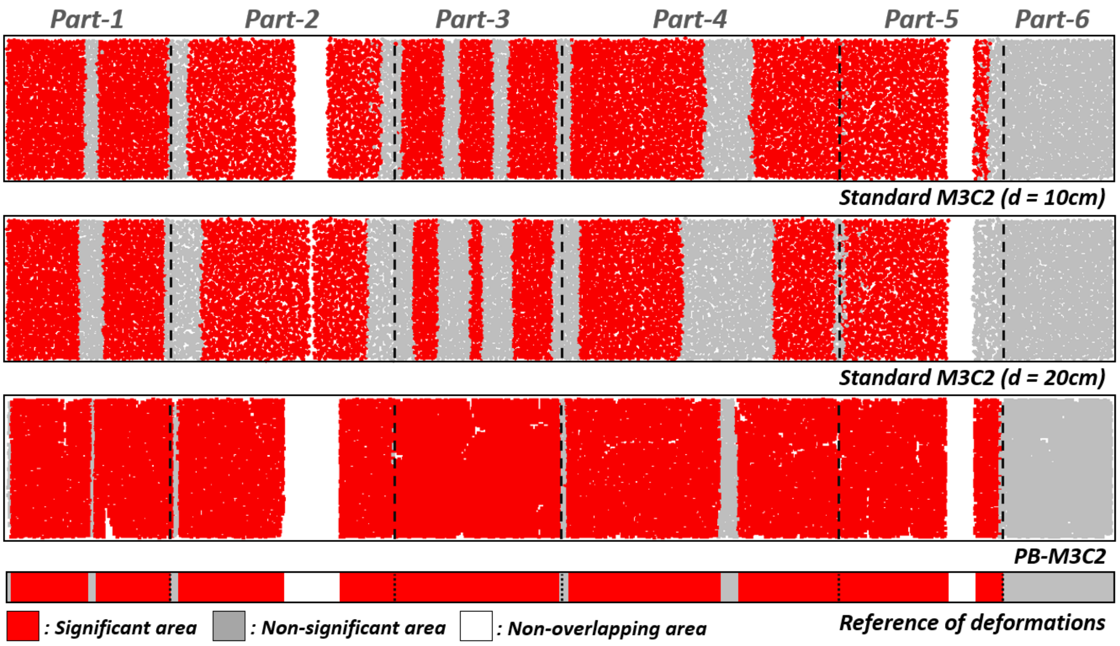


Figure 5.11: The significance test (P = 95%) of quantified deformations in dataset 3 (Yang and Schwieger, 2023).

5.1.2.2 Indoor Scene with Simulated Changes (Dataset 1)

As depicted in Figure 5.12, a sub-area of the indoor simulated scene used for evaluating the targetless registration methods in Section 4.2.2 is used again to validate the performance of PB-M3C2. Notably, the point clouds of the entire scene in Figure 4.6 are not used in this dataset because of the lack of reference (including deformation magnitudes and directions) for accuracy evaluation in all areas. In this partial area, objects with different shapes arranged on the ground are manually changed and deformed. Two acquired scans are precisely registered by six artificial targets (see Figure 4.6(a)) with an average registration error of 0.3 mm. The first scan only incorporates two flat wooden boards, while the second represents a deformed surface with complex geometries and high roughness. This specific design could qualitatively evaluate the derived deformations by comparing the visualized deformation magnitudes to the surface geometry of PC-2. Although it is difficult to obtain the actual deformation magnitude of each point (e.g., vertical distances between two surfaces) as the reference, this indoor experiment establishes a highly controlled situation where the stable and deformed parts are known, which can be used as the reference to evaluate the significance of quantified deformations.

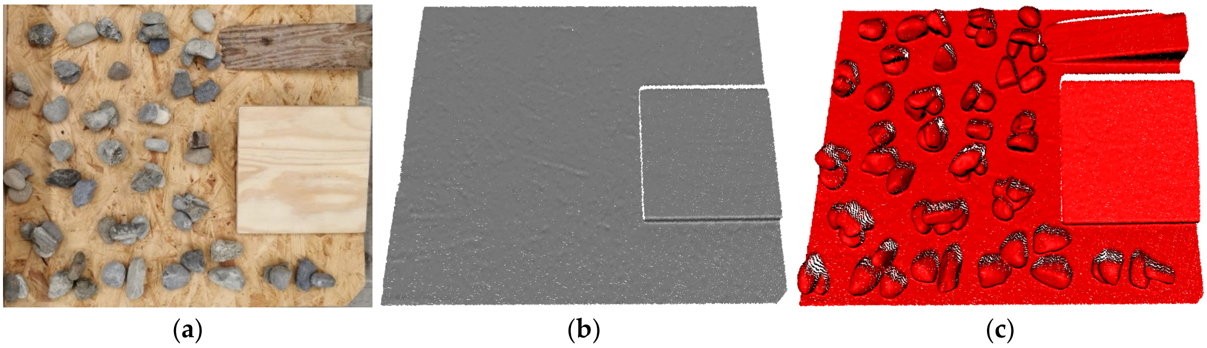
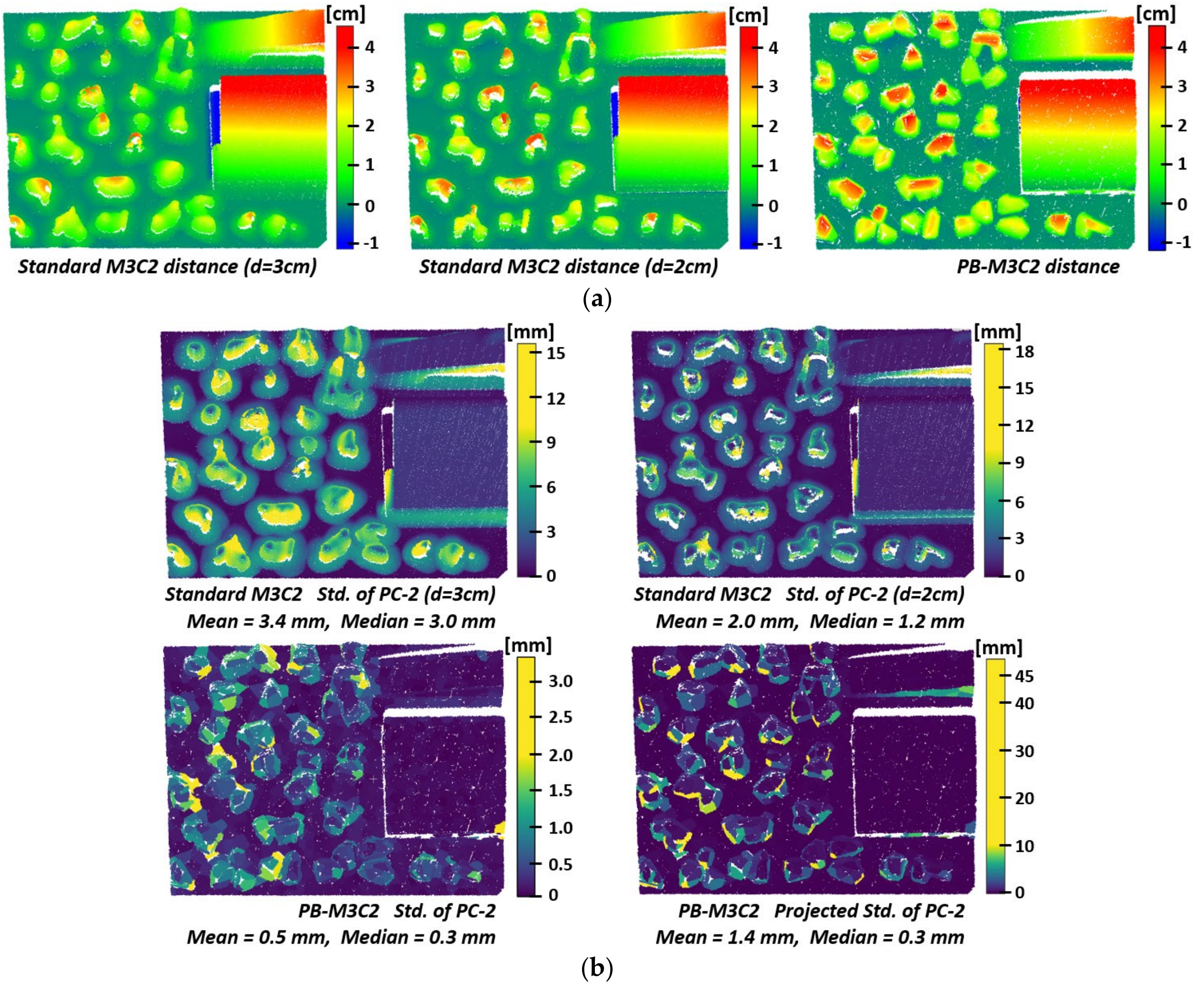


Figure 5.12: Experimental setup of the indoor scene (one subarea): (a) shows the artificial changes and deformations by placing irregular objects (gravels and a wedged wood block) on the ground and lifting a wooden board on one side; (b) and (c) are captured point clouds in epoch 1 (PC-1) and epoch 2 (PC-2) respectively (Yang and Schwieger, 2023).

The deformation direction in this dataset is again defined as vertical for comparison purposes. For clarity, quantified deformations and associated uncertainties are displayed on PC-2, as presented in Figure 5.13, so that the influence of surface variations (especially by the added objects) can be shown clearly. Since noisy distance quantifications or over-smoothing effects may arise with a too-small or too-large cylinder scale, d = 3 cm (the same as the initial size of supervoxels set in PB-M3C2) and d = 2 cm are set as trade-offs in standard M3C2. The calculated vertical distances near gravels' boundaries are underestimated by standard M3C2 (see Figure 5.13(a)) due to its average effect by the cylinder in rough areas, whereas the uncertainties are overestimated near these areas by both cylinder scales (see Figure 5.13(b)). In PB-M3C2, derived deformations are more consistent with the surface variations of PC-2 (see Figure 5.13(a)), and estimated uncertainties are more uniform and much smaller than that estimated by standard M3C2 except some steep areas of gravels with higher *projected Std.* (see Figure 5.13(b)).



Standard M3C2 distance (d=3cm)

Standard M3C2 distance (d=2cm)

PB-M3C2 distance

Standard M3C2 Std. of PC-2 (d=3cm)
Mean = 3.4 mm, Median = 3.0 mm

Standard M3C2 Std. of PC-2 (d=2cm)
Mean = 2.0 mm, Median = 1.2 mm

PB-M3C2 Std. of PC-2
Mean = 0.5 mm, Median = 0.3 mm

PB-M3C2 Projected Std. of PC-2
Mean = 1.4 mm, Median = 0.3 mm

Figure 5.13: The quantified deformations (a) and estimated standard deviations of PC-2 (b) by standard M3C2 and PB-M3C2 (Yang and Schwieger, 2023).

The statistical significance of quantified deformations (at a confidence level of 95%) and the reference by manually classifying the stable and deformed parts are presented in Figure 5.14. Due to the very short spatial distance (1.5 mm on average) and locally consistent material between neighboring points, high spatial correlation probably exists within each patch in this dataset. Hence, the LoDetection for each patch in PB-M3C2 is determined by Equation (5.5). The associated effective number of measurements is calculated by Equation (3.7), using a constant correlation coefficient (K). In addition, an average registration uncertainty ($\hat{\sigma}_{reg}$ = 0.3 mm) estimated from a target-based registration procedure is employed in calculating the LoDetection for both methods. For standard M3C2, boundaries between detected significant and non-significant areas are either over-smoothed (when d = 3 cm) or indistinguishable (when d = 2 cm), especially around the edges of gravels (see Figure 5.14(a) and (b)). This inconsistency with the gravel's contour is caused by the underestimated deformations and overestimated uncertainties in these areas. For PB-M3C2, some stable areas are incorrectly classified as significantly deformed without considering measurement correlations (see Figure 5.14(c)), while the distribution of significant deformations is closer to the reference (see Figure 5.14(d)) by adopting an empirical correlation coefficient of 0.65 from Kauker and Schwieger (2017). However, some steep areas can still not be detected as significant due to the overestimated *projected Std*. compared to their realistic uncertainties.

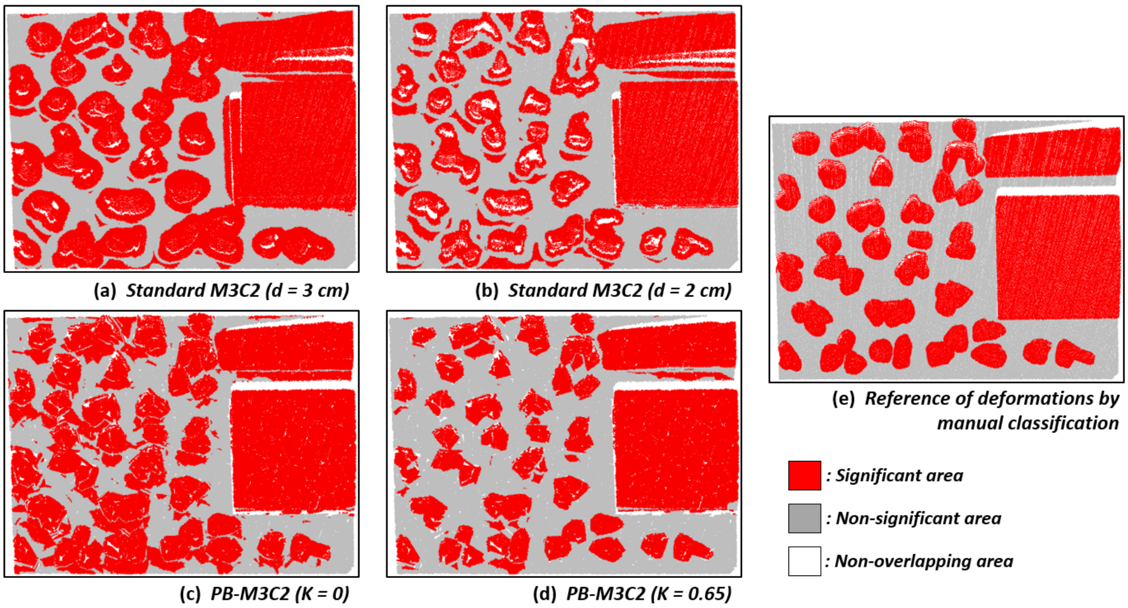
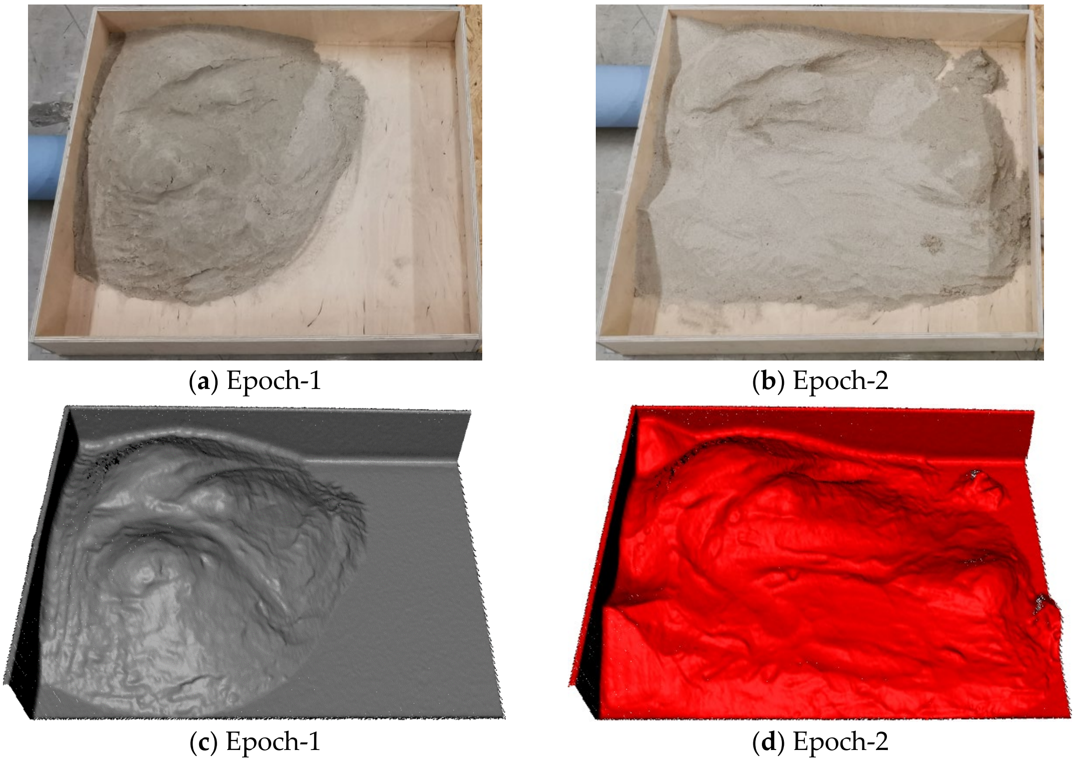


Figure 5.14: The significance test (P = 95%) of quantified deformations in dataset 1 (Yang and Schwieger, 2023).

For the purpose of comparison, the simulated shape deformations in another subarea within the scene in Figure 4.6, which features relatively smooth surfaces (as shown in Figure 5.15, depicting an area with partially reshaped sand), are also computed by both PB-M3C2 and the standard M3C2. It's worth noting that, in the absence of a reference for evaluating the accuracy of calculated deformations and their significance, the following section exclusively presents and compares the results obtained from the two methods.


(**a**) Epoch-1

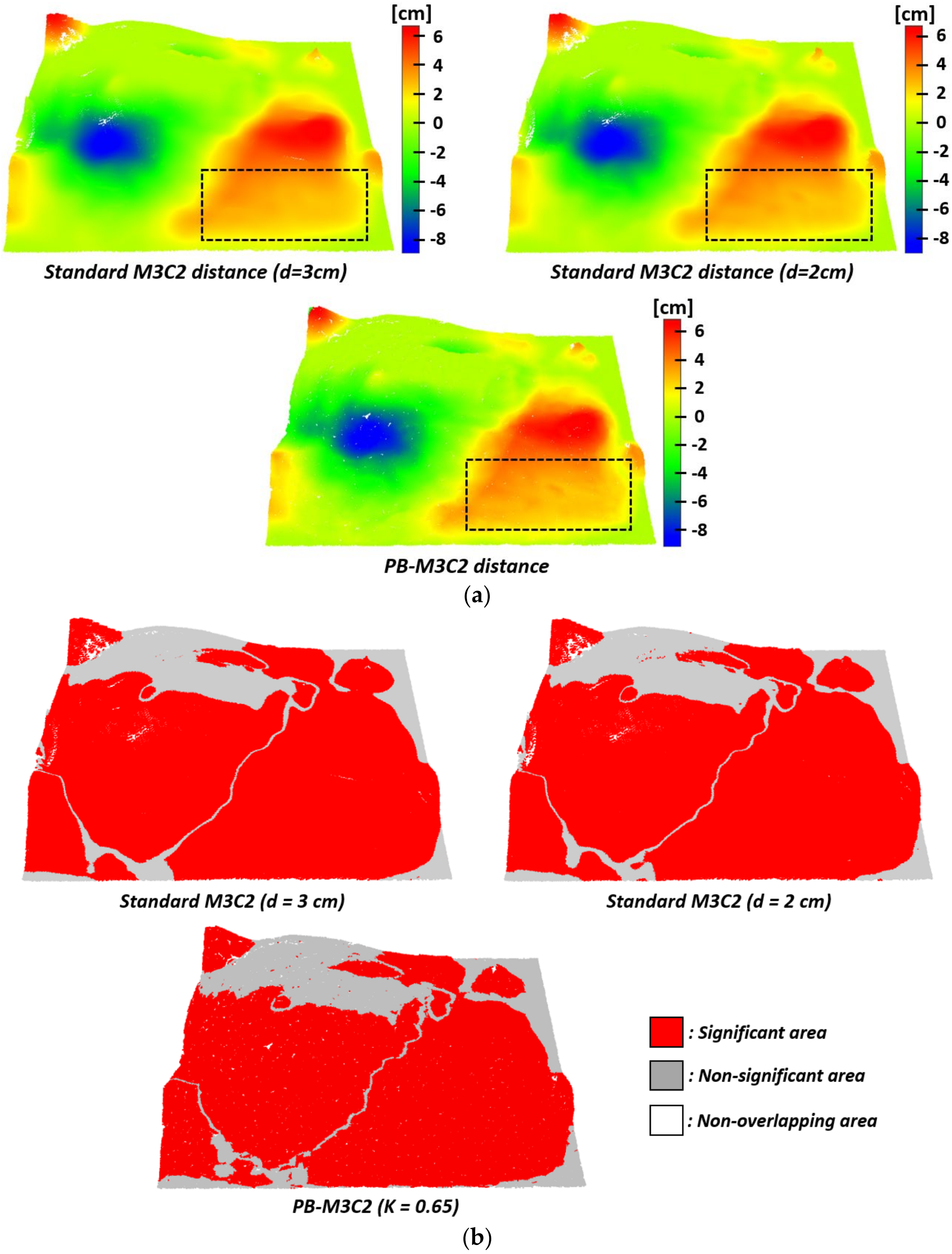
(**b**) Epoch-2

(**c**) Epoch-1

(**d**) Epoch-2

Figure 5.15: Experimental setup of the indoor scene (another subarea): (a) and (b) show the deformations by reshaping the sand in a stable wooden box; (b) and (c) are the captured point clouds in epoch 1 (PC-1) and epoch 2 (PC-2), respectively.

The deformation direction is first defined as vertical, and thus the vertical planes of the wooden box are not involved in the calculations. The quantified deformations and their significance (at a confidence level of 95%) are displayed on PC-2, as presented in Figure 5.16. Again, d = 3 cm (the same as the initial size of supervoxels set in PB-M3C2) and d = 2 cm are chosen in standard M3C2. The quantified distances and their statistical significance by standard M3C2 (with two different d) and PB-M3C2 are basically consistent, except for the over-smoothed distances arising in the lower right area (highlighted in the black dashed frames) by standard M3C2 with a larger d. Thereby, PB-M3C2 performs nearly the same as standard M3C2 in surfaces characterized by high smoothness and continuity.



Standard M3C2 distance (d=3cm)

Standard M3C2 distance (d=2cm)

PB-M3C2 distance

Standard M3C2 (d = 3 cm)

Standard M3C2 (d = 2 cm)

PB-M3C2 (K = 0.65)

Figure 5.16: The quantified deformations (a) and their significance (P = 95%) (b) of the selected sand area. The defined deformation directions are uniformly vertical.

The estimated *Std.* by both methods and the *projected Std.* by PB-M3C2 are also given in Figure 5.17. The magnitudes and spatial distributions of *Std.* by standard M3C2 with two diameters are generally similar, where the *Std.* in rough or steep areas with a larger diameter (d = 3 cm) are slightly higher than that with a smaller diameter (d = 2 cm). The *Std.* derived by PB-M3C2 are slightly smaller than that by standard M3C2, showing independence from surface variations. The *projected Std.* from PB-M3C2 are generally consistent with the *Std.* in smooth and continuous areas regarding both magnitudes and distributions while showing higher values in steep areas as highlighted in the red dashed frames.

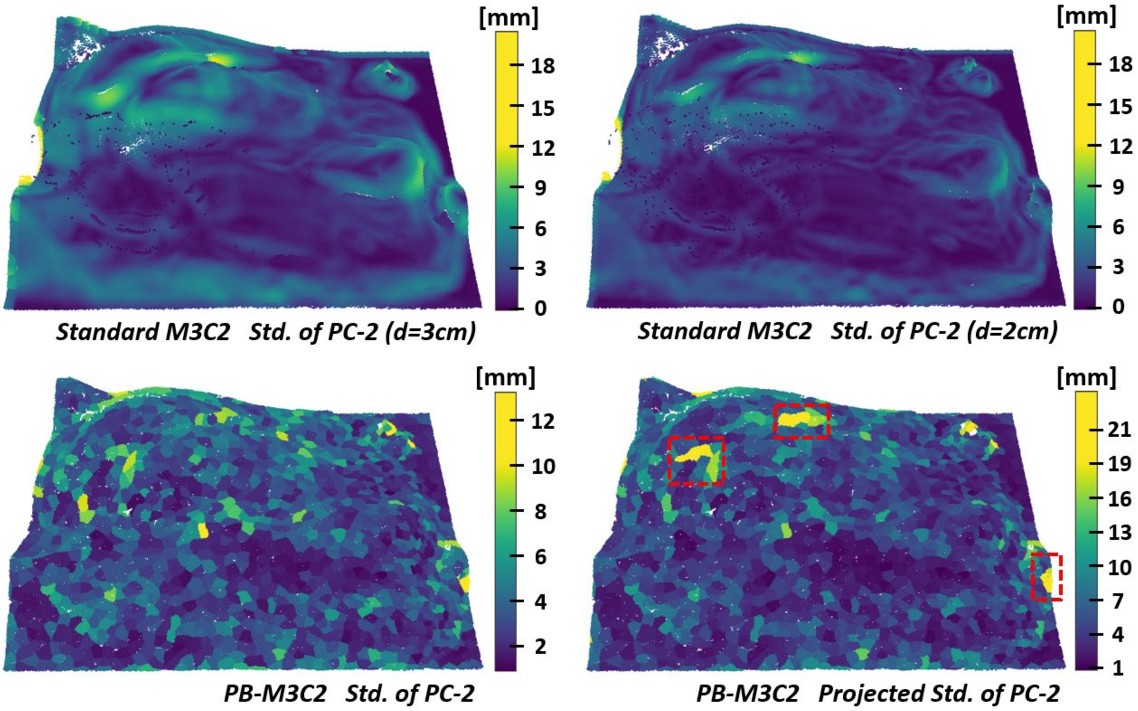
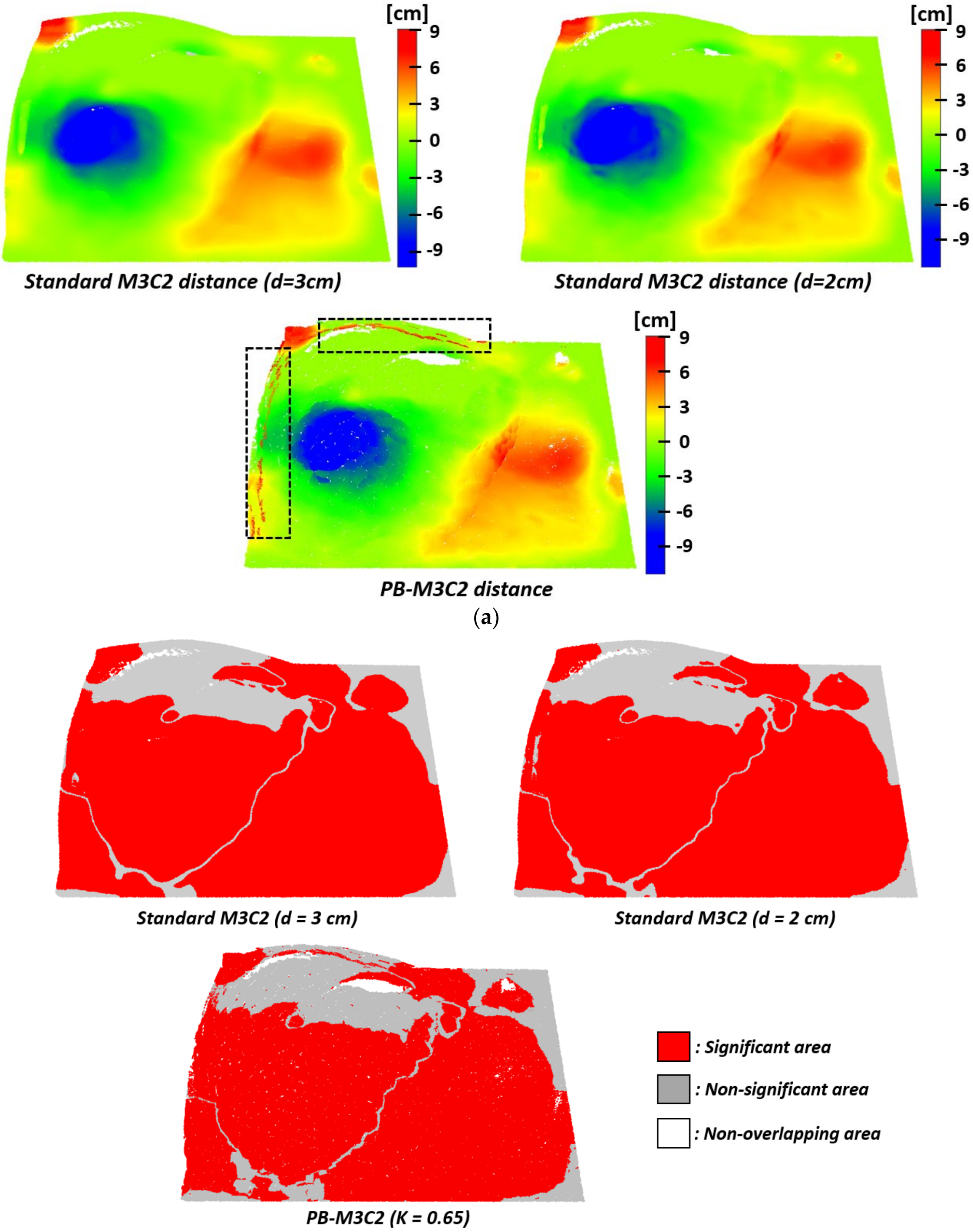


Figure 5.17: The estimated standard deviations of PC-2 by standard M3C2 and PB-M3C2.

To conduct a comparative analysis of the results under different definitions of deformation directions, Figure 5.18 presents the computed deformations and their associated statistical significance (at a confidence level of 95%), where patch normals serve as the defined deformation directions. Again, the calculated distances and associated significance by standard M3C2 and PB-M3C2 exhibit a consistent pattern. Furthermore, when comparing the results under two distinct deformation direction definitions, it is observed that the two methods yield similar deformation values in flat and horizontal areas. The minor disparities in the calculation results between the two methods can be attributed to the variations in deformation directions (i.e., local normals estimated by different neighbors) or different distance calculation approaches (i.e., using cylinders or prisms). Notably, some edge areas with larger variations (highlighted in the black dashed frames) show certain incorrect distance calculations by PB-M3C2. This is caused by the inaccurate estimations of local surface normals based on the segmented patches that represent elongated shapes within these areas. Therefore, the geometric characteristics of generated patches, like their shapes and planarity, may influence the determination of deformation directions when employing the patch-based normal direction. Given these findings, it is advisable to consider alternative deformation direction definitions, such as multi-scale normal directions (as outlined in Section 5.1.1.2), to improve the local normal estimation.



(a)

(b)

Figure 5.18: The quantified deformations (a) and their significance ($P = 95\%$) (b) of the selected sand area. The defined deformation directions are the estimated normals of respective patches.

5.1.2.3 Façade of a Rock Cliff (Dataset 4)

To verify the applicability of PB-M3C2 on 3D point clouds at different scales, one TLS point cloud of a rock cliff façade and a simulated scan as the second epoch are used to evaluate the algorithms' performance in natural scenes (Hassan *et al.*, 2022). The rock cliff surface with high variation and roughness is shown in Figure 5.19. Since no significant deformations occurred over monitoring epochs, the second-epoch surface is generated by defining a certain translation of the whole surface oriented to its normal to simulate small deformations (i.e., movements in this case) on complex topographies. This translation direction is taken as the defined deformation direction in this dataset for both standard M3C2 and PB-M3C2. The defined translation magnitude is used as the reference to evaluate the absolute accuracy of quantified deformations. It should be noted that some overhanging areas exist in the selected rock cliff façade, and approx. 8% generated patches from these areas are nearly parallel to the defined translation direction (i.e., $\theta_{d,n} > 88°$). These areas (approx. 8% of the selected façade) are not considered in the following deformation calculation, as explained in Section 5.1.1.3.

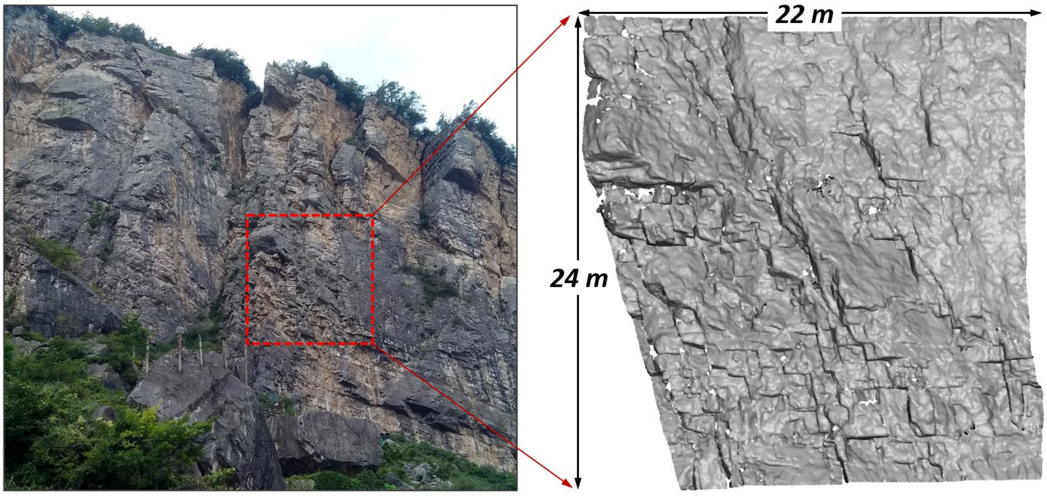


Figure 5.19: The scale and surface geometry of the selected rock cliff façade (Yang and Schwieger, 2023).

As shown in Figure 5.20, the estimated uncertainties (*Std.* and *projected Std.*) and associated LoDetection by both methods are presented in two histograms. The distribution of *Std.* by PB-M3C2 is concentrated near the mean of 13 mm, potentially indicating the intrinsic random measurement noise. The mean and median of *projected Std.* are slightly higher than the *Std.* from patch plane fitting but still concentrated and much smaller than the *Std.* by standard M3C2 (see Figure 5.20(a)). The associated LoDetection of two methods shown in Figure 5.20(b) directly provide the ratio of significant deformations among calculated distances. For example, if the defined translation (i.e., simulated deformation) is 3 cm, almost all quantified deformations will be tested as significant by PB-M3C2 (larger than nearly all LoDetection). Using standard M3C2, however, the areas with a LoDetection larger than 3 cm are erroneously regarded as non-significantly deformed.

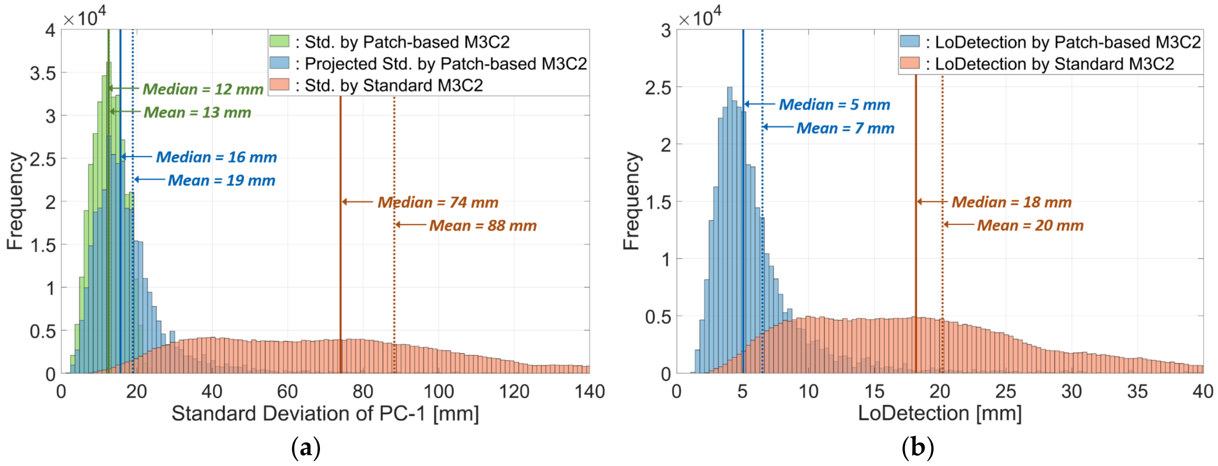


Figure 5.20: Histograms of standard deviations (a) and LoDetection (b) by standard M3C2 and PB-M3C2 (Yang and Schwieger, 2023).

The visualized spatial distributions of LoDetection are depicted in Figure 5.21(a). It can be seen that the highly rough areas display much higher LoDetection in standard M3C2, and thus small deformations are unable to be detected in these regions. In contrast, PB-M3C2 shows a more random spatial distribution of LoDetection, and the variation of surface topographies has a minor influence on it, except for some overhanging areas that are almost parallel to the translation direction.

Figure 5.21(b) presents the statistical significance of quantified deformations by both methods, where a 3 cm translation is simulated to generate the second-epoch scan. PB-M3C2 has significantly detected 98.5% deformations on the surface. However, 21.5% of deformations near the highly rough areas are incorrectly tested as non-significant by standard M3C2. This result exactly verifies the inference based on the histogram of LoDetection demonstrated in Figure 5.20(b).

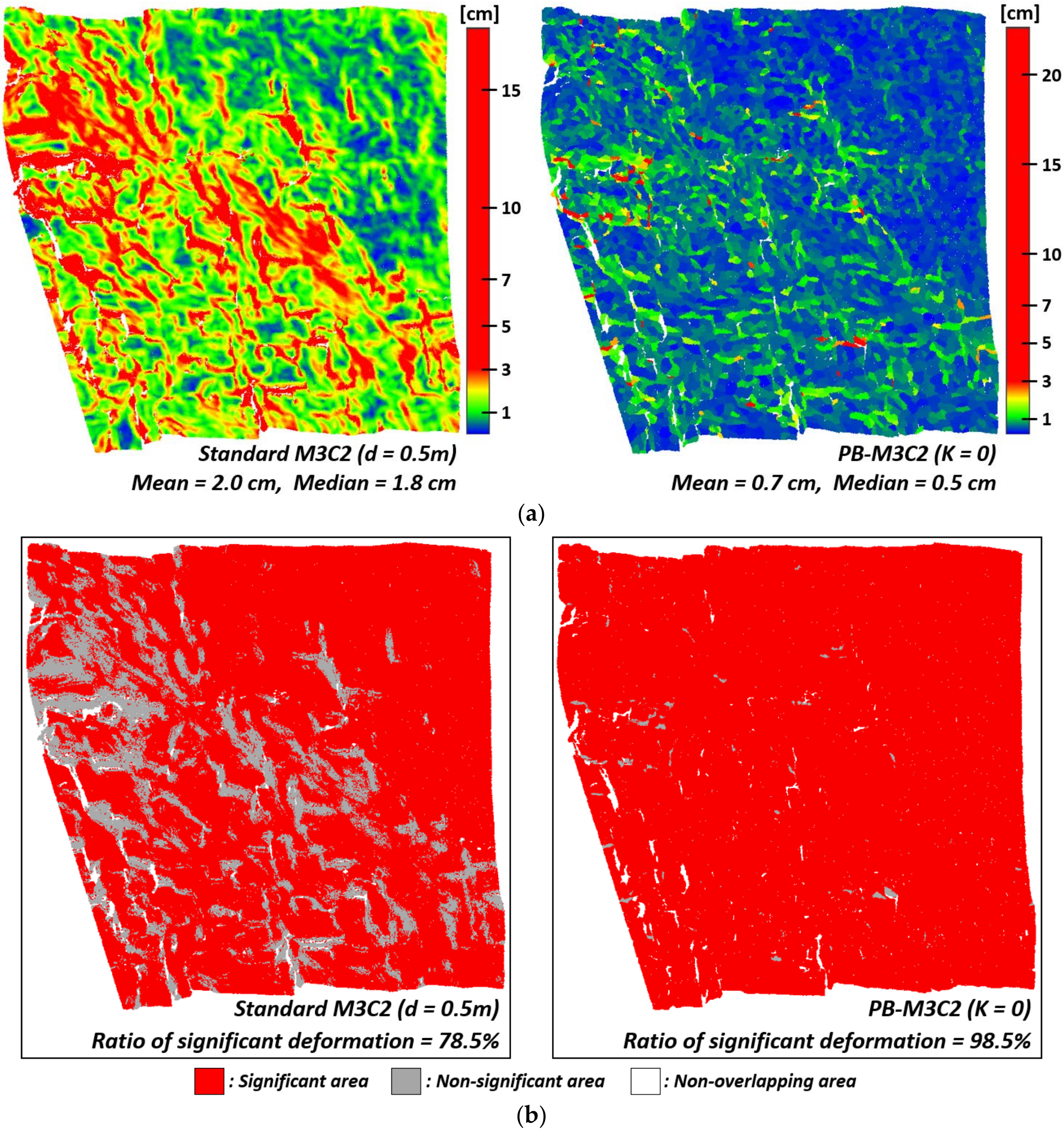


Figure 5.21: The spatial distributions of LoDetection (a) and the significance test ($P = 95\%$) of quantified deformations (b) by standard M3C2 and PB-M3C2 (Yang and Schwieger, 2023).

To evaluate the absolute accuracy of deformations quantified by traditional approaches (C2C, C2M and standard M3C2) and the proposed method (PB-M3C2), deformations of three different magnitudes (3 cm, 6 cm and 10 cm) are simulated by simple surface translations, including the case of no deformation (0 cm). Then the distributions of detected deformations are analyzed. To simulate the differences of laser scans between two epochs, the second-epoch point cloud is subsampled and two levels of additional Gaussian noise (with standard deviations of 5 mm and 10 mm) are imposed on each measurement.

To quantitatively evaluate the proximity between the calculated deformations and the translation value (as the reference), herein only the percentage of deformations falling within a specific interval (±5 mm and ±10 mm are chosen respectively considering the imposed Gaussian noise) near the reference is calculated. As shown in Figure 5.22, when no deformation occurs, C2C shows the worst results due to the influences of measurement noise and point spacing. C2M overcomes the impact of non-uniform point spacing but is still sensitive to measurement uncertainties. Standard M3C2 and PB-M3C2 show much higher ratios of correct deformation detections (i.e., no deformation), even though both methods have not completely eliminated the influence of imposed noise. When the magnitude of simulated deformation is increased, the distribution of detected deformations by PB-M3C2 is closest to the reference compared to the other three methods (more than 95% of derived deformations are within the specified interval in all cases). The second-best result is from standard M3C2. Since C2C and C2M employ the nearest correspondence for distance calculation without considering the deformation direction, the derived deformations are generally smaller. Compared to C2M, slightly more deformations detected by C2C are actually caused by the inconsistent point spacing (mistakenly quantified as deformations) in two point clouds. In contrast, C2M can overcome the influence of varying point spacing by using the point-to-mesh distance. Thereby, for point clouds with larger point spacing, the computed deformations by C2M are smaller than those by C2C.

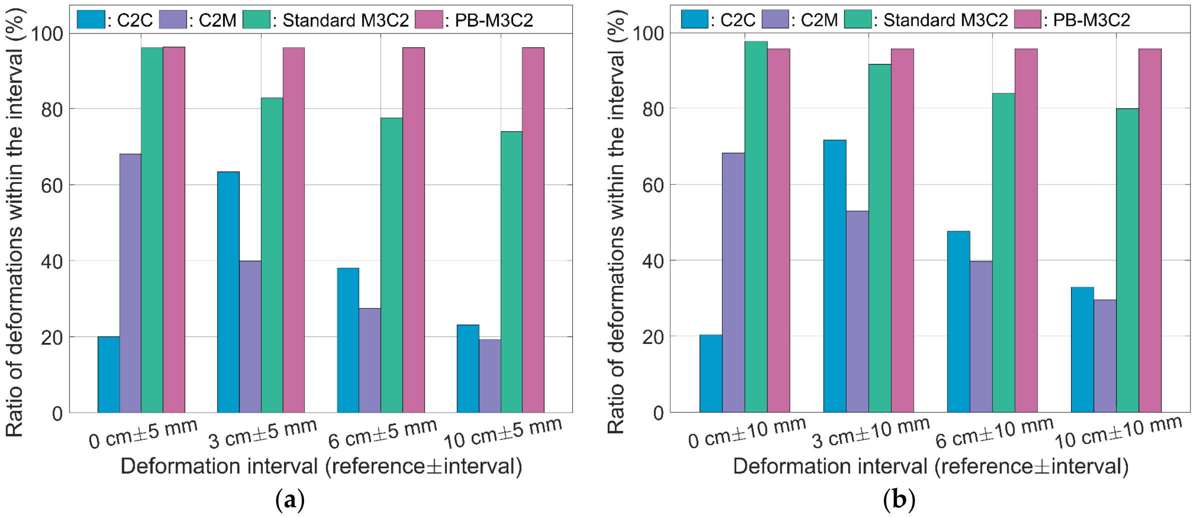


Figure 5.22: The ratio of deformations within specified intervals near the reference: (a) Imposing standard deviation of 5 mm; (b) Imposing standard deviation of 10 mm (Yang and Schwieger, 2023).

5.1.3 Discussion

5.1.3.1 Main Advantages of Patch-based M3C2

The proposed PB-M3C2 basically inherits the underlying principle and framework of standard M3C2 as well as its advantages (Lague *et al.*, 2013):

(1) Comparing two point clouds is directly conducted on the original measured points, and complex surface modeling or meshing is not required;
(2) The distance calculation of the non-overlapping areas between two surfaces is avoided, enabling the method to be more robust against missing data (due to changes in occlusion or scanning configuration);
(3) The defined deformation directions (using surface normals in standard M3C2) are used to compute signed distances;
(4) Average positions are estimated for distance calculation by local modeling (using cylinders in standard M3C2) against measurement uncertainties and the influence of point spacing variations;
(5) The statistical significance test is introduced by defining a confidence interval for quantified deformations merely based on the point cloud data (using local roughness in standard M3C2).

By using prisms constructed from planar patches instead of the cylinders in standard M3C2, the derived local uncertainties are drastically reduced in PB-M3C2, thereby enabling more robust detection of small-magnitude deformations. Furthermore, in PB-M3C2, the utilization of points projected onto the corresponding patch plane for distance calculations prevents excessive averaging of local neighbors within the cylinder, thus enhancing the accuracy of quantified deformations, particularly on highly rough surfaces. It should be mentioned that for smooth and continuous surfaces, there is little difference in the performance of distance quantification between the standard M3C2 and PB-M3C2.

The local consistency assumption and local modeling of planar patches allow PB-M3C2 to estimate empirical standard deviations of the point cloud, supposing that the measurement noise is subject to Gaussian distribution in 3D. This may provide an alternative for empirically determining a simplified stochastic model of the 3D point cloud.

5.1.3.2 Influence of the Patch Quality on Deformation Analysis

The quality of generated patches plays a crucial role in PB-M3C2, including the correctness of boundary preservation, patch shapes, dimensions, etc. Due to the surface variation and measurement noise, the shape and dimension of generated patches are typically inhomogeneous, or the boundaries may not be completely separated in a patch (i.e., imperfect segmentation). This is particularly evident at the boundaries with small curvatures and slight discontinuities. As a result, these segmentation results may cause differences in estimated uncertainties among patches, even when the standard deviations of the underlying points are identical, as illustrated in Figure 5.10(c). Furthermore, within these imperfect segments, distance calculations are prone to be inaccurate if the projected points deviate from the actual surface, as observed in the connection area between Part-4 and Part-5 in Figure 5.9(b)). Therefore, good-quality patches should exclude discontinuities and keep similar shapes.

The prism constructed based on a convex hull requires the shape of the generated patch to approximate a convex polygon. Otherwise, overlaps may arise between adjacent polygons and associated prism spaces, potentially leading to inconsistent deformation values calculated in the overlapping areas of adjacent convex polygons (see Figure 5.4). Compared to other supervoxel-based segmentation algorithms, the patches generated by the methods using a fusion—exchange (or merge—swap) optimization framework (Lin *et al.*, 2018; Xiao *et al.*, 2020) basically show a shape pattern of a convex polygon.

5.1.3.3 Current Limitations and Potential Improvements

PB-M3C2 is designed to quantify spatial distances along defined directions between two point cloud surfaces and thus cannot detect in-plane movements or deformations parallel to the surface. The distance may not agree with the magnitudes of the real displacement when the defined deformation directions deviate from the actual deformation directions. Other approaches should be adopted as alternatives for detecting such changes (e.g., feature-based methods).

Similar to standard M3C2 and CD-PB M3C2, the proposed PB-M3C2 also empirically estimates the standard deviation of a selected local area as the uncertainty for determining LoDetection. Since the local modeling of each patch is based on the local consistency assumption, equal weight is imposed on each point for the least squares-based plane fitting. However, realistic uncertainties of the point cloud may not be subject to a strict Gaussian distribution and thus not agree with the empirical estimation. Moreover, systematic errors are not considered in PB-M3C2. In these cases, locally modeled patch planes would deviate from the actual surface, thus leading to inaccurate distance calculations. Possible improvements by introducing a more realistic stochastic model for the point cloud can be considered to improve the modeling accuracy (see Section 6.2).

A spatially uniform registration error ($\hat{\sigma}_{reg}$) adopted in this article is not reasonable to calculate LoDetection, as the registration accuracy is usually spatially variable due to the uncertainties of targets or imperfections of registration algorithms (Lague *et al.*, 2013). Further improvements can integrate anisotropic registration uncertainties imposed on each point into LoDetection by using error propagation from the VCM of transformation parameters ($\hat{\sigma}_{reg}$ in the LoDetection of an individual patch can be approximated by using the registration uncertainty at the patch centroid).

As discussed in Section 5.1.3.2, the patches with incomplete boundary separation will yield inaccurate patch planes and overestimated local uncertainties. Besides, overlapping polygons from the convex hulls of adjacent patches make deformation calculations in these areas ambiguous. In this case, a combination of convex and concave polygons can be considered to weaken this impact (Wei, 2008).

In the workflow of PB-M3C2, the modules of patch generation and definition of deformation direction could be further improved independently. For example, with the rapid progress of deep learning, learning-based patch-generation methods could be integrated into the proposed pipeline (Landrieu and Boussaha, 2019; Hui *et al.*, 2021). In addition, more advanced approaches can also be introduced for the automatic and reliable determination of deformation directions (Williams *et al.*, 2021).

5.2 Geometric Primitive-based Deformation Analysis

The extracted geometric primitives from point clouds provide characteristic parameters that can represent the positions, orientations, and shapes to a certain extent. Hence these parameters could be further used to validate the rigidity and calculate deformations between epochs. The overall workflow of geometric primitive-based deformation analysis for 3D point clouds is presented in Figure 5.23. The segmentation part using a coarse-to-fine framework has been elaborated in Section 3.2.2. Notably, small planar patches generated by supervoxel-based over-segmentation (see Section 3.2.1) can partially or entirely occupy the existing planes in the point cloud. Nevertheless, these supervoxels may split a large plane (as one geometric primitive) into two or more parts and thus break the completeness of extracted geometric primitives (as explained in Section 3.2.2). Therefore, supervoxels are not used to generate plane primitives in the workflow.

After segmentation, parameter estimation should be carried out on each separated geometric primitive based on a Gauss-Markov model (for estimating approximate parameters) or a rigorous Gauss-Helmert model, along with deriving associated standard deviations. The corresponding primitives in other epochs can be found by comparing the shape parameters (e.g., the sphere radius) or by spatial distances (i.e., finding the nearest one if deformations are small). Deformations are then estimated on the basis of estimated parameters. These parametric deformations could be spatial distances (displacements), angular changes (orientation differences), volumetric increases/decreases, and so on. Eventually, a statistical significance test can be conducted for the estimated deformation magnitudes based on the derived stochastic information of relevant parameters.

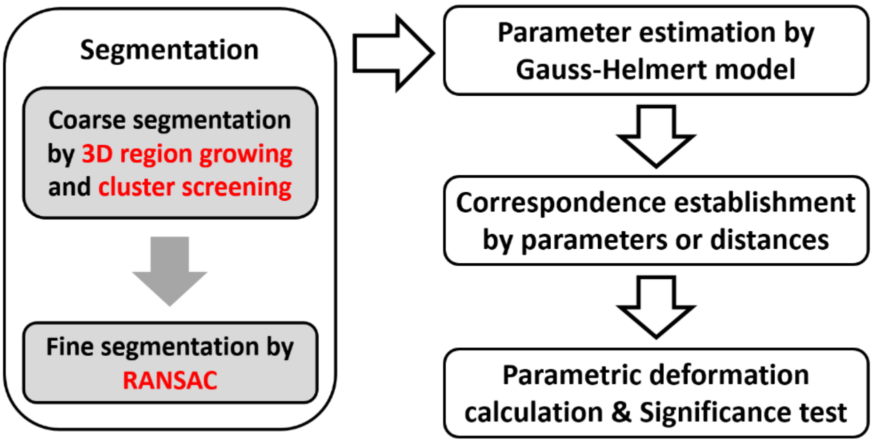

Segmentation
Coarse segmentation by 3D region growing and cluster screening
Fine segmentation by RANSAC
Parameter estimation by Gauss-Helmert model
Correspondence establishment by parameters or distances
Parametric deformation calculation & Significance test

Figure 5.23: The workflow of geometric primitive-based deformation analysis for 3D point clouds.

5.2.1 Parameter Estimation of Geometric Primitive by Gauss-Helmert Model

Parameter estimation for extracted and matched geometric primitives can be conducted to perform a parameter-based deformation analysis. The selected parameters of geometric primitives to be estimated should meet the demands of geomonitoring tasks. For example, the angle between normal vectors of the plane in two monitoring epochs should be estimated to quantify the change of plane orientations; the difference of the radius of a cylinder in two epochs could be estimated to indicate the change of its dimension or volume; the Euclidean distance of the center of a sphere in two epochs should be estimated to calculate the change of its position.

To obtain a rigorous solution of geometric primitives' parameters, the Gauss-Helmert model can be adopted under an iterative process (Lenzmann and Lenzmann, 2004; Niemeier, 2008). The linear or nonlinear observation equation of the geometric primitive can be written as

$$F(\underset{n\times 1}{\boldsymbol{L}}+\underset{n\times 1}{\boldsymbol{V}},\underset{u\times 1}{\boldsymbol{X}})=\underset{n\times 1}{\boldsymbol{0}}\,, \tag{5.6}$$

where $\boldsymbol{V}$ is the improvement vector for the observation vector $\boldsymbol{L}$ (the observations of point clouds are the 3D coordinates of each measured point); $\boldsymbol{X}$ contains the parameters to be estimated; n and u are the number of observations and parameters.

The model matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ are derived by a linearization of Equation (5.6) and denoted as Equation (5.7) and Equation (5.8) respectively, where $\boldsymbol{V^o}$ are the initial improvements and $\boldsymbol{X^o}$ are the approximate parameters that can be simply obtained by a linearized Gauss-Markov model.

$$\boldsymbol{A}(\boldsymbol{V},\boldsymbol{X})=\left(\left.\frac{\partial F_i}{\partial \boldsymbol{X}}\right|_{L_i+v_i^o,X^o}\right)_{n\times u} \tag{5.7}$$

$$\boldsymbol{B}(\boldsymbol{V},\boldsymbol{X})=\left(\left.\frac{\partial F_i}{\partial \boldsymbol{V}}\right|_{L_i+v_i^o,X^o}\right)_{n\times 3n} \tag{5.8}$$

Then, the observation equation in the Gauss-Helmert model is linearized and written as (Lenzmann and Lenzmann, 2004)

$$F(\boldsymbol{V},\boldsymbol{X})=\boldsymbol{A}\hat{\boldsymbol{x}}+\boldsymbol{B}\boldsymbol{V}+\boldsymbol{w}_m=\boldsymbol{0}\,, \tag{5.9}$$

where $\hat{\boldsymbol{x}}=\hat{\boldsymbol{X}}-\boldsymbol{X}^o$, $\boldsymbol{w}_m=-\boldsymbol{B}\boldsymbol{V}^o+F(\boldsymbol{V}^o,\boldsymbol{X}^o)$. At each iteration, the solution of the increment of the parameters $\hat{\boldsymbol{x}}$ and the improvements $\boldsymbol{V}$ can be derived from Equation (5.10) and Equation (5.11), respectively, where $\boldsymbol{Q}_{LL}$ is the cofactor matrix of the observations and $\boldsymbol{k}_m$ represents the so-called *Lagrange* multiplier. The cofactor matrix can be acquired from the VCM of observations $\sum_{LL}$ (i.e., $\boldsymbol{Q}_{LL}=\frac{1}{\sigma_0^2}\sum_{LL}$, σ_0^2 is the reference variance factor). The current estimated parameters $\hat{\boldsymbol{X}}$ are obtained from Equation (5.12) and used to update the approximate parameters $\boldsymbol{X^o}$, while $\boldsymbol{V^o}$ is updated by Equation (5.11) directly after each iteration until $\|\Delta\hat{\boldsymbol{x}}\|$ is less than a defined accuracy limit (e.g., 10^{-6} m).

$$\begin{bmatrix} \underset{n\times n}{\boldsymbol{B}\boldsymbol{Q}_{LL}\boldsymbol{B}^T} & \underset{n\times u}{\boldsymbol{A}} \\ \underset{u\times n}{\boldsymbol{A}^T} & \underset{u\times u}{\boldsymbol{0}} \end{bmatrix}\begin{bmatrix} \underset{n\times 1}{\boldsymbol{k}_m} \\ \underset{u\times 1}{\hat{\boldsymbol{x}}} \end{bmatrix}=\begin{bmatrix} \underset{n\times 1}{-\boldsymbol{w}_m} \\ \underset{u\times 1}{\boldsymbol{0}} \end{bmatrix} \tag{5.10}$$

$$\boldsymbol{V}=-\boldsymbol{Q}_{LL}\boldsymbol{B}^T(\boldsymbol{B}\boldsymbol{Q}_{LL}\boldsymbol{B}^T)^{-1}(\boldsymbol{A}\hat{\boldsymbol{x}}+\boldsymbol{w}_m) \tag{5.11}$$

$$\hat{\boldsymbol{X}}=\boldsymbol{X}^o+\hat{\boldsymbol{x}} \tag{5.12}$$

The VCM of estimated parameters $\sum_{\hat{X}\hat{X}}$ can be calculated by

$$\sum\nolimits_{\hat{X}\hat{X}} = \hat{\sigma}_0^2 \boldsymbol{Q}_{\hat{X}\hat{X}}, \tag{5.13}$$

where the posteriori variance factor $\hat{\sigma}_0$ and the cofactor matrix of the estimated parameters $\boldsymbol{Q}_{\hat{X}\hat{X}}$ can be calculated by Equation (5.14) and Equation (5.15), respectively. $\boldsymbol{P}$ is the weight matrix of observations which can be calculated by Equation (5.16).

$$\hat{\sigma}_0 = \sqrt{\frac{\boldsymbol{V}^T\boldsymbol{P}\boldsymbol{V}}{f}} \text{ (with the degree of freedom } f = n - u\text{)} \tag{5.14}$$

$$\boldsymbol{Q}_{\hat{X}\hat{X}} = (\boldsymbol{A}^T(\boldsymbol{B}\boldsymbol{Q}_{LL}\boldsymbol{B}^T)^{-1}\boldsymbol{A})^{-1} \tag{5.15}$$

$$\boldsymbol{P} = \boldsymbol{Q}_{LL}^{-1} \text{ (with } \boldsymbol{Q}_{LL} = \tfrac{1}{\sigma_0^2}\Sigma_{LL}\text{)} \tag{5.16}$$

When the number of estimated parameters u is beyond the minimum number of necessary parameters t, constraints between these parameters should be integrated into the Gauss-Helmert model. The constraint equations of parameters can be linearly written as

$$\underset{u\times 1}{R(\boldsymbol{X})} = \boldsymbol{C}\hat{\boldsymbol{x}} + \boldsymbol{w}_c = \underset{s\times 1}{\boldsymbol{0}} \quad \text{(with } \boldsymbol{C}(\boldsymbol{X}) = \left(\left.\frac{\partial R_i}{\partial \boldsymbol{X}}\right|_{X^o}\right)_{s\times u}\text{)}, \tag{5.17}$$

where $\boldsymbol{w}_c = R(\boldsymbol{X}^o)$, and s is the number of constraint equations of parameters (i.e., $s = u - t$).

In this case, the solution of the increment of parameters $\hat{\boldsymbol{x}}$ should be derived from Equation (5.18) which can be reduced into Equation (5.19), while the calculation of $\hat{\boldsymbol{X}}$ and $\boldsymbol{V}^o$ keeps in line with the standard Gauss-Helmert model given above. Both $\boldsymbol{k}_m$ and $\boldsymbol{k}_c$ represent the *Lagrange* multipliers in the *Lagrange* function constructed for obtaining optimal solutions.

$$\begin{bmatrix} \underset{n\times n}{\boldsymbol{B}\boldsymbol{Q}_{LL}\boldsymbol{B}^T} & \underset{n\times u}{\boldsymbol{A}} & \underset{n\times s}{\boldsymbol{0}} \\ \underset{u\times n}{\boldsymbol{A}^T} & \underset{u\times u}{\boldsymbol{0}} & \underset{u\times s}{\boldsymbol{C}^T} \\ \underset{s\times n}{\boldsymbol{0}} & \underset{s\times u}{\boldsymbol{C}} & \underset{s\times s}{\boldsymbol{0}} \end{bmatrix} \begin{bmatrix} \underset{n\times 1}{\boldsymbol{k}_m} \\ \underset{u\times 1}{\hat{\boldsymbol{x}}} \\ \underset{s\times 1}{\boldsymbol{k}_c} \end{bmatrix} = \begin{bmatrix} \underset{n\times 1}{-\boldsymbol{w}_m} \\ \underset{u\times 1}{\boldsymbol{0}} \\ \underset{s\times 1}{-\boldsymbol{w}_c} \end{bmatrix} \tag{5.18}$$

$$\begin{bmatrix} \underset{u\times u}{-\boldsymbol{A}^T(\boldsymbol{B}\boldsymbol{Q}_{LL}\boldsymbol{B}^T)^{-1}\boldsymbol{A}} & \underset{u\times s}{\boldsymbol{C}^T} \\ \underset{s\times u}{\boldsymbol{C}} & \underset{s\times s}{0} \end{bmatrix} \begin{bmatrix} \underset{u\times 1}{\hat{\boldsymbol{x}}} \\ \underset{s\times 1}{\boldsymbol{k}_c} \end{bmatrix} = \begin{bmatrix} \underset{u\times 1}{\boldsymbol{A}^T(\boldsymbol{B}\boldsymbol{Q}_{LL}\boldsymbol{B}^T)^{-1}\boldsymbol{w}_m} \\ \underset{s\times 1}{-\boldsymbol{w}_c} \end{bmatrix} \tag{5.19}$$

The VCM by the Gauss-Helmert model with constraints is still computed by Equation (5.13), while using new calculations for $\hat{\sigma}_0$ and $\boldsymbol{Q}_{\hat{X}\hat{X}}$ denoted by Equation (5.20) and Equation (5.21), respectively.

$$\hat{\sigma}_0 = \sqrt{\frac{\boldsymbol{V}^T\boldsymbol{P}\boldsymbol{V}}{f}} \text{ (with } f = n - u + s\text{)} \tag{5.20}$$

$$\boldsymbol{Q}_{\hat{X}\hat{X}} = \boldsymbol{K} - \boldsymbol{K}\boldsymbol{C}^T(\boldsymbol{C}\boldsymbol{K}\boldsymbol{C}^T)^{-1}\boldsymbol{C}\boldsymbol{K} \quad \text{(with } \boldsymbol{K} = (\boldsymbol{A}^T(\boldsymbol{B}\boldsymbol{Q}_{LL}\boldsymbol{B}^T)^{-1}\boldsymbol{A})^{-1}\text{)} \tag{5.21}$$

In the following part, the detailed process of estimating the critical parameters of geometric primitives by the Gauss-Helmert model is given, taking the sphere fitting (the center and radius are estimated) as an example.

Sphere Fitting and Parameter Estimation

Sphere fitting is the process of estimating the center coordinates and radius of each sphere based on a certain number of spherical points. A sphere in the Cartesian coordinate system can be described by the equation in Table 3.1. Considering the measurement noise, the non-linear observation equation can be written as

$$F_i(\boldsymbol{L}+\boldsymbol{V},\boldsymbol{X}) = (x_i + v_{x_i} - x_c)^2 + (y_i + v_{y_i} - y_c)^2 + (z_i + v_{z_i} - z_c)^2 - r^2 = 0\,, \tag{5.22}$$

where (x_i, y_i, z_i) with $i = 1, 2, \ldots, n$ are the coordinates of measured points on the sphere, (x_c, y_c, z_c) and r are the center point and radius of the sphere.

Equation (5.22) can be linearized and solved by the Gauss-Helmert model in an iterative process. The initial approximate parameters $\hat{\boldsymbol{X}}^o$ could be determined by solving the linear Equation (5.23) which regards x_c, y_c, z_c, and $r^2 - x_c^2 - y_c^2 - z_c^2$ as the parameters. The parameter vector $\hat{\boldsymbol{X}}^o = [x_c^o \;\; y_c^o \;\; z_c^o \;\; r^o]^T$ in Equation (5.23) can be readily obtained by Equation (5.24) (i.e., the Gauss-Markov model), where $\boldsymbol{P}$ is the weight matrix. The initial improvements $\boldsymbol{V}^o$ can be zero vector.

$$\underbrace{\begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix}}_{\boldsymbol{V}} = \underbrace{\begin{bmatrix} 2x_1 & 2y_1 & 2z_1 & 1 \\ 2x_2 & 2y_2 & 2z_2 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ 2x_n & 2y_n & 2z_n & 1 \end{bmatrix}}_{\boldsymbol{A}} \underbrace{\begin{bmatrix} x_c^o \\ y_c^o \\ z_c^o \\ r^{o^2} - x_c^{o^2} - y_c^{o^2} - z_c^{o^2} \end{bmatrix}}_{\hat{\boldsymbol{X}}} - \underbrace{\begin{bmatrix} x_1^2 + y_1^2 + z_1^2 \\ x_2^2 + y_2^2 + z_2^2 \\ \vdots \\ x_n^2 + y_n^2 + z_n^2 \end{bmatrix}}_{\boldsymbol{L}} \tag{5.23}$$

$$\hat{\boldsymbol{X}} = (\boldsymbol{A}^T\boldsymbol{P}\boldsymbol{A})^{-1}\boldsymbol{A}^T\boldsymbol{P}\boldsymbol{L} \tag{5.24}$$

For the sphere fitting by the Gauss-Helmert model, matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ are in the form of Equation (5.25) and Equation (5.26), respectively.

$$\boldsymbol{A}(\boldsymbol{V},\boldsymbol{X})_{n\times 4} = -2\cdot\begin{bmatrix} x_1 + v_{x_1}^o - x_c^o & y_1 + v_{y_1}^o - y_c^o & z_1 + v_{z_1}^o - z_c^o & r^o \\ x_2 + v_{x_2}^o - x_c^o & y_2 + v_{y_2}^o - y_c^o & z_2 + v_{z_2}^o - z_c^o & r^o \\ \vdots & \vdots & \vdots & \vdots \\ x_n + v_{x_n}^o - x_c^o & y_n + v_{y_n}^o - z_c^o & z_n + v_{z_n}^o - z_c^o & r^o \end{bmatrix} \tag{5.25}$$

$$\boldsymbol{B}(\boldsymbol{V},\boldsymbol{X})_{n\times 3n} = 2\cdot\left[diag(x_i + v_{x_i}^o - x_c^o) \;\; diag(y_i + v_{y_i}^o - y_c^o) \;\; diag(z_i + v_{z_i}^o - z_c^o)\right] \tag{5.26}$$

If the radius of sphere is known and the size of the sphere will not change (i.e., the sphere keeps rigid) during the monitoring process, the radius r can be set to a constant and the parameter vector only contains the center coordinate of the sphere (i.e., $\hat{\boldsymbol{X}} = [x_c \;\; y_c \;\; z_c]^T$). In this case, the matrix $\boldsymbol{A}$ will be in the form of Equation (5.27), while the adjustment procedure remains the same as above.

$$\boldsymbol{A}(\boldsymbol{V},\boldsymbol{X})_{n\times 3} = -2 \cdot \begin{bmatrix} x_1 + v^o_{x_1} - x^o_c & y_1 + v^o_{y_1} - y^o_c & z_1 + v^o_{z_1} - z^o_c \\ x_2 + v^o_{x_2} - x^o_c & y_2 + v^o_{y_2} - y^o_c & z_2 + v^o_{z_2} - z^o_c \\ \vdots & \vdots & \vdots \\ x_n + v^o_{x_n} - x^o_c & y_n + v^o_{y_n} - z^o_c & z_n + v^o_{z_n} - z^o_c \end{bmatrix} \tag{5.27}$$

The standard deviations of the estimated sphere center ($\hat{\sigma}_{c_x}$, $\hat{\sigma}_{c_y}$, $\hat{\sigma}_{c_z}$) and the radius ($\hat{\sigma}_r$) are listed as ($\hat{\sigma}_r$ does not exist if r is known)

$$\hat{\sigma}_{c_x} = \sqrt{\Sigma_{\hat{X}\hat{X}(1,1)}},\; \hat{\sigma}_{c_y} = \sqrt{\Sigma_{\hat{X}\hat{X}(2,2)}},\; \hat{\sigma}_{c_z} = \sqrt{\Sigma_{\hat{X}\hat{X}(3,3)}},\; \hat{\sigma}_{r} = \sqrt{\Sigma_{\hat{X}\hat{X}(4,4)}}\;, \tag{5.28}$$

where the VCM of estimated parameters $\Sigma_{\hat{X}\hat{X}}$ can be calculated by Equation (5.15) and Equation (5.13) with u = 4 (or u = 3 if r is known). Since there is no constraint between the parameters (x_c, y_c, z_c and r), s = 0.

The deformation analysis of the position change (movement) of a sphere refers to the displacement of the sphere center $\hat{\boldsymbol{X}} = (x_c,\, y_c,\, z_c)^T$ in two epochs (*epoch_1* and *epoch_2*), which can be calculated by

$$\boldsymbol{D}^{sp}_i = \begin{bmatrix} D_{X,i} \\ D_{Y,i} \\ D_{Z,i} \end{bmatrix} = \hat{\boldsymbol{X}}_{epoch\_2} - \hat{\boldsymbol{X}}_{epoch\_1} = \begin{bmatrix} x_{c,2} - x_{c,1} \\ y_{c,2} - y_{c,1} \\ z_{c,2} - z_{c,1} \end{bmatrix}. \tag{5.29}$$

5.2.2 Significance Test of Parametric Deformations

A statistical significance test should be performed with the estimated deformation of parameters and the associated standard deviations (or the VCM). Assuming the null hypothesis H_0 is that no significant deformation occurs (i.e., the alternative hypothesis H_A: deformation occurs) between two epochs, the test decision can be made by

$$\begin{array}{l} S_{epoch\_i} \le S_{f,1-\alpha} \;\; or \;\; \left|S_{epoch\_i}\right| \le S_{f,1-\alpha/2} : \;\; H_0 \; is\; accepted \\ S_{epoch\_i} > S_{f,1-\alpha} \;\; or \;\; \left|S_{epoch\_i}\right| > S_{f,1-\alpha/2} : \;\; H_0 \; is\; rejected \end{array}\;, \tag{5.30}$$

where S_{epoch_i} is the test quantity determined according to the assumed distribution, $S_{f,1-\alpha}$ (for one-tailed problems) and $S_{f,1-\alpha/2}$ (for two-tailed problems) are respective quantiles of corresponding distributions, α is the significance level and f is the degrees of freedom. For example, a *t*-test can be performed for an individual deformation value D_i (calculated by estimated parameters) as

$$S_{epoch\_i} = T_{epoch\_i} = \frac{D_i}{\sigma_{D_i}}, \tag{5.31}$$

where σ_{D_i} is the standard deviation of the parametric deformation value. While a *F*-test can be performed for a deformation vector $\boldsymbol{D}_i = (D_{i,1}, D_{i,2}, \ldots, D_{i,p})$ (calculated by the estimated parameters) as

$$S_{epoch\_i} = F_{epoch\_i} = \frac{\boldsymbol{D}_i^T \Sigma_{\boldsymbol{D}_i}^{-1} \boldsymbol{D}_i}{p}, \tag{5.32}$$

where $\Sigma_{\boldsymbol{D}_i}$ is the VCM of the deformation vector that can be calculated based on the VCM of estimated parameters using variance-covariance propagation, and p is the dimension of parameter values.

Hence, significant parametric deformations are detected at a significance level α and with the degrees of freedom f when the test statistic S_{epoch_i} exceeds the corresponding quantile ($S_{f,1-\alpha}$ or $S_{f,1-\alpha/2}$) from the relevant distribution (e.g., *t*-distribution or *F*-distribution). $S_{f,1-\alpha}$ is adopted when the deformation direction is known, while $S_{f,1-\alpha/2}$ is adopted when the deformation direction is unknown, which is the more common case.

In the following part, the *F*-test procedure is demonstrated on the parametric deformations of the sphere position to detect significant movements.

Significance Test of the Sphere Position Change

The test quantity for localizing the sphere movements is defined as

$$F_{epoch\_i} = \frac{\boldsymbol{D}_i^{sp^T} (\hat{\sigma}_0^2 \cdot \boldsymbol{Q}_{D,i})^{-1} \boldsymbol{D}_i^{sp}}{p}, \tag{5.33}$$

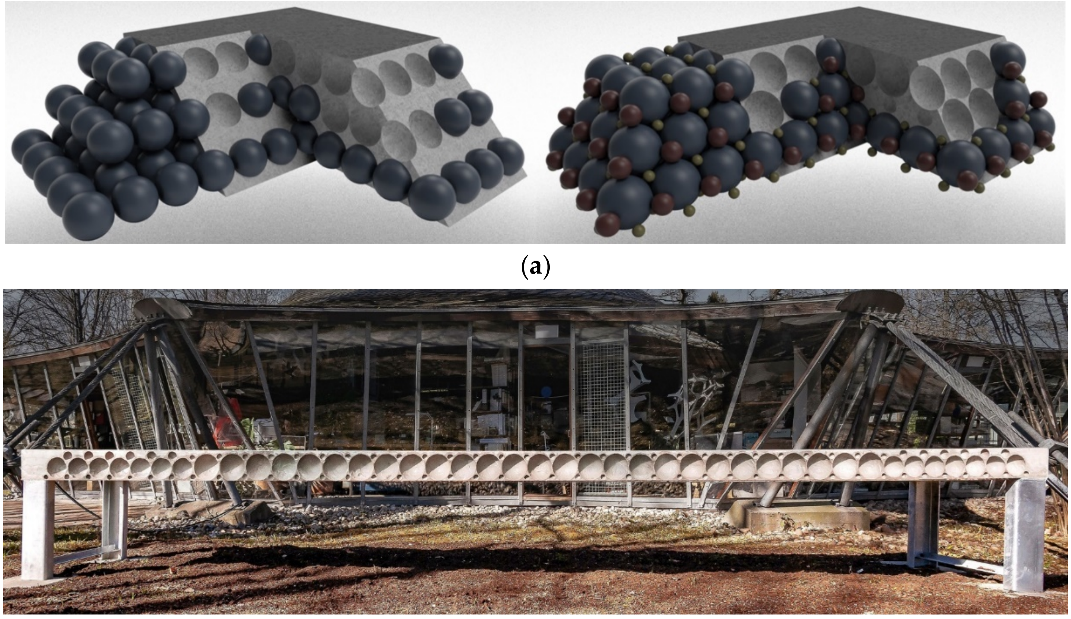
where the cofactor matrix of deformations $\boldsymbol{Q}_{D,i} = \boldsymbol{Q}_{\hat{X}\hat{X}_1} + \boldsymbol{Q}_{\hat{X}\hat{X}_2}$, $\hat{\sigma}_0^2 = \frac{\boldsymbol{V}_1^T \boldsymbol{Q}_{LL_1}^{-1} \boldsymbol{V}_1 + \boldsymbol{V}_2^T \boldsymbol{Q}_{LL_2}^{-1} \boldsymbol{V}_2}{f_1 + f_2}$ ($\boldsymbol{V}_1$, $\boldsymbol{V}_2$ and f_1, f_2 are the improvements and the degrees of freedom in *epoch-1* and *epoch-2* respectively) and p = 3. The reference test quantity $F_{p,f,1-\alpha}$ (one-tailed) or $F_{p,f,1-\alpha/2}$ (two-tailed) can be taken from the table of *F*-distribution. To test the significance of sphere movements in 3D, the two-tailed cases are often used.

5.2.3 Application of Monitoring the Production Process of Graded Concrete Components

In this section, the geometric primitive-based deformation analysis workflow is demonstrated on monitoring the positions of multiple hollow spheres embedded in the graded concrete component during its production process. Detailed processing steps and relevant discussions of the method will be presented based on this application case. The content of this section is mainly based on the author's publication: Yang Y, Balangé L, Gericke O, Schmeer D, Zhang L, Sobek W, Schwieger V. **Monitoring of the production process of graded concrete component using terrestrial laser scanning** in the *Remote Sensing*, 2021 (Yang *et al.*, 2021).

5.2.3.1 Background and Motivation

To increase resource efficiency and reduce the emissions in the concrete industry, new technologies are needed that aim at minimizing component mass and embodied energy as well as enabling full recyclability (Sobek, 2015). One approach to reducing the weight of load-bearing concrete components is the use of graded concrete. Compared to conventional concrete components, the weight of the graded component and the associated use of resources can be reduced while its performance is fully maintained (Wörner *et al.*, 2016; Kovaleva *et al.*, 2019). The gradation inside the component can be achieved by the so-called meso gradation. Hereby hollow concrete spheres are placed as close as possible to the interior of the component, as shown in Figure 5.24(a). The positions and sizes of the hollow bodies depend on the stress situation in the component (Figure 5.24(b)).



(**a**)

(**b**)

Figure 5.24: The technology of meso-gradation: (a) Sphere packing approaches for the distribution of hollow spheres inside the component (Schmeer and Sobek, 2019); (b) A graded concrete slab (Blandini and Sobek, 2020).

The precise match of the intended and the actual position of each hollow sphere is of vital importance to the performance of the meso-graded component. A major challenge for the production is to ensure the precise position of the mineral hollow spheres and to maintain it throughout the production process. A layer-by-layer casting process using self-compacting concrete was developed, which prevents the hollow spheres from floating when the concrete is cast (Sobek, 2016; Schmeer and Sobek, 2019). This process begins after the hollow spheres and reinforcement are correctly placed and positioned in the formwork of the component.

Appropriate monitoring measures are required to meet the quality requirements of the production of weight-minimized graded concrete components. These must be able to precisely determine the positions and radii of the hollow spheres in the component with the accuracy at millimeter level. The level of the fresh concrete surface is supposed to be monitored during the layer-by-layer casting process. A monitoring system needs to report on the positional stability of the hollow spheres throughout the production and needs to reconstruct the increasing concrete level in the formwork. All measurements and data processing should be completed in a short time before the concrete hardens.

5.2.3.2 Monitoring Concept, Algorithm Choice, and Overview of the Processing Workflow

For monitoring hollow spheres and the concrete level during the component production, a monitoring concept is designed for point cloud data acquisition as shown in Figure 5.25. A terrestrial laser scanner is set up on the stable ground and close to the component area, and it is fixed on a single station throughout the monitoring process. Four or more artificial targets are fixed around the formwork to evaluate the scanner's stability between epochs and register all point clouds to the first epoch. The scanning range of horizontal angle θ_{Hor} must cover the whole component formwork and all fixed targets. Additionally, the scanner should be mounted as high as possible to ensure that all hollow spheres and the surface of fresh concrete in the formwork can be scanned without a large area of occlusion.

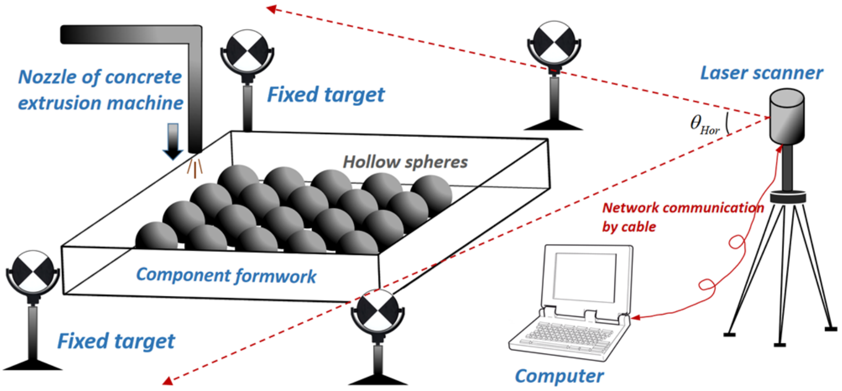


Figure 5.25: Monitoring concept and configuration for the production process (Yang *et al.*, 2021).

Spherical objects are crucial primitives found in 3D spatial data. Plenty of approaches for sphere segmentation or extraction from point cloud have been proposed, such as the clustering-based method (Attene and Patanè, 2010; Li *et al.*, 2013), sampling-based method (Wang *et al.*, 2014; Schnabel *et al.*, 2007), and Hough transform-based method (Abuzaina *et al.*, 2013; Camurri *et al.*, 2014), etc. However, limitations still exist in these methods, including the robustness to outliers or noise, computation time, requirements for the scanned spherical area and prior information, etc. In order to avoid the high complexity and operational difficulty as well as assure the robustness in such a production monitoring scenario, the coarse-to-fine segmentation framework elaborated in Section 3.2.2 is applied. In this framework, 3D region growing could extract these spherical areas (may contain outliers) without knowing the prior information of sphere positions, while RANSAC enables the model segmentation to be more robust against high-ratio noise or outliers (e.g., hollow spheres covered by unintentional fresh concrete). Therefore, the combination of these two algorithms is desirable to extract multiple concrete hollow spheres from the acquired point clouds.

Figure 5.26 shows the flowchart of the following data processing, including determining the positions of hollow spheres and the height of the concrete level. According to the scanning sequence, the whole process is composed of two main parts.

The first part is to estimate the initial parameters (i.e., the center position and the radius) of hollow spheres based on the first scanning epoch (before the casting process), as shown in the left rectangle. Taking the preprocessed point cloud data as the input, a preliminary segmentation and clustering of spherical objects, including region growing and the extraction of spherical clusters, are performed to extract the point clouds lying on hollow spheres. Then, fine segmentation of spheres using RANSAC is conducted. Finally, the initial center position and the radius of each sphere are estimated by least squares under a rigorous Gauss-Helmert model (see Section 5.2.1).

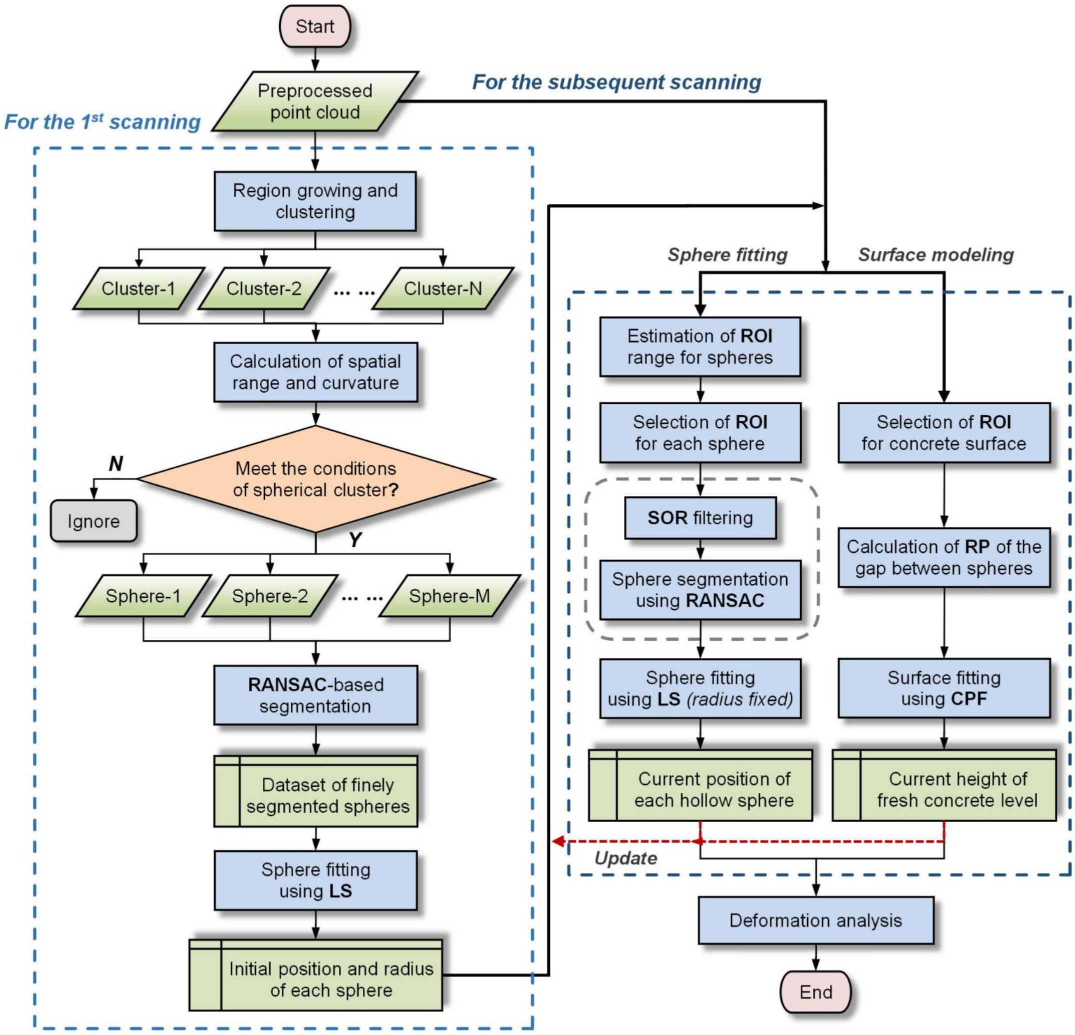

Start
Preprocessed point cloud
For the 1st scanning
For the subsequent scanning
Region growing and clustering
Cluster-1
Cluster-2
… …
Cluster-N
Calculation of spatial range and curvature
Meet the conditions of spherical cluster?
N
Ignore
Y
Sphere-1
Sphere-2
… …
Sphere-M
RANSAC-based segmentation
Dataset of finely segmented spheres
Sphere fitting using LS
Initial position and radius of each sphere
Sphere fitting
Surface modeling
Estimation of ROI range for spheres
Selection of ROI for each sphere
SOR filtering
Sphere segmentation using RANSAC
Sphere fitting using LS (radius fixed)
Current position of each hollow sphere
Selection of ROI for concrete surface
Calculation of RP of the gap between spheres
Surface fitting using CPF
Current height of fresh concrete level
Update
Deformation analysis
End

Figure 5.26: Workflow of data processing for monitoring the production process of graded concrete components (Yang *et al.*, 2021). Abbreviations in the figure: *LS: least squares, RP: representative point, CPF: cubic polynomial fitting.*

The second part involves the estimation of the hollow spheres and the concrete level in subsequent epochs, as depicted in the right rectangle. At first, the range of the region of interest (ROI) for spheres is determined based on the current height of the concrete level. Then, the ROI for each sphere and the concrete level are selected with an adaptive range (see Section 5.2.3.4 for details). A combination of the SOR filter (to reduce the ratio of outliers) and RANSAC-based segmentation is employed to obtain the point set of the exposed spherical surface within each ROI. The point at the median height within each gap between adjacent spheres is selected as the representative point for fitting the current fresh concrete surface. Finally, least squares and cubic polynomial fitting methods are utilized to fit the hollow spheres and concrete level, respectively, based on their associated point sets. After the first scanning, the radius of each sphere is fixed as a constant by using the estimation from the first epoch (i.e., subsequent epochs estimate only the sphere centers).

The advantage of the first part is that the prior information is not needed with respect to approximate positions of spheres. Due to the significant time consumption of region growing and spherical cluster extraction, however, the preliminary segmentation should only be used before concrete casting. For the subsequent scanning, with the initial positions and the estimated radii, the selection of ROI can be performed to get the local area of each sphere as the input dataset for RANSAC-based segmentation. The range of ROI for each sphere depends on the height of the current concrete level, enabling the ROI to cover the whole exposed spherical part in any case under a high inlier proportion. It is also optional to update the positions of spheres based on the current-epoch results to redefine the ROI in the next epoch. Details of each step in the workflow are described in the following subsections. It should be noted that the surface modeling for concrete level is not the focus in this application of geometric primitive-based deformation analysis, thus relevant details about this part can be referred to Yang *et al.* (2021).

5.2.3.3 *Initial State Estimation of Hollow Spheres*

As stated in Section 5.2.3.2, the initial positions and radii of hollow spheres are supposed to be estimated after the first scanning (before casting) if there is no prior information on sphere parameters. This part consists of preliminary segmentation and spherical cluster extraction, and then the parameters of spheres can be estimated (see Section 5.2.1) after further segmentation by RANSAC.

Preliminary Segmentation using 3D Region Growing

The 3D region growing typically starts from one or more points (seed points) featuring specific characteristics and then gathers the nearest neighbors in the seed area on the basis of specific constraints such as similar surface orientations or curvatures (Wang *et al.*, 2018). The detailed process of the 3D region growing algorithm is described in Appendix A.1. Vo *et al.* (2015) found that the method requires a considerable amount of time for parameter tuning, especially for curved surface or sphere segmentation. The influence factors in the standard region growing algorithm include (1) the neighborhood size for normal vector and curvature estimation, (2) the threshold of angle difference (TAD) between normal vectors of each nearest neighbor point and current seed point, (3) the threshold of local curvature (TLC) for each point, (4) the size of neighborhood for searching candidates, and (5) the threshold of point number as a clustered region, etc. There are two ways to define the size of the neighborhood:

(1) The point number of *k*-nearest neighbors and (2) the radius of the nearest neighbors (RNN). Herein, RNN is adopted to calculate the normal vector, local curvature, and growing neighborhood, which enables an optimal neighborhood under different point densities.

In this contribution, PCA is applied to estimate the normal vector and curvature of each point (Rusu, 2009). In consideration of the computational complexity and the insensitivity to the noise, the ratio between the smallest eigenvalue and the sum of eigenvalues (derived from the covariance matrix of the neighboring point set for the query point) is used to approximate the surface variation, and to indicate the local curvature around the query point indirectly (Pauly *et al*., 2002).

To verify the feasibility of 3D region growing for multiple sphere segmentation and to investigate the optimal selection of critical parameters, a setup of eight plastic spheres with a radius of 74 mm is used to simulate the layout of hollow spheres. Figure 5.27 shows the placement of spheres and the point cloud of this setup. Part of the background is retained to show the process of detecting spheres from a complex scenario. Notably, this multiple-sphere dataset is only used to demonstrate the processing steps and the strategies of parameter selections. Accuracy and performance evaluations are not performed for this dataset.

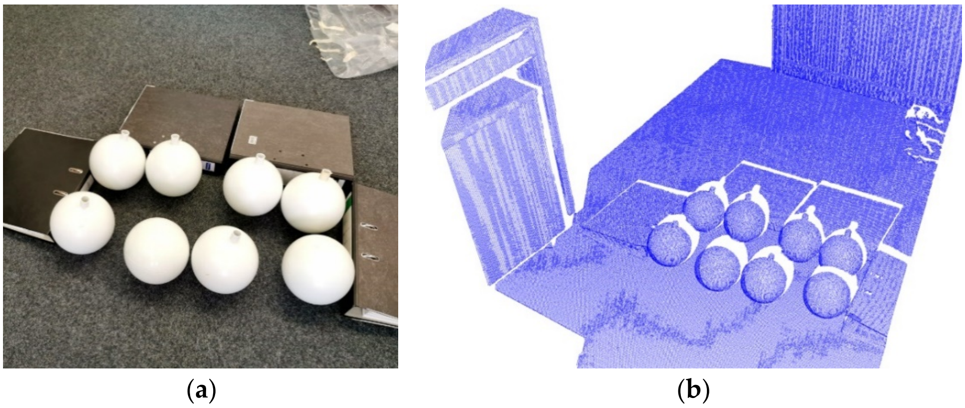


Figure 5.27: The multiple-sphere scanning test: (a) The placement of multiple spheres; (b) Preprocessed point cloud (voxel size in downsampling: 3 × 3 × 3 mm^3) (Yang *et al*., 2021).

The calculations of normal vector and curvature for each point are the basis of 3D region growing, while the size of the corresponding neighborhood will directly influence the PCA-based normal estimation. Figure 5.28 shows part of the segmentation results under different normal and curvature calculations, where RNN is set to different sizes (from 10 mm to 50 mm) while other parameters are fixed as empirical values. The segmented clusters are labelled with random colors, and the red parts are not considered as a cluster because of too few points (less than the minimum point number as a region). The bad or wrong segmentations are marked by black dotted rectangles according to the visual comparison with the ground truth.

Figure 5.28 shows that smaller neighborhood selection caused over-segmentation (see Figure 5.28(a)) while a large RNN setting led to under-segmentation (see Figure 5.28(c)). For this multiple-sphere test, the optimal RNN is empirically determined from 20 mm to 30 mm according to the segmentation effects within the investigated range of RNN.

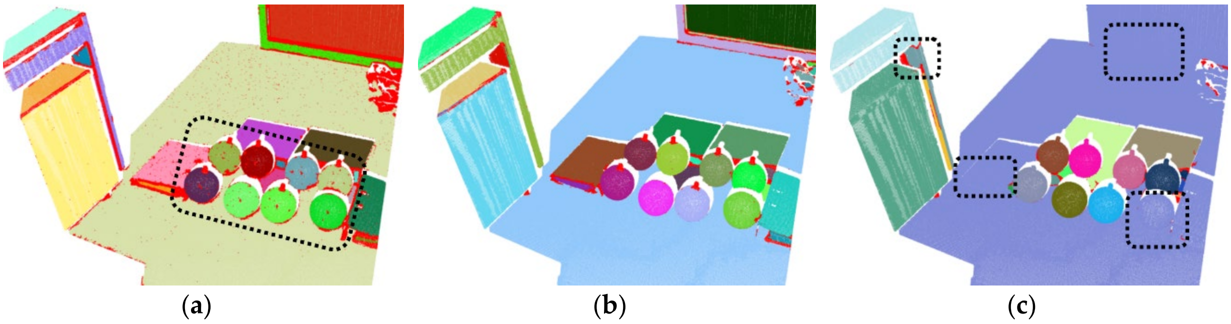


Figure 5.28: The segmentation results of multiple spheres under different RNN (TAD = 3.00°, TLC = 0.10): (a) RNN = 10 mm; (b) RNN = 20 mm; (c) RNN = 50 mm (Yang *et al.*, 2021).

In addition to the neighborhood size in normal and curvature calculations, TAD and TLC also influence the quality of segments by 3D region growing. Figure 5.29 and Figure 5.30 show parts of segmentation results under different settings of TAD (from 2.00° to 4.00°) and TLC (from 0.01 to 20.0).

Figure 5.29 indicates that the segmentation result is sensitive to the change of TAD. Similar to the results under different RNN, a small TAD setting caused over-segmentation on the spheres (see Figure 5.29(a)), while a large TAD setting led to under-segmentation to a certain extent (see Figure 5.29(c)). If only concerning whether the spherical points can be segmented independently and completely, the range from 2.75° to 3.75° of TAD is suggested for the sphere with the given radius.

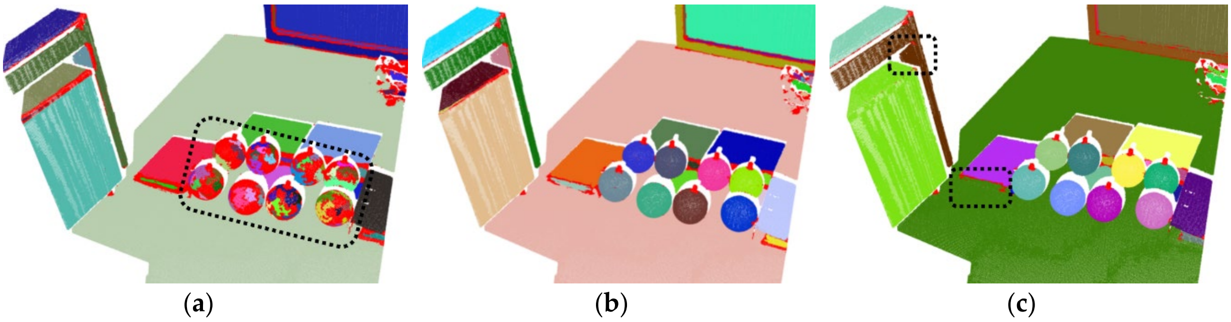


Figure 5.29: The segmentation results of multiple spheres with different TAD (RNN = 20 mm, TLC = 0.10): (a) TAD = 2.00°; (b) TAD = 3.00°; (c) TAD = 4.00° (Yang *et al.*, 2021).

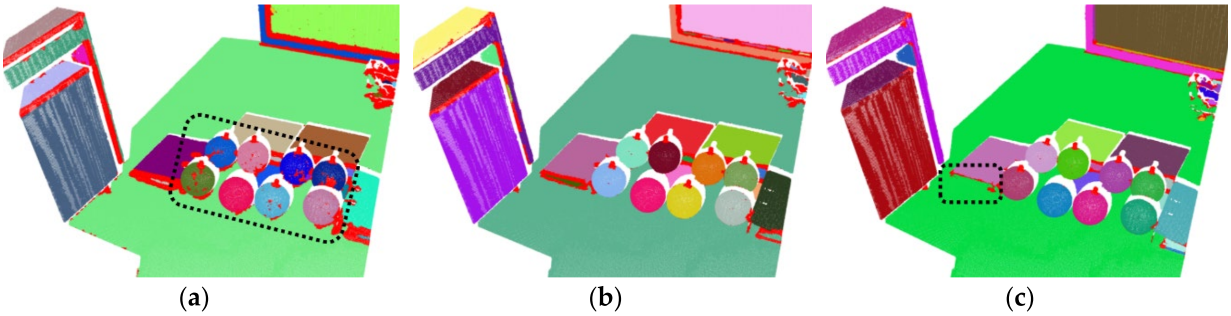


Figure 5.30: The segmentation results of multiple spheres with different TLC (RNN = 20 mm, TAD = 3.00°): (a) TLC = 0.01; (b) TLC = 0.05; (c) TLC = 20.0 (Yang *et al.*, 2021).

Compared to TAD, TLC has not such a significant influence on the segmentation effects (see Figure 5.30). For a TLC between 0.05 and 10, all obvious geometric primitives could be segmented and clustered almost perfectly. There is a slight under-segmentation when TLC reached 20 (see Figure 5.30(c)). This is because TLC only determines the generation of seed points rather than region points, while the clustering of region points depends on TAD merely. However, too small TLC may hinder the generation of seed points and stop the process of region growing in large-curvature areas, like edges and spheres. These regions would be independently clustered from the point cloud and are labeled as red due to insufficient points to be treated as a cluster (see Figure 5.30(a)), which caused discontinuity on the spherical surfaces. Therefore, TLC from 0.05 to 10 is suggested for this scanning test. In addition, for the point number of the searching neighborhood in the growing process, an empirical value from 10 to 30 is chosen, taking into account the point density, since the segmentation result is also not sensitive to this parameter.

Screening Conditions for Spherical Clusters

Spherical clusters are supposed to be completely extracted from the preliminary segmentation results. Herein, two simple screening conditions based on the spatial ranges and the medians of sorted curvatures of segmented clusters are proposed to extract the spherical clusters generated by region growing-based segmentation. The extraction steps are summarized as follows:

(1) The spatial ranges in the direction of three axes of each cluster whose number of points is greater than a threshold will be calculated;
(2) The curvature of points in each cluster within the threshold of spatial ranges (TSR) will be calculated and sorted;
(3) The cluster of which the median of sorted curvatures is within a specific threshold (TCM) will be regarded as a sphere and extracted into the dataset storing spherical points.

The determination of TSR ought to refer to the radius of the sphere and the proportion of visible part. When scanning with a single station, always less than 50% of the surface of a sphere is exposed, and it may be even less under the occlusion of other spheres. Considering the radius of these spheres is 74 mm in this multiple-sphere scanning test, the range of TSR is set from 0.05 m to 0.16 m in the direction of X and Y axes and from 0.02 m to 0.16 m in the direction of Z-axis empirically.

The curvature estimations of spherical clusters may be not consistent because of the errors at some edge points. Figure 5.31 shows the curvature distributions of two extracted spherical clusters in the scanning test (results of other six spheres are given in Figure A.2). It can be seen that the majority of point curvatures are within a specific range that is strongly related to the sphere radius. The mean and median of point curvatures are also displayed. The average curvatures of these spherical clusters are between 0.014 to 0.017, while the median is slightly higher than the mean of curvatures. To weaken the influence of outlier curvatures at some edge points and make a more stable threshold setting, the median of sorted curvatures of a cluster is adopted as the second screening condition for the extraction of spherical clusters. According to Figure 5.31, the range of TCM is empirically set from 0.014 to 0.018 in this scanning test.

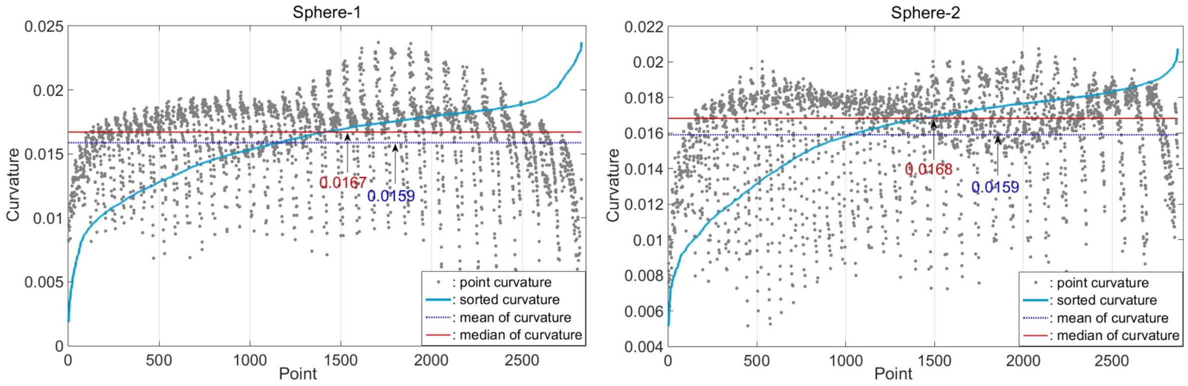


Figure 5.31: The curvature distributions of two spherical clusters in the scanning test (Yang *et al.*, 2021).

The extracted spherical clusters by the two screening conditions from the region growing segments are shown in Figure 5.32. These eight spheres are extracted from the segmented point cloud individually and completely. However, these spherical clusters may still contain outliers or noise, which can be further eliminated by RANSAC-based segmentation (see Section 5.2.3.4). Then the center coordinates and radii of spheres are estimated by least squares under the Gauss-Helmert model (see Section 5.2.1). In addition, for an unorganized point cloud, the extracted clusters will be sorted by their number of points after the region growing-based segmentation, resulting in the actual sequences being disordered in the horizontal direction. It is thus crucial to reorder these spherical clusters according to their center positions and number these spheres in accordance with the X-axis or Y-axis direction. Correspondences of identical spheres between epochs can be established by the reordered number of each sphere.

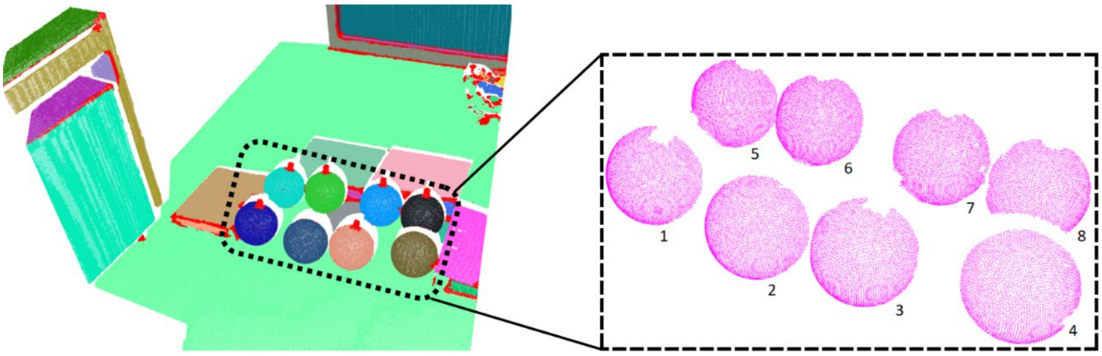


Figure 5.32: The extracted spherical clusters from region growing segments in the scanning test (Yang *et al.*, 2021).

5.2.3.4 Subsequent State Estimation of Hollow Spheres

Owing to the dense packing of the hollow spheres, their horizontal movements would normally be in a small extent during the casting process. Therefore, it is possible to estimate the subsequent state of a hollow sphere within a limited horizontal range (i.e., ROI). In this case, the sphere can be segmented by RANSAC with fewer iterations under a higher proportion of spherical points (inliers) in a ROI. Besides, the time consumption is significantly less than that of region growing-based segmentation, enabling timely feedback during the production process. Thus, this RANSAC-based segmentation is adopted to estimate the subsequent positions of the hollow spheres after the first epoch.

Selection of Regions including Spheres

Since the standard RANSAC algorithm segments one single object, the input dataset should only contain one sphere, and the proportion of inliers ought to be as high as possible. Based on the initial positions and radii estimated from the first epoch, it is possible to select a local area as the ROI including one single sphere. Supposing that the horizontal movement of each sphere is within the range of 0.3-times radius, the region with a range of 1.3-times radius from the center of each sphere should be selected as the ROI. A horizontal square area whose sides are along the *X*-axis and *Y*-axis direction is selected as the ROI for each sphere, as demonstrated in Figure 5.33. These ROIs are merely determined by the center position and side length, and half of the side length is defined as the range of ROI.

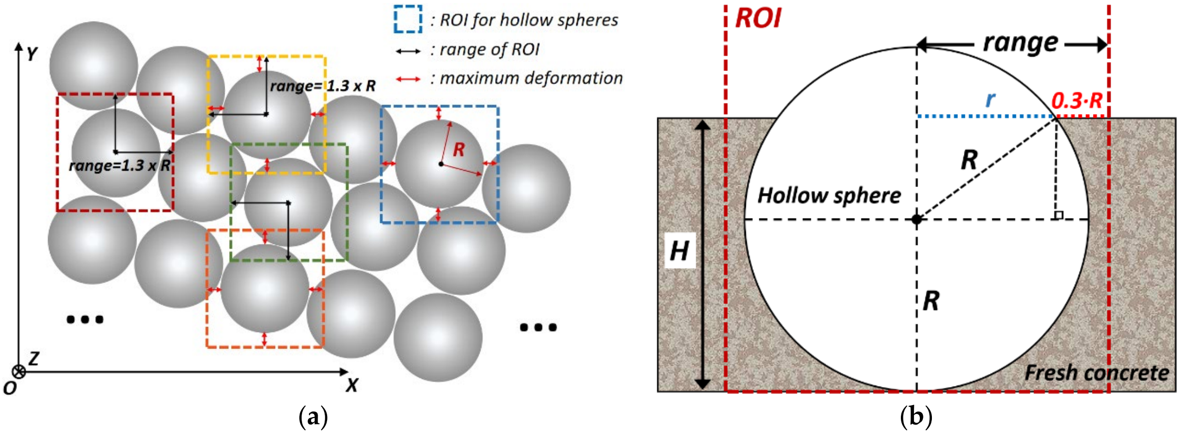


Figure 5.33: The diagram of the selection of ROI for hollow spheres: (a) Top view of the whole component; (b) Front view of one sphere (Yang *et al.*, 2021).

The center position of any ROI is entirely consistent with its corresponding spheres, and it can be acquired from the initial state estimation and updated by the current state estimations (when the horizontal movement tends to be significant). In order to ensure that ROI can completely cover the sphere and maintain a high inlier proportion as well, the range of ROI can be set by

$$R_{ROI} = \begin{cases} 1.3 \cdot R & H \le R \\ \sqrt{R^2 - (H-R)^2} + 0.3 \cdot R & H > R \end{cases}, \tag{5.34}$$

where R_{ROI} is the range of ROI after each casting, R is the radius of the sphere and H is the current height of the concrete level from the bottom of the sphere. H can be obtained from the surface modeling of concrete level, i.e., the height of the fitted surface at the position of sphere center). Equation (5.34) makes the selection of ROI more adaptive and ensures the inlier ratio accordingly with the decrease of the sphere's exposed part. Figure 5.34 shows the ROI for each sphere (R_{ROI} is 1.3-times radius) in the multiple-sphere scanning test.

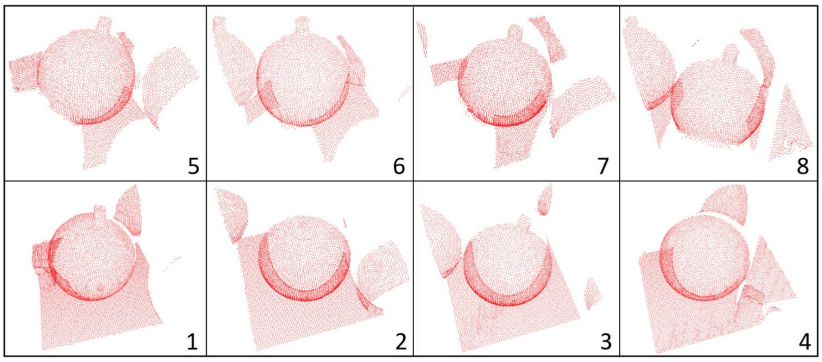


Figure 5.34: The selection of ROI for hollow spheres in the scanning test (Yang *et al.*, 2021).

Sphere Segmentation using RANSAC

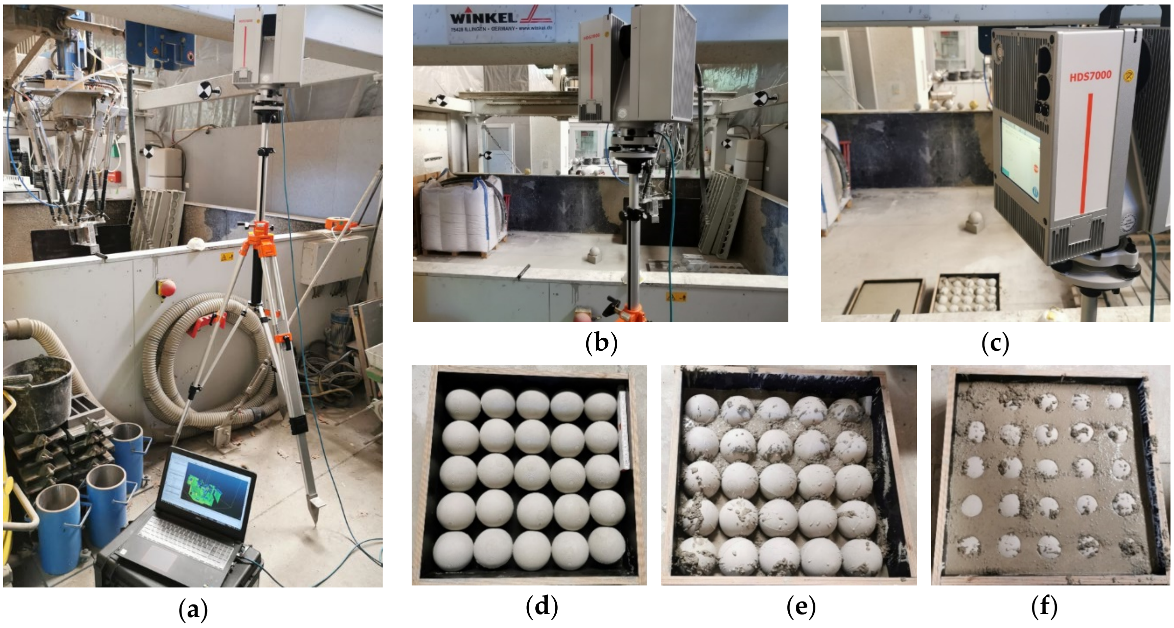
As an efficient and robust method of model (the geometric primitive) segmentation from the dataset with a high amount of noise and outliers, RANSAC can be performed in each ROI to segment the included sphere. Although the descendants of standard RANSAC (e.g., Least Median of Squares (LMEDS), Randomized RANSAC (RRANSAC), Progressive Sample Consensus (PROSAC), etc.) are able to promote robustness and efficiency to a certain extent, limitations or imperfections still exist in specific cases (Choi *et al.*, 2009). Considering the comprehensive performance on both computing time and robustness, standard RANSAC (see Appendix A.2 for details) is employed to finely segment the spheres herein. The distance threshold (DT_{RANSAC}) between inlier points and the estimated model is the crucial parameter in RANSAC-based segmentation. A large DT_{RANSAC} yields more noise and even outliers in the segments, whereas too small DT_{RANSAC} may derive insufficient number of points and even fail to obtain the points on the model (deviated segments). Both cases will eventually reduce the accuracy of estimated parameters. Taking into account the approximate uncertainties of the point cloud and the size of spheres, 2 mm is empirically set in the scanning test.

5.2.3.5 Experiment Description and Data Acquisition

An experiment is conducted to verify and evaluate the workflow for monitoring the production process of meso-graded concrete components, particularly the geometric primitive-based deformation analysis for the positions of hollow spheres. The center position and radius of each sphere are acquired as reference values by manually selecting the spherical points from the original point cloud and making an optimal estimation in a commercial software called *Geomagic Studio* (www.geomagic.com; Fu and Zhang, 2007) .

In this experiment, the production of a small-scale meso-graded concrete component is monitored in a laboratory. The monitoring configuration and casting process are shown in Figure 5.35. 25 spheres with a designed radius of 50 mm are placed in an orthogonal packing pattern within the formwork. These hollow spheres are all rigid bodies where no shape deformation occurs during the production process. A Leica HDS7000 laser scanner is set up on a tripod about 2 m away from the formwork. Four black & white planar targets are set around the working area for georeferencing. Owing to the indoor environment and short measuring distance, the influence of atmospheric variation on the measurement is negligible. The sample is produced manually by casting a defined amount of fresh liquid concrete into the gaps between the hollow spheres. As a result of the manual production, the surface of some hollow spheres is partially obstructed with concrete (see Figure 5.35(e) and Figure 5.35(f)). However, these unintentional imperfections allow to improve and evaluate the robustness of the sphere detection method against such outliers.

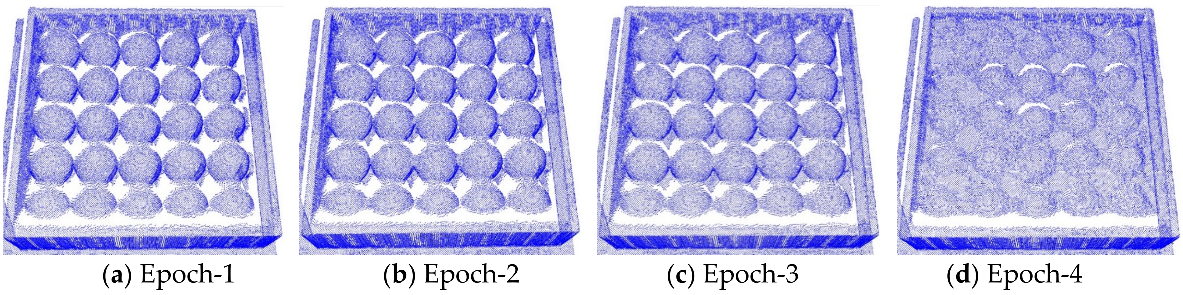
There are four epochs of scanning during the monitoring process. Epoch-1 is the first scanning to estimate the initial center position and the radius of each sphere before casting, thus the initial parameters of spheres are estimated and compared to the reference values in epoch-1, while the movements of hollow spheres and the concrete level are monitored from epoch-2 to epoch-4.



(**a**) (**b**) (**c**) (**d**) (**e**) (**f**)

Figure 5.35: The experimental setup (a-c) and the production process of an exemplary concrete component (d-f) (Yang *et al.*, 2021).

The point cloud data acquisition consists of data transmission and format conversion. To obtain unified-format point cloud data in real-time, an independent program with a user interface for the HDS7000 laser scanner is developed based on the corresponding Software Developer Kits (SDK). The scanner can be controlled to scan the objects with a predetermined angle range, resolution, and quality. In this experiment, the scanning process in each epoch takes approx. 2 minutes with a high-resolution setting and 90° horizontal scanning range (to cover the target area) in the scanner. The point cloud data is finally stored in PCD format by the format conversion module.



(**a**) Epoch-1 (**b**) Epoch-2 (**c**) Epoch-3 (**d**) Epoch-4

Figure 5.36: Preprocessed point clouds of the formwork area in the experiment (Yang *et al.*, 2021).

During the experiment, the scanner is fixed in one station. Registration by fixed targets is performed in each epoch to check the stability of the scanner. Using the voxel grid filter with 3 × 3 × 3 mm^3 grid size and the SOR filter (Rusu *et al.*, 2008) with the threshold of double standard deviation, the preprocessed point clouds of component areas are obtained as shown in Figure 5.36, followed by the detailed descriptions in Table 5.1.

Table 5.1: Descriptions of four-epoch preprocessed point clouds (Yang *et al.*, 2021).

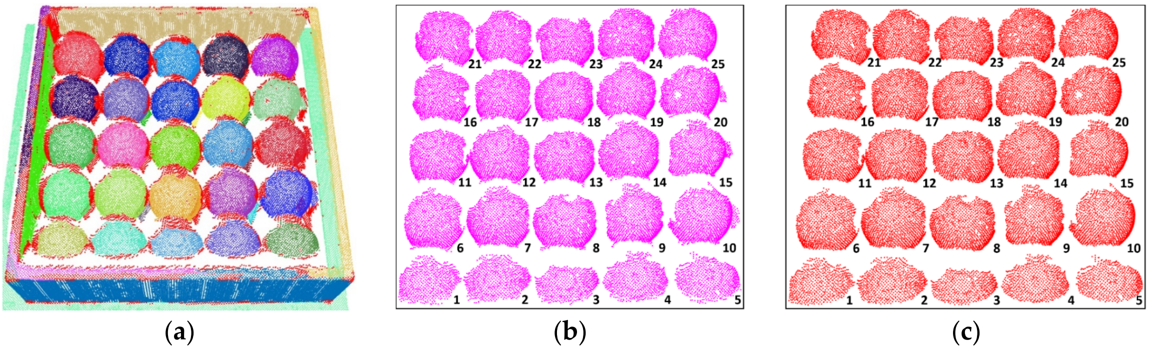
Description	Epoch-1	Epoch-2	Epoch-3	Epoch-4
Number of Points	64,711	65,081	64,825	62,305
Average Point Spacing	2.31 mm	2.29 mm	2.28 mm	2.23 mm
Size of Selected Area	600 × 600 × 200 mm^3			

5.2.3.6 Experimental Results

Figure 5.37 presents the sphere segmentation for initial parameter estimation in the first epoch based on the combination of 3D region growing and RANSAC. According to the parameter tuning described in Section 5.2.3.3, the optimal parameters of TAD, TSR and TCM are obtained and listed in Table 5.2. The errors of estimated position (E_X, E_Y, E_Z) and radius (E_R) of each sphere by comparing with the reference values from *Geomagic Studio* are given in Figure 5.38. The *Y*-axis direction aligns with the scanner's initial scanning direction, extending towards the left side of the formwork. The *Z*-axis is oriented vertically, and the *X*-axis is orthogonal to both the *Y* and *Z* axes. All errors are below 1 mm, thus the initial positions and radii of spheres estimated in epoch-1 can be used for the following epochs.

Table 5.2: The parameter settings in the experiment (Yang *et al.*, 2021).

RNN	TAD	TLC	TSR	TCM	DT$_{RANSAC}$
20 mm	4.0°	0.5	Horizontal: 0.02 ~ 0.12 m Vertical: 0.02 ~ 0.12 m	0.025 ~ 0.035	2 mm



(a) (b) (c)

Figure 5.37: Sphere segmentation for initial parameter estimation in epoch-1: (a) Segmentation result by region growing; (b) Extracted spherical clusters from (a); (c) Segmentation of (b) by RANSAC (Yang *et al.*, 2021).

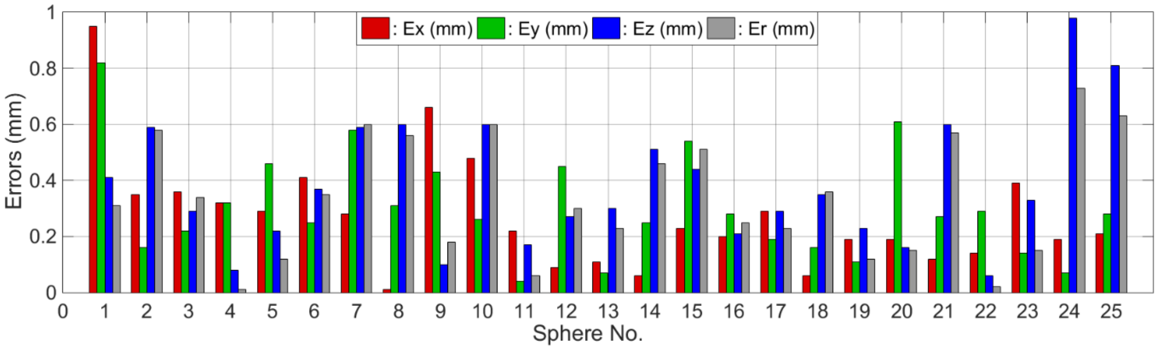
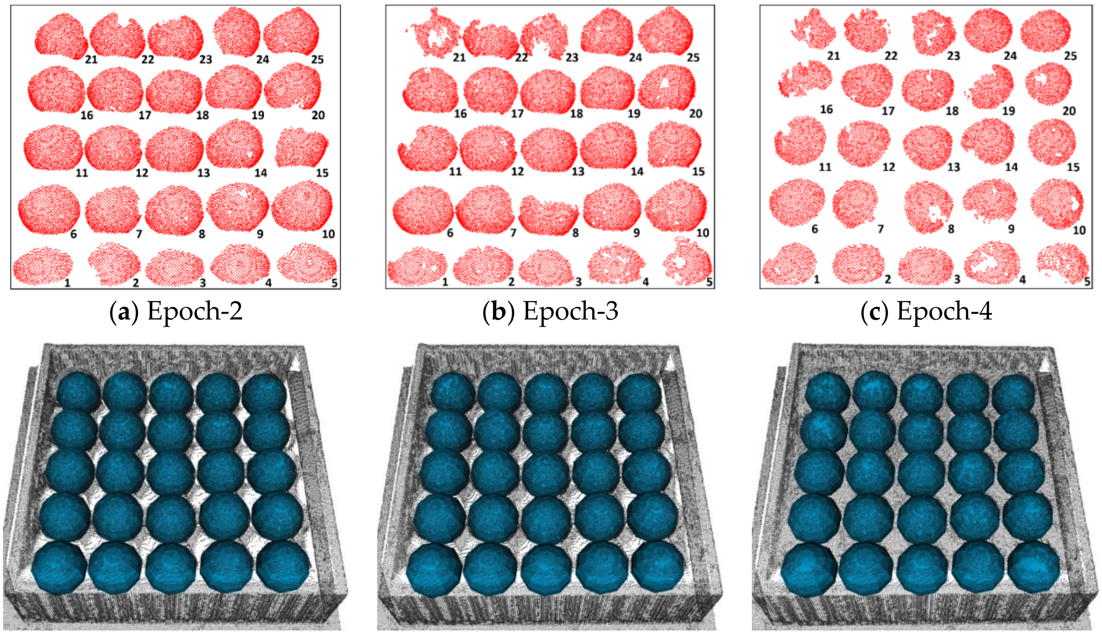


Figure 5.38: The estimation errors of initial parameters of spheres in epoch-1 (Yang *et al.*, 2021).

In this experiment, the radius of each sphere is fixed to a constant (i.e., the estimated initial value) in the parameter estimation of sphere centers after the first epoch, for these hollow spheres keep rigid during the production process. Besides, the limit of the maximum number of iterations in RANSAC is increased to get a more reliable estimation. The final segmentation and fitting results of hollow spheres are presented in Figure 5.39.

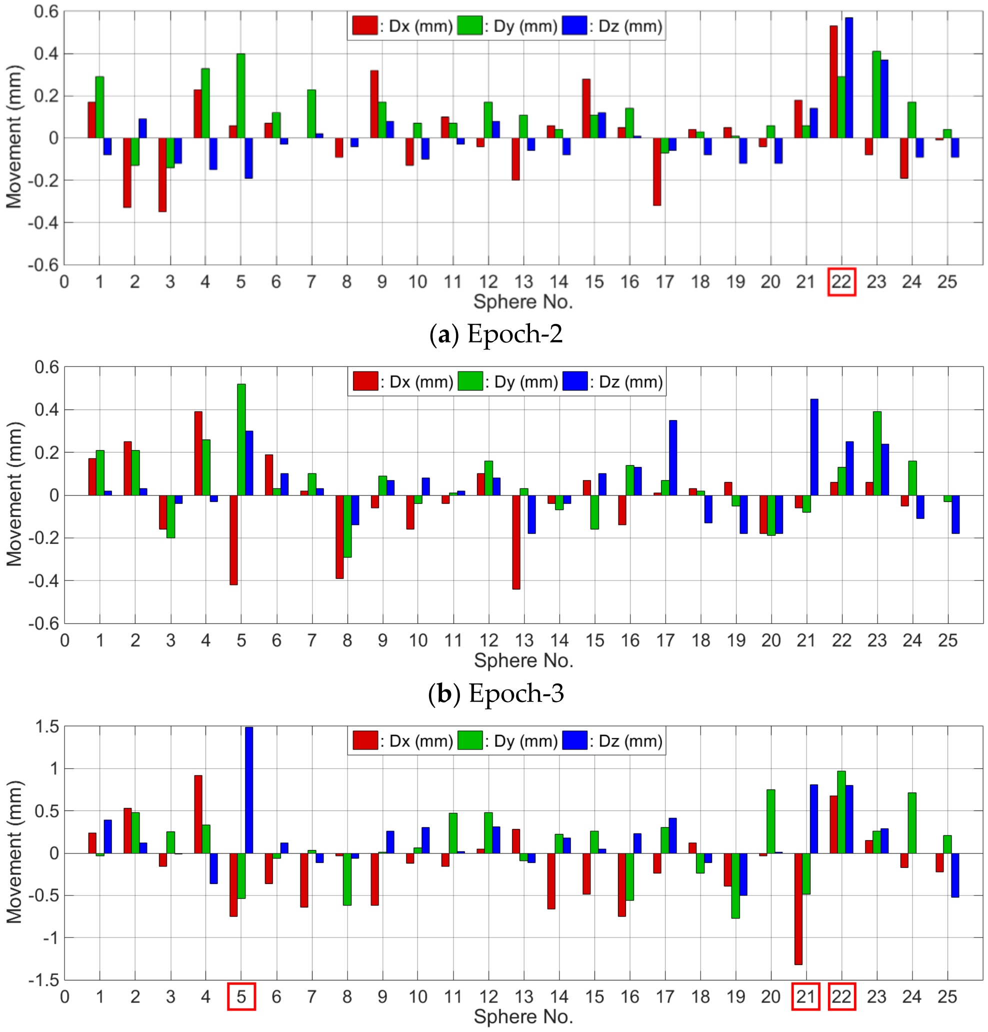


(**a**) Epoch-2 (**b**) Epoch-3 (**c**) Epoch-4

(**d**) Epoch-2 (**e**) Epoch-3 (**f**) Epoch-4

Figure 5.39: Sphere segmentation and fitting from epoch-2 to epoch-4: (a-c) The final segments of spheres; (d-f) The fitted sphere models displayed in original point clouds (Yang *et al.*, 2021).

The movements (in three directions) of each hollow sphere compared to epoch-1 are calculated by Equation (5.29) based on the estimation of sphere centers (see Section 5.2.1). To test whether these spheres are moved significantly, the F-test is conducted for each sphere by the procedure in Section 5.2.2. According to the monitoring requirements, the level of significance α is set to 0.3% in this experiment based on the 3-sigma rule. Besides, the uncertainties from the georeferencing process are also considered in the significance test by integrating the average standard deviations of fixed targets' positions between epochs into Equation (5.33). The results of parametric deformations (i.e., movements) from epoch-2 to epoch-4 are shown in Figure 5.40, where spheres with significant movements (with a 99.7% level of confidence) are marked on the X-axis with a red frame.



(**a**) Epoch-2

(**b**) Epoch-3

(**c**) Epoch-4

Figure 5.40: The movement of each sphere and its significance (epoch-2 to epoch-4) (Yang *et al.*, 2021).

The results in Figure 5.40 show that one sphere moved below 1 mm in epoch-2 and no sphere moved in epoch-3 significantly. In epoch-4, three spheres moved within 1.5 mm in the three orthogonal directions. These movements are probably caused by the manual casting process in this experiment, which could be improved by an automatic casting procedure that utilizes a concrete extrusion machine. In any case, this level of movement of these spheres is so negligible that it will not affect the functionality of the component negatively.

Two characteristics are adopted to assess the performance of the proposed workflow for multiple geometric primitive (sphere) detection: completeness and accuracy (Schweitzer and Schwieger, 2011; Balangé *et al.*, 2021). The completeness is defined by the ratio of detected spheres with respect to the total sphere number, and it reflects the quality of the segmentation process. The accuracy is quantified by the average (*avg*) and the maximum (*max*) standard deviations of the estimated position ($\hat{\sigma}_P$) of fitted spheres which can be calculated by

$$\hat{\sigma}_P = \sqrt{\hat{\sigma}_{c_x}^2 + \hat{\sigma}_{c_y}^2 + \hat{\sigma}_{c_z}^2}\,, \tag{5.35}$$

where the standard deviations of the estimated sphere center ($\hat{\sigma}_{c_x}$, $\hat{\sigma}_{c_y}$, $\hat{\sigma}_{c_z}$) can be acquired from Equation (5.28). The quality evaluation of sphere detection and fitting is presented in Table 5.3, demonstrating the successful detection of all spheres using the proposed segmentation strategy. Moreover, the accuracy of sphere position estimations in this application can achieve submillimeter-level precision.

Table 5.3: Quality evaluation of the sphere detection and fitting (Yang *et al*., 2021).

Evaluation	Epoch-1	Epoch-2	Epoch-3	Epoch-4
Completeness [%]	100	100	100	100
Accuracy (avg/max) [mm]	0.18 / 0.42	0.12 / 0.21	0.14 / 0.23	0.22 / 0.63

5.2.4 Discussion

5.2.4.1 Influence of Segmented Areas and Outliers

From the results of the experiment in Section 5.2.3, the errors of estimated sphere parameters (by comparing with reference values or calculated from the sphere fitting) typically increase with the casting process (see Table 5.3). One of the causes is the reduction of the sphere areas captured by the laser scanner due to the increasing concrete layer, which leads to low coverage of the segmented spherical points. Therefore, low captured areas for the geometric primitives may yield insufficient points in the segments for the following parameter estimation, thus reducing the accuracy of estimated parametric deformations.

In addition, small objects attached on the geometric primitives (e.g., the fresh concrete sticking on the exposed spheres after casting in Figure 5.34(f)) may be erroneously segmented as inliers with inappropriate parameter settings (e.g., the distance threshold in RANSAC). Most outliers could be eliminated by the RANSAC-based segmentation. However, when most areas of the geometric primitive are occupied by outliers, a wrong (deviated) segment from the RANSAC process may arise, which fails the following parameter estimation and deformation analysis. This kind of systematic error from the segmentation process cannot be detected by the significance test. Thereby, ensuring the ratio of inliers (e.g., > 50%) in the ROI for RANSAC-based segmentation is crucial to obtain accurate patches (representing geometric primitives) and associated parameter estimations.

5.2.4.2 Influence of Spatial Correlations

In the sphere fitting of the experiment, spatial correlations between the observation points are neglected in the stochastic model, and all spherical points are considered equally weighted. Hence, the VCM was simply defined as an identity matrix in this experiment. Different from planes, the orientations of points on a sphere are inconsistent. Thus, high spatial correlations are not assumed in the geometric primitives with curved surfaces. However, low correlations between measured points may still exist, provided a high point density on the quadratic surface.

To investigate the influence of spatial correlations on the significance test, the effective number of measurements is again introduced to recalculate the test quantities by Equation (5.33) in the experiment. Table 5.4 shows the significantly moved spheres by considering different approximate averages of spatial correlations (i.e., *K* calculated by Equation (3.7)). The results show that, with a very small increase of *K*, the test quantity decreases dramatically, resulting in less significantly moved spheres. When the average correlations increase by 0.1, all deformations (movements) are tested as non-significant. Therefore, the spatial correlations between points have a significant influence on the geometric primitive-based deformation analysis. The empirical determination of approximate spatial correlations for geometric primitives with quadratic surfaces is still challenging and is not discussed in this thesis.

Table 5.4: The spheres with significant deformations considering spatial correlations.

K	**Epoch-2**	**Epoch-3**	**Epoch-4**
0	No.22	none	No.5, No.21, No.22
0.01	No.22	none	No.22
0.1	none	none	none

5.3 Deriving Dense 3D Displacement Vectors via Local Registration

Rigid or quasi-rigid areas (patches) can be segmented automatically or manually from two deformed point clouds, and their correspondences are established based on areal feature descriptors (see Section 3.2.3) or spatial distances (i.e., find the nearest one if deformations are small). Hence, a local rigid registration can be performed between two corresponding patches to align the patch in the *first* (or *second*) epoch to the one in the *second* (or *first*) epoch. This alignment will produce a synthetic transformation matrix by a coarse-to-fine registration process, as well as the stochastic information of transformation parameters. By using this transformation matrix and measured points within the *first* (or *second*)-epoch patch, displacement vectors (from the *first* epoch to the *second* epoch) can be generated for all points or resampled positions located on the patch. Thus, a dense 3D displacement field can be generated if the corresponding patches cover the monitored area. Besides, a statistical significance test can be conducted for the calculated displacements based on the stochastic information derived from the registration process. The overall workflow of this local registration-based deformation analysis for 3D point clouds is depicted in Figure 5.41.

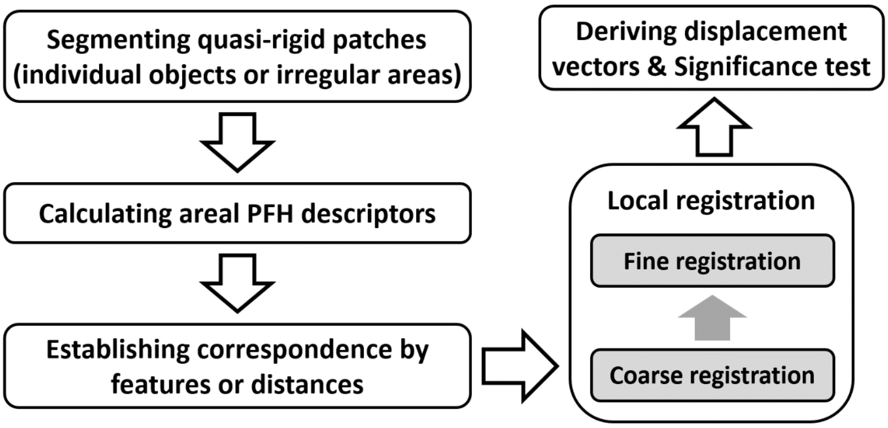

Segmenting quasi-rigid patches (individual objects or irregular areas)
Calculating areal PFH descriptors
Establishing correspondence by features or distances
Local registration
Coarse registration
Fine registration
Deriving displacement vectors & Significance test

Figure 5.41: The workflow of local registration-based deformation analysis for 3D point clouds.

This local registration-based method can theoretically derive displacement vectors between any two patches from different epochs. These patches should display sufficient surface variations (i.e., the local surface orientations have distribution in as many directions as possible), which makes the ICP-based registration more reliable. Since most geometric primitives have insufficient variations of surface orientations and symmetric geometries (e.g., planes and cylinders), the local registration may fail to align two primitives correctly if their poses have significant differences (e.g., a reverse alignment along the axis of two cylinders), leading to erroneous computation of displacement vectors. Therefore, the local registration-based method is not recommended for planar patches and geometric primitives with insufficient surface variations or symmetric structures.

5.3.1 Synthetic Transformation Matrix Derived from Local Registration

Since the segmented patches are assumed quasi-rigid between epochs, deformations (movements) of all points within the rigid areas can be described by one set of transformation parameters (or one transformation matrix). Correspondences for deformation analysis are implicitly derived through a rigid registration process that minimizes the distances of correspondences for the registration. Feature detection or parameterization is not involved in this method, which improves its computational efficiency and usability on point cloud-based deformation analysis. In addition, compared to a single or a small amount of points, the uncertainties of derived displacement vectors are reduced, benefiting from the contribution of all points in the rigid area to the registration process, which brings the potential to detect small-magnitude deformations.

For computational convenience, in this thesis, a synthetic transformation matrix is defined to incorporate all transformation parameters during the two-stage (coarse-to-fine) registrations. To generalize the local ICP-based method (Teza *et al.*, 2007; Pfeiffer *et al.*, 2018; Raffl and Holst, 2022), the synthetic transformation matrix $\boldsymbol{T}_{syn}^{1\to2}$ (register the patch in the first epoch to that in the second epoch) calculated by Equation (5.36) is derived not only from a fine registration (ICP or its variants), but also from a coarse registration (e.g., PCA-based (Wang and Bao, 2021) or 4PCS-based (Aiger *et al.*, 2008)) to avoid the local optimization (minimum) of ICP in the case of larger in-plane movement between two patches.

$$\boldsymbol{T}_{syn}^{1\to2} = \begin{bmatrix} \boldsymbol{R}_{3\times3} & \boldsymbol{t}_{3\times1} \\ 0 & 1 \end{bmatrix} = \boldsymbol{T}_{fine}^{1\to2} \cdot \boldsymbol{T}_{coarse}^{1\to2} \tag{5.36}$$

Based on the synthetic transformation matrix ($\boldsymbol{T}_{syn}^{1\to2}$ or $\boldsymbol{T}_{syn}^{2\to1}$), the displacement vector (i.e., $\boldsymbol{D}_{disp} = [\Delta x \;\; \Delta y \;\; \Delta z]^T$) of every single point in the first epoch (x_{epoch_1}, y_{epoch_1}, z_{epoch_1}) or the second epoch (x_{epoch_2}, y_{epoch_2}, z_{epoch_2}) within the rigid area of the related patch can be directly calculated by

$$\begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \\ 0 \end{bmatrix} = \begin{bmatrix} x_{epoch\_2} \\ y_{epoch\_2} \\ z_{epoch\_2} \\ 1 \end{bmatrix} - \begin{bmatrix} x_{epoch\_1} \\ y_{epoch\_1} \\ z_{epoch\_1} \\ 1 \end{bmatrix} = (\boldsymbol{T}_{syn}^{1\to2} - \boldsymbol{I}_4) \begin{bmatrix} x_{epoch\_1} \\ y_{epoch\_1} \\ z_{epoch\_1} \\ 1 \end{bmatrix} = (\boldsymbol{I}_4 - \boldsymbol{T}_{syn}^{2\to1}) \begin{bmatrix} x_{epoch\_2} \\ y_{epoch\_2} \\ z_{epoch\_2} \\ 1 \end{bmatrix}, \tag{5.37}$$

where $\boldsymbol{I}_4$ is a 4×4 identity matrix. It should be emphasized that the points for deriving displacement vectors can be not only original measured points but also subsampled points or interpolated points as long as they are located within the associated rigid areas. The generalization from local ICP to local registration for point cloud-based deformation analysis enables more advanced and robust registration techniques integrated into this framework, thus producing more reliable and accurate deformation results.

Since the segmented quasi-rigid patches may still contain non-rigid parts (i.e., the distortion areas that cannot be represented by displacement vectors), the supervoxel-based method as elaborated in Chapter 4 can be used to automatically identify the non-rigid parts (i.e., the unstable areas identified in the registration process) and merely derive displacement vectors for the rigid parts. Thereby, the synthetic transformation matrix $\boldsymbol{T}_{syn}$ should be replaced by the estimated transformation matrix $\boldsymbol{T}_{final}$ in Equation (4.5).

5.3.2 Significance Test of Derived Displacement Vectors

Since the displacement vectors are derived by point cloud registration, standard deviations of deformation magnitudes are accordingly associated with the uncertainties of transformation parameters and involved points. There are different ways to roughly or precisely estimate the uncertainties of registered points, so as to calculate the standard deviations of displacement magnitudes. Lague *et al.* (2013) simply adopted an approximate constant value as the average registration error for all points that is further used for the significance test of deformations. Raffl and Holst (2022) created virtual target points (using the centroids of segmented patches) and converted them into polar pseudo-observations to estimate their uncertainties by free network adjustment. Besides these approaches, standard deviations of derived displacement vectors, in a rigorous way, can be estimated through the variance-covariance propagation law based on the registration process (by Equation (4.5)) and deformation quantification (by Equation (5.37)), and this approach has to simultaneously consider the stochastic model of the points in patches and its impact on the estimation of the synthetic transformation matrix $\boldsymbol{T}_{syn}$.

The deformation uncertainties are solely related to the estimated $\boldsymbol{T}_{syn}$ if the positions (e.g., measured, subsampled, or interpolated points) on which displacement vectors are imposed are assumed to have no errors. Thereby, the variance-covariance propagation could be simply performed based on Equation (5.37) with the VCM of transformation parameters. Herein, the VCM of $\boldsymbol{T}_{syn}$ is estimated by a least squares-based adjustment based on the last iteration of the point-to-plane ICP adopted in the proposed registration method.

In the point-to-plane ICP, the estimation of transformation parameters is regarded as a minimization problem of point-to-plane distances:

$$\hat{\boldsymbol{T}} = \underset{\boldsymbol{R},\boldsymbol{t}}{\operatorname{argmin}} \sum_{i=1}^{N} \left\| (\boldsymbol{R}\boldsymbol{q}_i + \boldsymbol{t} - \boldsymbol{p}_i) \cdot \boldsymbol{n}_{p_i} \right\|^2 , \tag{5.38}$$

where $\boldsymbol{q}_i$ is the point in the registered patch and $\boldsymbol{p}_i$ is its corresponding point in the reference patch ($\boldsymbol{q}_i$ and $\boldsymbol{p}_i$ are all located in the identified rigid areas), $\boldsymbol{n}_p$ is the normal vector at $\boldsymbol{p}_i$, N is the number of correspondences (i.e., identified rigid points), $\hat{\boldsymbol{T}}$ is the estimated transformation matrix in each ICP iteration.

Due to the small rotation angles from the last iteration of ICP, the estimated rotation matrix $\boldsymbol{R}$ can be written in a linear form. Then the optimal transformation matrix $\hat{\boldsymbol{T}}$ can be solved by a linear least squares:

$$\boldsymbol{V} = \boldsymbol{AX} - \boldsymbol{L} = (\hat{\boldsymbol{T}} \cdot \boldsymbol{q} - \boldsymbol{p}) \cdot \boldsymbol{n}_p = \left(\begin{pmatrix} 1 & -R_z & R_y & t_x \\ R_z & 1 & -R_x & t_y \\ -R_y & R_x & 1 & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} q_{ix} \\ q_{iy} \\ q_{iz} \\ 1 \end{pmatrix} - \begin{pmatrix} p_{ix} \\ p_{iy} \\ p_{iz} \\ 1 \end{pmatrix} \right) \cdot \begin{pmatrix} n_{ix} \\ n_{iy} \\ n_{iz} \\ 0 \end{pmatrix}^T$$

$$= \underbrace{\begin{pmatrix} n_{1z}q_{1y} - n_{1y}q_{1z} & n_{1x}q_{1z} - n_{1z}q_{1x} & n_{1y}q_{1x} - n_{1x}q_{1y} & n_{1x} & n_{1y} & n_{1z} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ n_{Nz}q_{Ny} - n_{Ny}q_{Nz} & n_{Nx}q_{Nz} - n_{Nz}q_{Nx} & n_{Ny}q_{Nx} - n_{Nx}q_{Ny} & n_{Nx} & n_{Ny} & n_{Nz} \end{pmatrix}}_{\boldsymbol{A}\ (N\times 6)} \cdot \underbrace{\begin{pmatrix} R_x \\ R_y \\ R_z \\ t_x \\ t_y \\ t_z \end{pmatrix}}_{\boldsymbol{X}\ (6\times 1)}$$

$$- \underbrace{\begin{pmatrix} n_{1x}(p_{1x} - q_{1x}) + n_{1y}(p_{1y} - q_{1y}) + n_{1z}(p_{1z} - q_{1z}) \\ \vdots \\ n_{Nx}(p_{Nx} - q_{Nx}) + n_{Ny}(p_{Ny} - q_{Ny}) + n_{Nz}(p_{Nz} - q_{Nz}) \end{pmatrix}}_{\boldsymbol{L}\ (N\times 1)} \qquad (5.39)$$

The estimated transformation parameters can be obtained by Equation (5.24) (i.e., the Gauss-Markov model) and their VCM is calculated by

$$\Sigma_{\boldsymbol{T}_{syn}} = \Sigma_{\hat{X}\hat{X}} = \sigma_0^2 \boldsymbol{Q}_{\hat{X}\hat{X}} \ \text{(with } \sigma_0^2 = \frac{\boldsymbol{V}^T \boldsymbol{PV}}{N-u} \text{ and } \boldsymbol{Q}_{\hat{X}\hat{X}} = (\boldsymbol{A}^T \boldsymbol{PA})^{-1}\text{)}, \qquad (5.40)$$

where $\boldsymbol{A}$ and $\boldsymbol{V}$ can be obtained in Equation (5.39), the weight matrix $\boldsymbol{P}$ is determined by the VCM of points involved as the correspondences in ICP if the stochastic information of measurements is given (otherwise $\boldsymbol{P}$ can be simply set as an identity matrix), $u = 6$ due to six transformation parameters.

As mentioned above, the uncertainties of positions $\boldsymbol{q}_{disp}$ (e.g., measured, subsampled or interpolated points within identified rigid areas) are assumed to be zero, and the calculated displacement vectors ($\boldsymbol{D}_{disp} = [\Delta x \ \Delta y \ \Delta z]^T$) are imposed on these positions. Therefore, the uncertainties of derived displacement vectors can be represented by the uncertainties of transformed positions $\boldsymbol{q}'_{disp}$. The calculation of $\boldsymbol{q}'_{disp}$ can be written in a linear form by moving the point coordinates to a coefficient matrix $\boldsymbol{F}$ as follows:

$$\boldsymbol{q}'_{disp,i} = \begin{pmatrix} q'_{ix} \\ q'_{iy} \\ q'_{iz} \end{pmatrix} = \boldsymbol{R} \cdot \boldsymbol{q}_{disp,i} + \boldsymbol{t} = \begin{pmatrix} 1 & -R_z & R_y \\ R_z & 1 & -R_x \\ -R_y & R_x & 1 \end{pmatrix} \cdot \begin{pmatrix} q_{ix} \\ q_{iy} \\ q_{iz} \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \\ t_z \end{pmatrix}$$

$$= \boldsymbol{F}\hat{\boldsymbol{X}} + \boldsymbol{q}_{disp,i} = \underbrace{\begin{pmatrix} 0 & q_{iz} & -q_{iy} & 1 & 0 & 0 \\ -q_{iz} & 0 & q_{ix} & 0 & 1 & 0 \\ q_{iy} & -q_{ix} & 0 & 0 & 0 & 1 \end{pmatrix}}_{\boldsymbol{F}} \cdot \underbrace{\begin{pmatrix} R_x \\ R_y \\ R_z \\ t_x \\ t_y \\ t_z \end{pmatrix}}_{\hat{\boldsymbol{X}}} + \begin{pmatrix} q_{ix} \\ q_{iy} \\ q_{iz} \end{pmatrix} \qquad (5.41)$$

Finally, the uncertainties for each displacement vector (measured by a VCM $\sum_{\boldsymbol{D}_{disp}}$) are estimated by the variance-covariance propagation law:

$$\begin{aligned}\sum\nolimits_{\boldsymbol{D}_{disp}} &= \sum\nolimits_{\boldsymbol{q}'_{disp,i}} = \boldsymbol{F} \sum\nolimits_{\hat{X}\hat{X}} \boldsymbol{F}^T \\ &= \boldsymbol{F}_{epoch\_1} \sum\nolimits_{\boldsymbol{T}_{syn}^{1\to 2}} \boldsymbol{F}_{epoch\_1}^T \;\; or \;\; = \boldsymbol{F}_{epoch\_2} \sum\nolimits_{\boldsymbol{T}_{syn}^{2\to 1}} \boldsymbol{F}_{epoch\_2}^T \end{aligned} . \tag{5.42}$$

Based on the derived VCM in Equation (5.42), the standard deviations of the displacement vector ($\hat{\sigma}_{Disp}$) and its components in three orthogonal directions ($\hat{\sigma}_{Disp_x}$, $\hat{\sigma}_{Disp_y}$, $\hat{\sigma}_{Disp_z}$) can be easily calculated by

$$\hat{\sigma}_{Disp_x} = \sqrt{\sum\nolimits_{\boldsymbol{D}_{disp}}(1,1)},\ \hat{\sigma}_{Disp_y} = \sqrt{\sum\nolimits_{\boldsymbol{D}_{disp}}(2,2)},\ \hat{\sigma}_{Disp_z} = \sqrt{\sum\nolimits_{\boldsymbol{D}_{disp}}(3,3)},\ \hat{\sigma}_{Disp} = \sqrt{\hat{\sigma}^2_{Disp_x} + \hat{\sigma}^2_{Disp_y} + \hat{\sigma}^2_{Disp_z}} . \tag{5.43}$$

Furthermore, with the estimated VCM for each displacement vector, a statistical significance test can be performed by the F-test. The test quantity is calculated by

$$F_{disp} = \frac{\boldsymbol{D}_{disp}^T \left(\sum_{\boldsymbol{D}_{disp}}\right)^{-1} \boldsymbol{D}_{disp}}{p}, \tag{5.44}$$

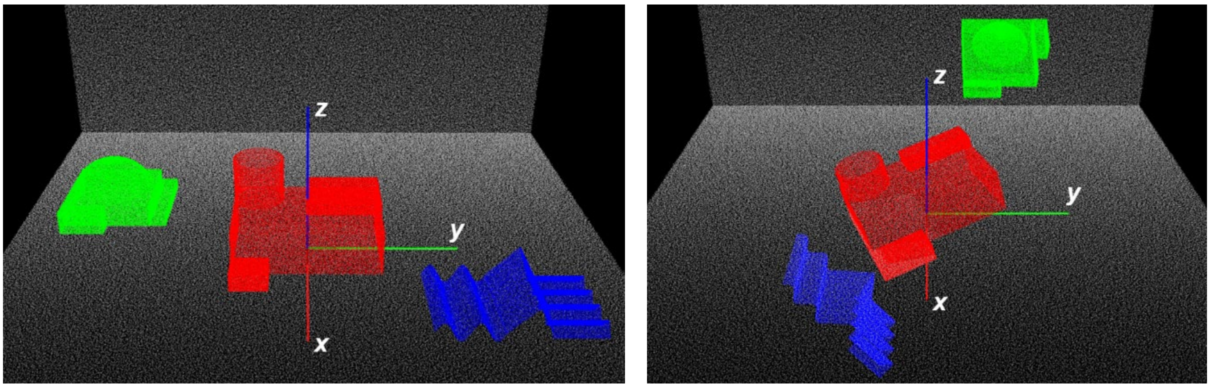
where p is the dimension of the displacement vectors (i.e., $p = 3$) and the displacement vector $\boldsymbol{D}_{disp} = [\Delta x \;\; \Delta y \;\; \Delta z]^T$ is calculated by Equation (5.37). The reference test quantity $F_{p,f,1-\alpha/2}$ can be taken from the table of F-distribution and the degrees of freedom $f = N - 6$.

5.3.3 Experimental Evaluation

5.3.3.1 Simulated Data

In order to demonstrate and evaluate the proposed local registration-based deformation analysis approach, two-epoch point clouds are simulated, each containing three objects and a background. In this simulated scene, as shown in Figure 5.42, rigid-body movements (including rotations and translations) are individually imposed on the three objects by defining different transformation parameters, which subsequently serve as the references for evaluation. These objects exhibit diverse dimensions and geometries by assembling different small parts such as planes, cuboids, and spheres. Inhomogeneous point densities (with point spacing varying from 1 cm to 5 cm) and Gaussian-distributed noise (with standard deviations varying from 5 mm to 2 cm) are imposed on different areas of each object. In addition, partial non-rigid areas are simulated to validate the performance of automatic identification of rigid areas by the registration method proposed in Chapter 4. These simulated objects with predefined geometries are used to represent the quasi-rigid areas introduced in Section 3.1.2 herein since simulating patches with arbitrary shapes in natural scenes is still challenging.

In Figure 5.42(a), the first object (red) is comprised of three cuboids and a cylinder. Except for the largest cuboid, partial shape deformations are simulated on the other three geometric primitives: one cuboid is extended along the y-axis, the other cuboid is compressed along the x-axis, and the height of the cylinder is reduced. The second object (green) consists of three cuboids and a half sphere. The biggest cuboid also stays rigid, while the front cuboid is compressed along the z-axis and the right cuboid is compressed along the x-axis and extended along the *z*-axis. The half sphere is shrunk by reducing its radius. The third object is comprised of two adjacent planes that keep rigid and two sine surfaces with different amplitudes and phases. Only two curved surfaces are deformed by decreasing (the left one) and increasing (the right one) their amplitudes in the second epoch to simulate the non-rigid areas.



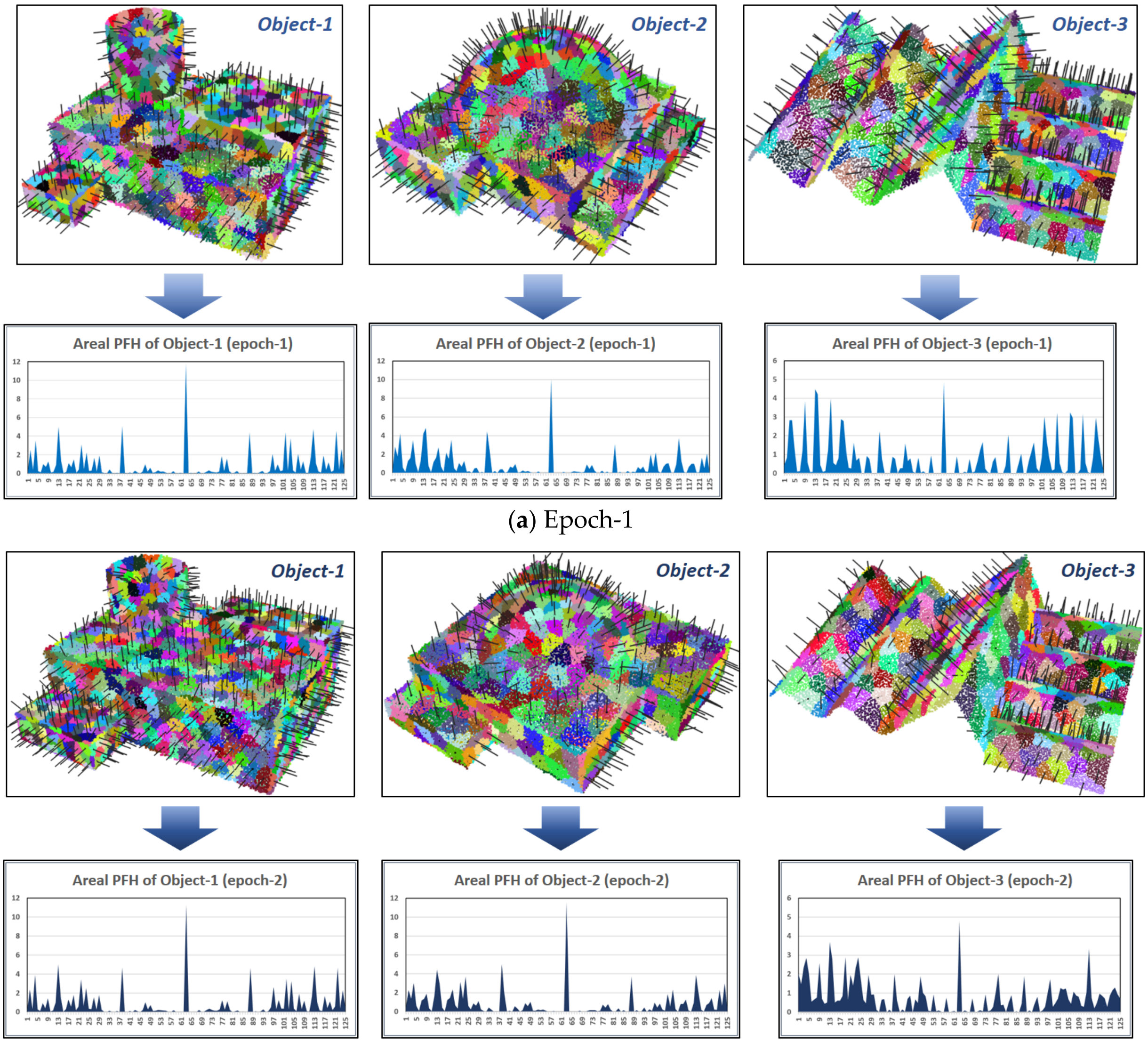
(**a**) Epoch-1 (**b**) Epoch-2

Figure 5.42: Three simulated objects with different geometries and partial non-rigid areas in two epochs. Rotations and translations are imposed on each object to simulate deformations.

Based on the simulated point clouds in the first epoch, rigid-body movements are respectively imposed on three objects to generate the second-epoch point cloud. This is achieved by utilizing three sets of defined transformation parameters. Specifically, the first object is rotated around both the *x*-axis and the *z*-axis, the second object is rotated around the *y*-axis and then translated, and the third object is translated, followed by a rotation around the *x*-axis. Transformed objects in the second epoch are shown in Figure 5.42(b). These simulated transformations represent object movements in various directions and with different magnitudes to a certain extent.

As the workflow presented in Figure 5.41, segmentation ought to be performed to separate each quasi-rigid patch (i.e., individual objects in this simulation) from the scenes of two epochs. RANSAC-based plane segmentation is first used to remove the background (simulated by two orthogonal planes), then individual objects are extracted by Euclidean clustering (Rusu, 2009). Based on the constructed areal PFH descriptors for each patch using the centroids and normals of its supervoxels (as shown in Figure 5.43), correspondences are established by finding the pair indicating the most correlated feature descriptors. The correlation matrix between three patches in two epochs and the established correspondences in the scenario are shown in Figure 5.44. It can be seen that objects are all correctly matched between epochs by high correlation coefficients even though inconsistent noise, large movements, and partial shape deformations are imposed on them.

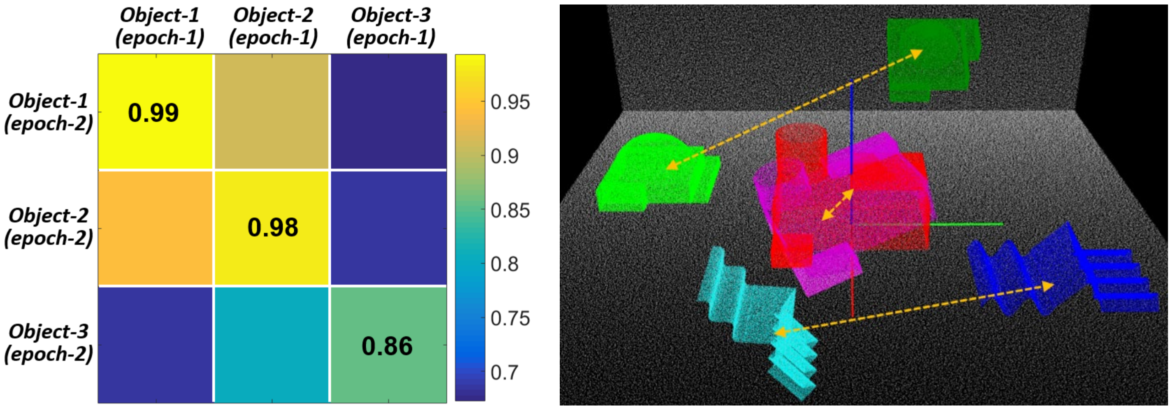
The supervoxel-based registration method proposed in Chapter 4 is utilized to align the patch in epoch-1 to its corresponding one in epoch-2, deriving a synthetic transformation matrix and identifying the rigid parts simultaneously. In order to remain more points as the rigid part, the initial size of supervoxels for local registration is set smaller than that in the areal feature calculation. Identified rigid/non-rigid parts and registered patches are presented in Figure 5.45. The ratios of identified rigid points for three objects are 86.2%, 68.3% and 43.4%, respectively. It shows that the identified non-rigid areas (see red parts) agree well with the simulated shape deformations. All three quasi-rigid patches (objects) are precisely aligned based on identified rigid areas (i.e., the identified stable areas after registration).



(**a**) Epoch-1

(**b**) Epoch-2

Figure 5.43: Segmented quasi-rigid patches (objects) represented by their supervoxels' centroids and normals and their respective areal PFHs in two epochs.



(**a**) (**b**)

Figure 5.44: The correlation matrix between every two patches (objects) in two epochs (a) and established corresponding pair of patches by the correlation of features descriptors (b).

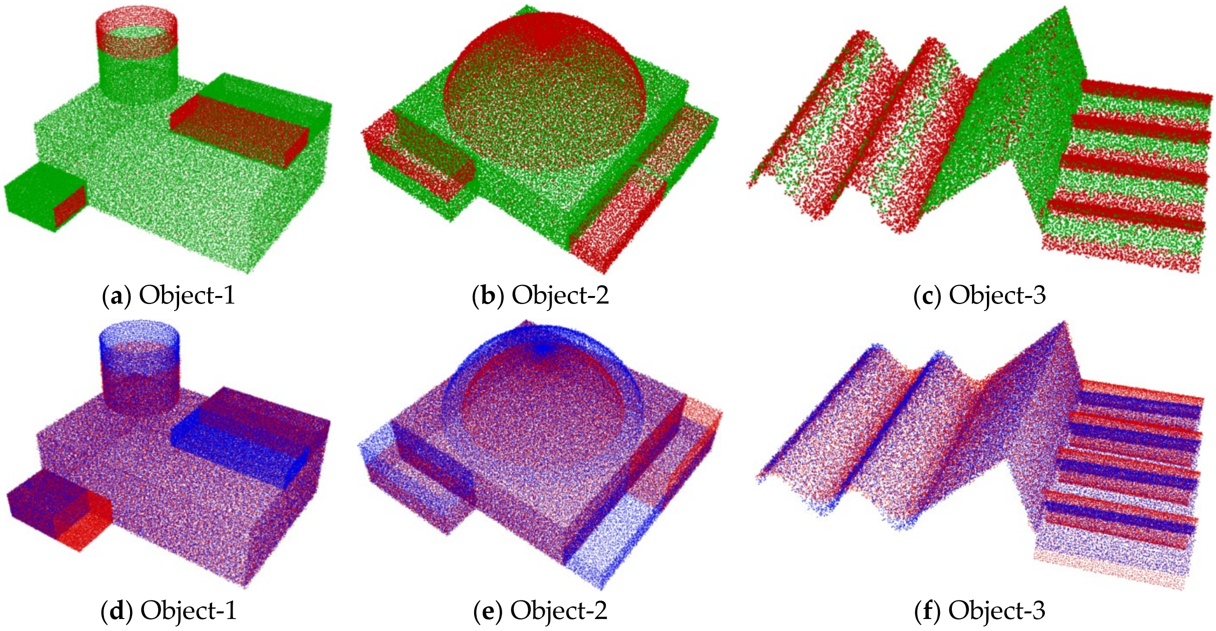


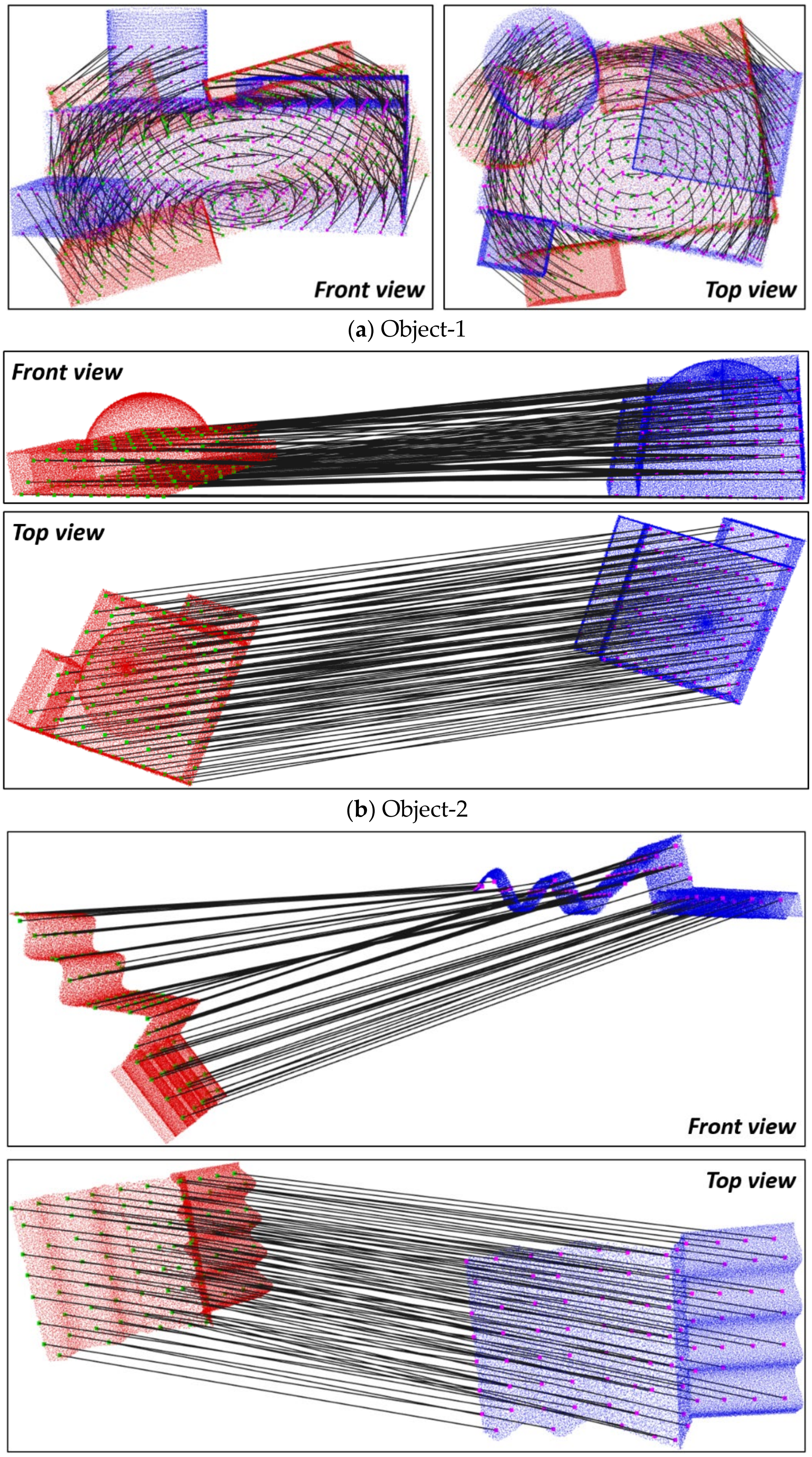
Figure 5.45: The identified rigid (green) and non-rigid (red) parts in each patch (object) by the proposed registration method (a-c) and the registered patches in epoch-1 (blue) and epoch-2 (red) (d-f).

To quantitatively evaluate the absolute accuracy of local registration for segmented quasi-rigid patches, the differences between the estimated and the simulated (used as the reference) transformation parameters are calculated and given in Table 5.5. The registration of the first two objects exhibits fairly good results, with almost submillimeter-level accuracy in translations. In contrast, the errors of object-3 are noticeably larger than those of the other two, especially in the y- and z-axis directions. This can be attributed to the symmetric geometry of object-3, which leads to a local optimization in the ICP process. These systematic shifts are further propagated to the errors of derived displacement vectors in the respective directions.

Table 5.5: Absolute errors of transformation parameters by the local registration for three objects.

Object	Rotation angle (°)			Translation (mm)		
	ΔR_x	ΔR_y	ΔR_z	Δt_x	Δt_y	Δt_z
1	0.002	0.001	0.016	0.32	0.25	0.48
2	0.012	0.001	0.002	0.80	0.01	1.26
3	0.721	0.019	0.015	10.26	49.10	55.63

Based on the estimated synthetic transformation matrix and identified rigid areas for each patch (object), realistic displacement vectors can be calculated at any position within the rigid areas, yielding a dense 3D displacement field. To enhance the clarity of visualized results, displacement vectors are imposed merely on uniformly subsampled positions, as demonstrated in Figure 5.46. These displacement vectors implicitly establish correspondences between two-epoch patches, avoiding the use of distinct feature points that can be challenging to detect on smooth surfaces. It can be seen that the displacement vectors are correctly derived within rigid areas for the three segmented patches (individual objects), even in cases of large translations (as in object-2) or rotations (as in object-3) occurring between the two epochs. The dense displacement vectors improve the interpretation of deformations in comparison to the point-wise ways, providing more intuitive representations of the deformation process. Furthermore, provided high-frequency acquisitions of the point cloud data, continuous motion tracks of specific points on a rigid object can be precisely revealed.



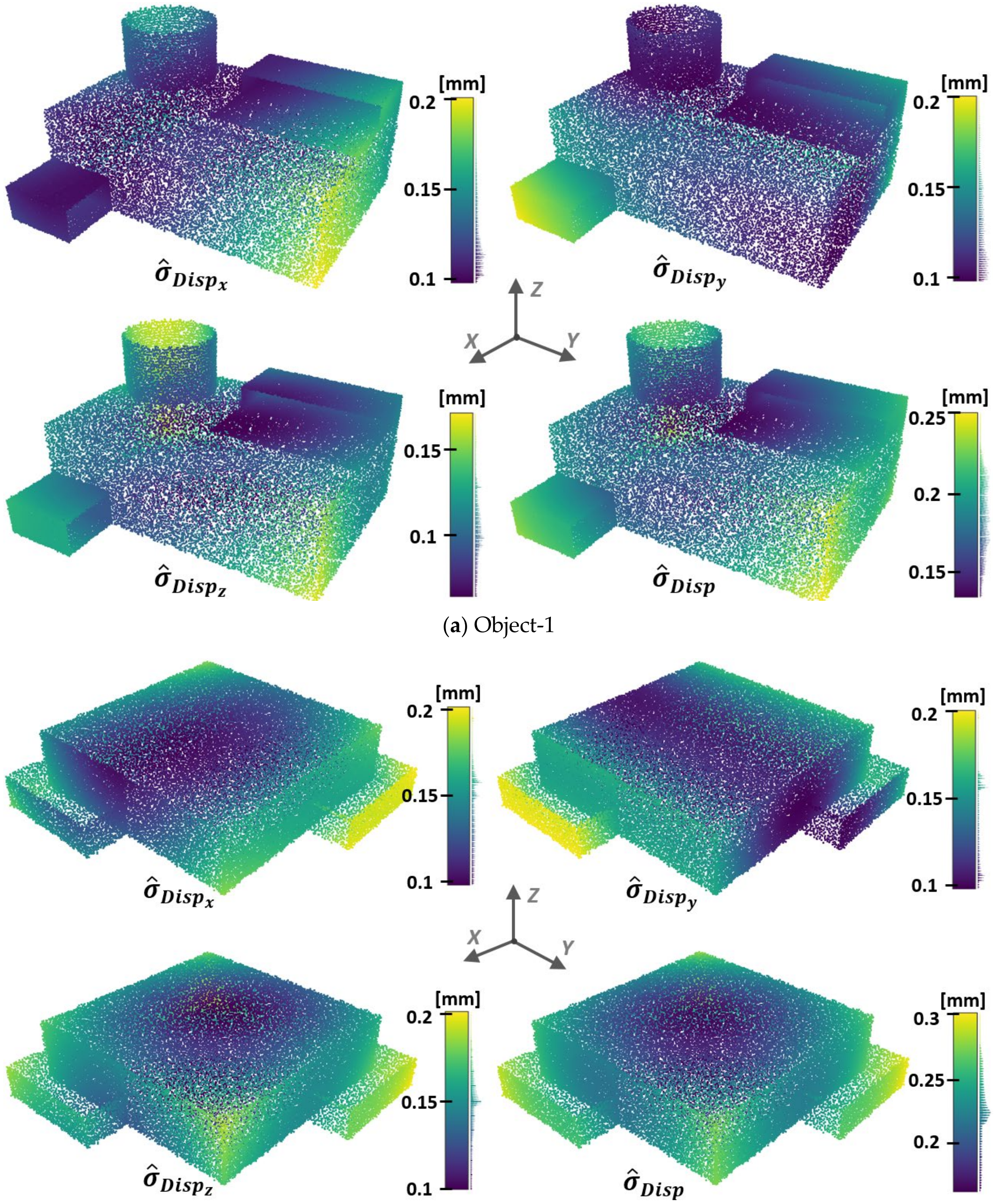
(**a**) Object-1

(**b**) Object-2

(**c**) Object-3

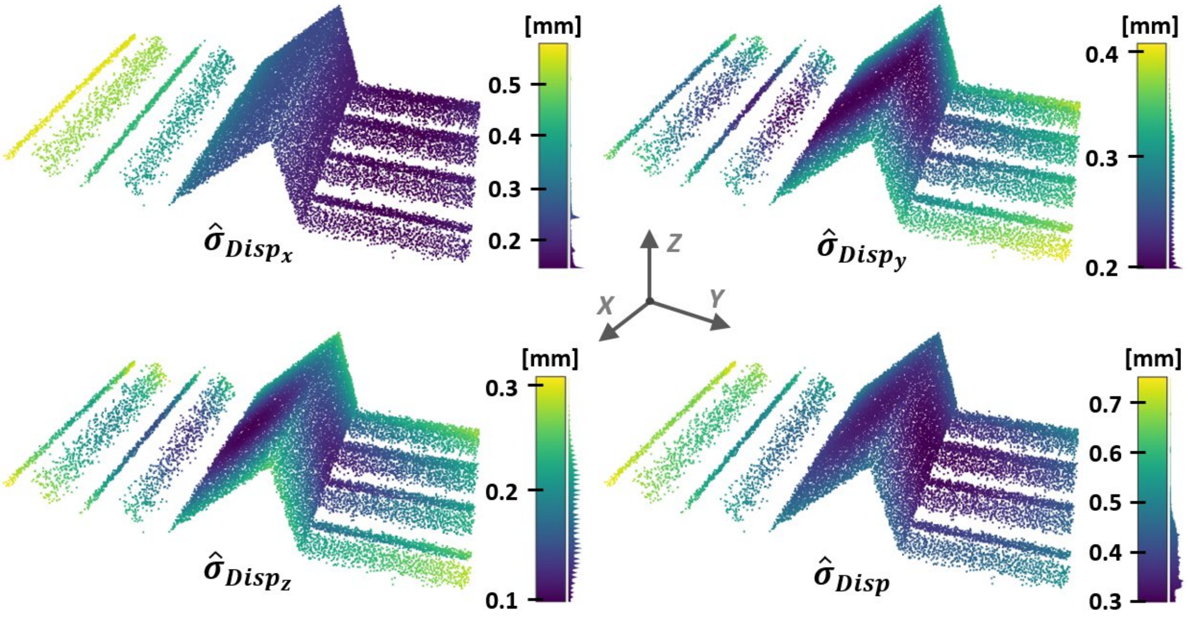
Figure 5.46: Derived dense displacement vectors (black lines) for each object by the local registration-based method (the red patch and the blue patch are in epoch-1 and epoch-2, respectively; Green points are sampled positions in epoch-1 for displaying derived displacement vectors and magenta points are the corresponding positions in epoch-2).

The standard deviations of displacement vectors are calculated by Equation (5.43), based on the derived VCM of transformation parameters and through the variance-covariance propagation law (see Section 5.3.2). These derived standard deviations ($\hat{\sigma}_{Disp}$), including their components in three orthogonal directions, are projected onto the registered objects (only within the identified rigid areas). Figure 5.47 presents the magnitudes and spatial distributions of these standard deviations. Simplified histograms of the standard deviations are also displayed alongside the color bar, indicating their frequency distributions. These standard deviations also represent the (relative) uncertainties of each registered point, which offers a means to roughly assess the registration precision.



(a) Object-1

(b) Object-2



(**c**) Object-3

Figure 5.47: Estimated standard deviations of displacement vectors imposed on the identified rigid areas after the registration process.

It is evident that the uncertainties of displacement vectors (or registered points) are inhomogeneous on the object's surface. Except for object-3, the standard deviations from the other two objects are below 0.5 mm. The higher uncertainties (less than 1 mm) observed on object-3 result from the imposed high noise in the simulation process. For all three objects, the standard deviations near the central areas are generally smaller than those near the peripheral areas. Based on the VCM of displacement vectors, significance tests (F-test) can be performed following the procedure outlined in Section 5.3.2. Due to the small magnitudes of uncertainties, all presented displacement vectors in Figure 5.46 are tested as significant, highlighting the capability of the local registration-based method in detecting small deformations.

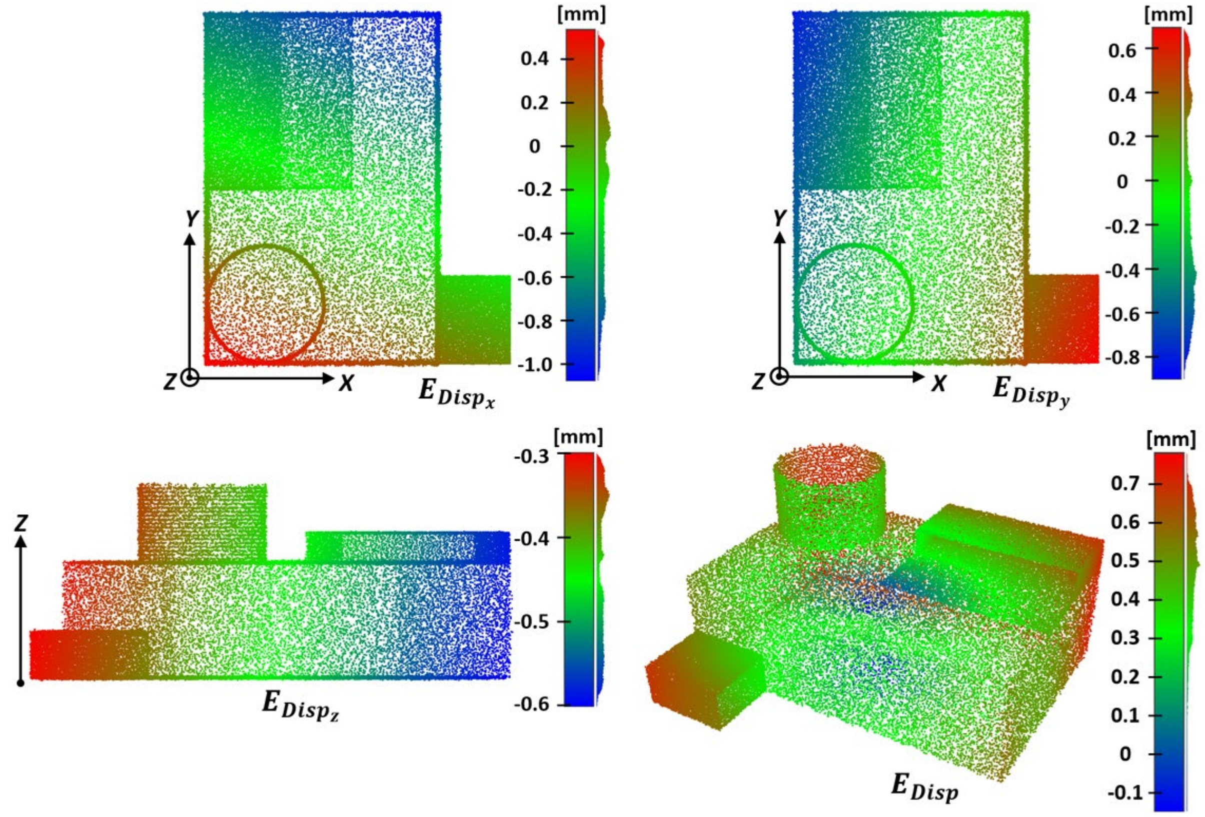
Since all deformations (only rigid-body movements here) of each object are simulated, the absolute errors of displacement vectors in the three-axis components (E_{Disp_x}, E_{Disp_y}, E_{Disp_z}) can be calculated by

$$\begin{bmatrix} E_{Disp_x} \\ E_{Disp_y} \\ E_{Disp_z} \\ 0 \end{bmatrix} = \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \\ 1 \end{bmatrix} - \begin{bmatrix} \Delta x_{ref} \\ \Delta y_{ref} \\ \Delta z_{ref} \\ 1 \end{bmatrix} = (\boldsymbol{T}^{1\to2}_{estimated} - \boldsymbol{T}^{1\to2}_{simulated}) \begin{bmatrix} x_{epoch\_1} \\ y_{epoch\_1} \\ z_{epoch\_1} \\ 1 \end{bmatrix}, \tag{5.45}$$

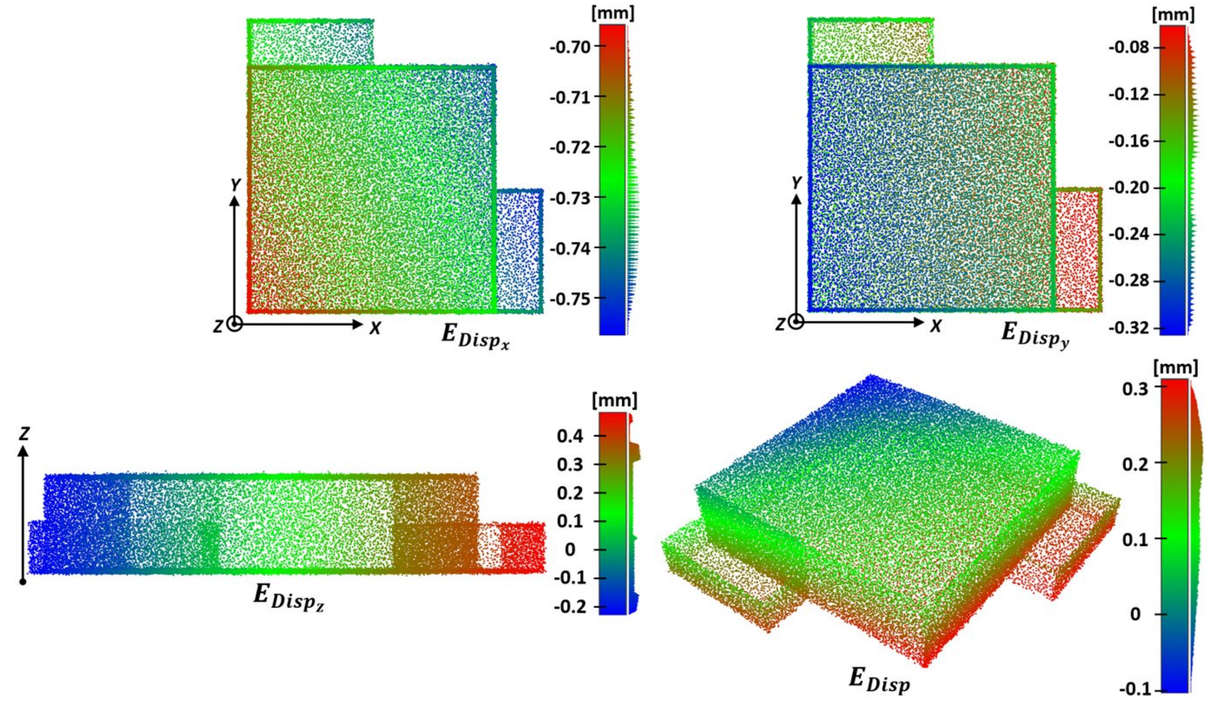
and thus the error of displacement vector's magnitude E_{Disp} is

$$E_{Disp} = \sqrt{\Delta x^2 + \Delta y^2 + \Delta z^2} - \sqrt{\Delta x_{ref}^2 + \Delta y_{ref}^2 + \Delta z_{ref}^2}\ , \tag{5.46}$$

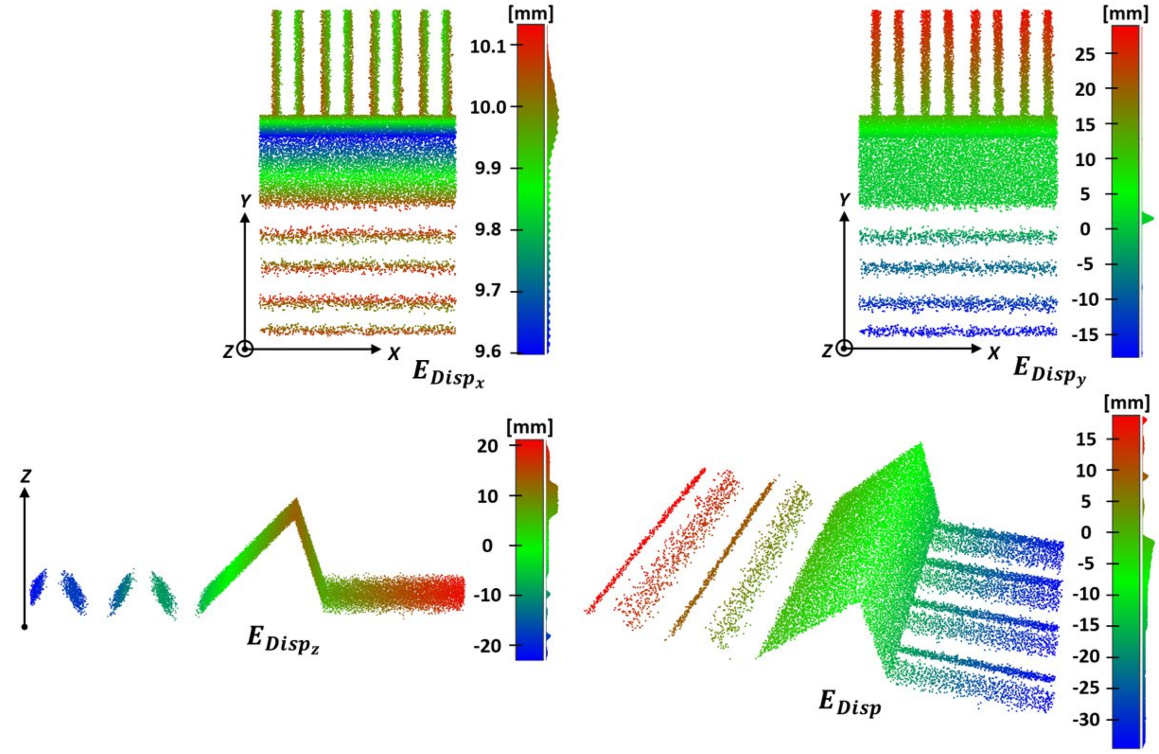
Figure 5.48 presents the calculated absolute errors (E_{Disp}) of displacement vectors (including their components in three orthogonal directions (E_{Disp_x}, E_{Disp_y}, E_{Disp_z})). The values of these errors are imposed on the identified rigid areas of each object (in epoch-1), thus showing their spatial distributions on the objects. The frequency distributions of these absolute errors are displayed next to the color bar.



(**a**) Object-1



(**b**) Object-2

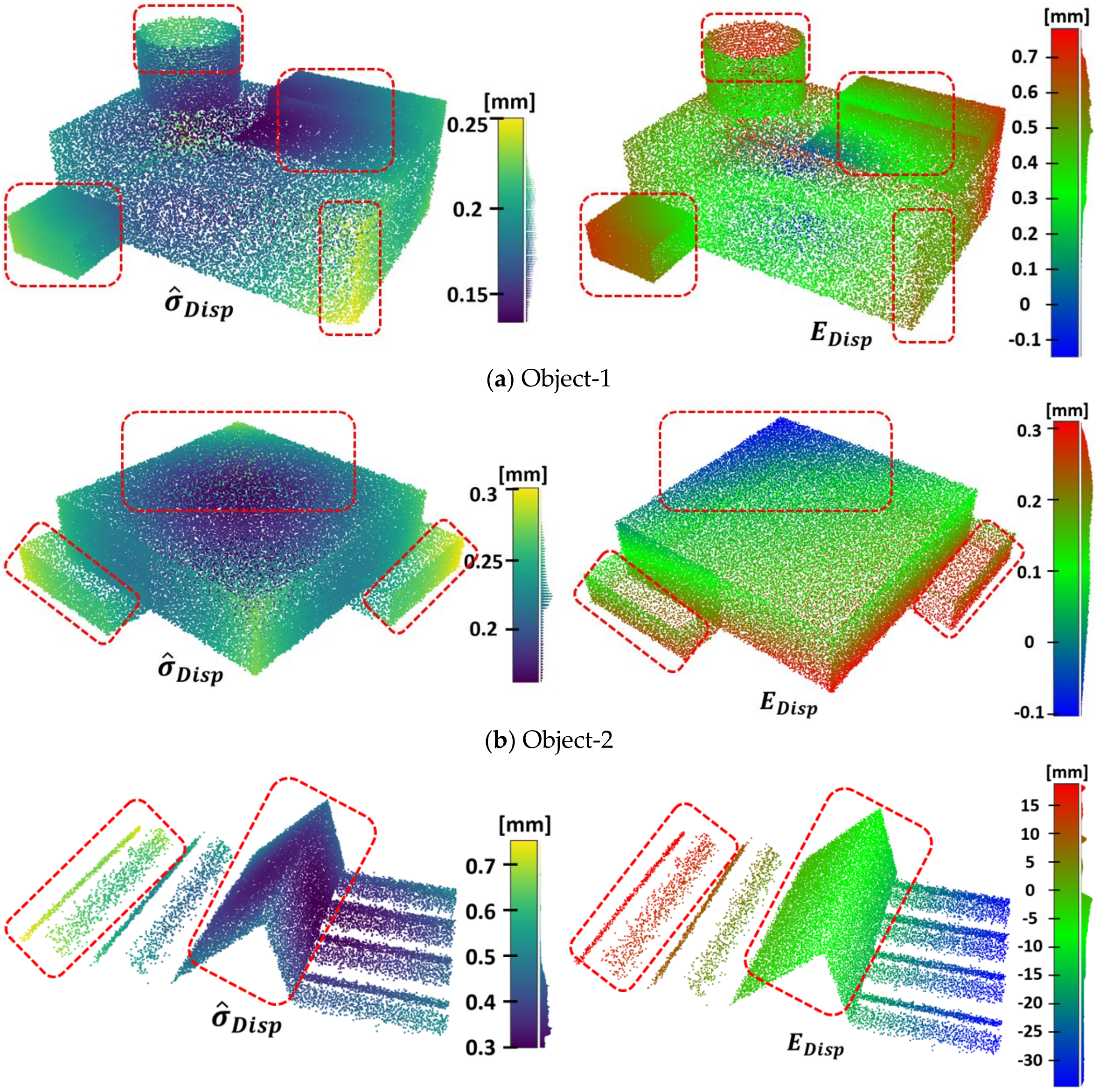


(**c**) Object-3

Figure 5.48: Absolute errors of displacement vectors imposed on the identified rigid areas on which displacement vectors are displayed.

Object-1 and object-2 exhibit submillimeter-level (absolute) accuracy owing to accurate estimations of transformation parameters while still larger than their standard deviations (relative errors), as depicted in Figure 5.47. The error distributions of object-3 present large systematic shifts, which agree with the absolute errors of the estimated transformation parameters (see Table 5.5). This is due to the local optimization that occurs in the ICP process which follows an inaccurate coarse registration (performed by 4PCS). Similar to the standard deviations ($\hat{\sigma}_{Disp}$) estimated based on the VCM of transformation parameters, the absolute errors near the central areas are typically smaller than those near the peripheral areas for all three objects. Notably, for object-3, the accuracy in the central areas (comprising two planes without deformations) is much higher than that in the two side areas with partial deformations (see Figure 5.48(c)). This can be attributed to the fact that the centroid of identified rigid areas is located in the central regions, where the standard deviations of transformation parameters are minimized due to the ICP-based estimation.

Although the magnitudes of absolute errors (in Figure 5.48) are considerably larger than the standard deviations (in Figure 5.47) for all three objects, there is a certain level of consistency observed in the spatial distributions (i.e., the change of magnitudes with respect to the position on the surface) of E_{Disp} and $\hat{\sigma}_{Disp}$. This consistency is highlighted within the red dashed frames in Figure 5.49. The differences are primarily caused by the systematic shifts of estimated transformation parameters and respective geometries of objects. Consequently, the uncertainties of displacement vectors (as well as the registered points) are underestimated when relying on the standard deviations derived from the approximate VCM of transformation parameters. Nevertheless, these standard deviations still provide an approximate representation of the spatial variations within error distributions, thus allowing for an evaluation of the registration quality to some extent.



(**a**) Object-1

(**b**) Object-2

(**c**) Object-3

Figure 5.49: Comparison between the spatial distributions of E_{Disp} and $\hat{\sigma}_{Disp}$ (red dashed frames highlight the similarities).

To evaluate the performance of the local registration-based method for deriving displacement vectors under different rotations and translations, different sets of transformation parameters are imposed on respective simulated objects, generating objects with varying positions and orientations in epoch-2. The new simulated scenes, the absolute errors of transformation parameters from the local registration, and corresponding error histograms are provided in Appendix A.5. From the results in Figure A.4, the absolute errors of derived displacement vectors are all below 1 mm for object-1 and object-2, while object-3 continues to exhibit lower accuracy at the centimeter-level. These error magnitudes are basically consistent with the results in Figure 5.48. Thereby, the magnitudes of rigid-body movements (including rotations and translations) have an insignificant impact on the accuracy of derived displacement vectors in this simulation. This can be explained by the fact that the point clouds to be registered, even with varying initial positions and orientations, do not differ much after successful coarse registration.

5.3.3.2 Multiple Spheres

Although the local registration-based method is not suitable for deriving displacement vectors on the surfaces of the geometric primitives with insufficient surface variations or symmetric structures, the key features (or parameters) of a geometric primitive can be used as the positions to compute displacement vectors if rotations or translations do not change these features. For example, the rotation (around the cylindrical axis) or the translation (along the cylindrical axis) will not affect the orientation of the axis, and the 3D rotations (around the sphere center) will not affect the position of the sphere center. Therefore, even if these kinds of systematic shifts occur after the local registration due to insufficient surface variations or symmetric structures, the displacements calculated on these features theoretically remain unchanged. Especially for spheres, even if the spherical points used for the local registration are located in different areas of the sphere in two epochs, the displacement of the sphere center would be consistent with the estimated parametric deformation (i.e., the distance between estimated sphere centers in two epochs), as illustrated in Figure 5.50.

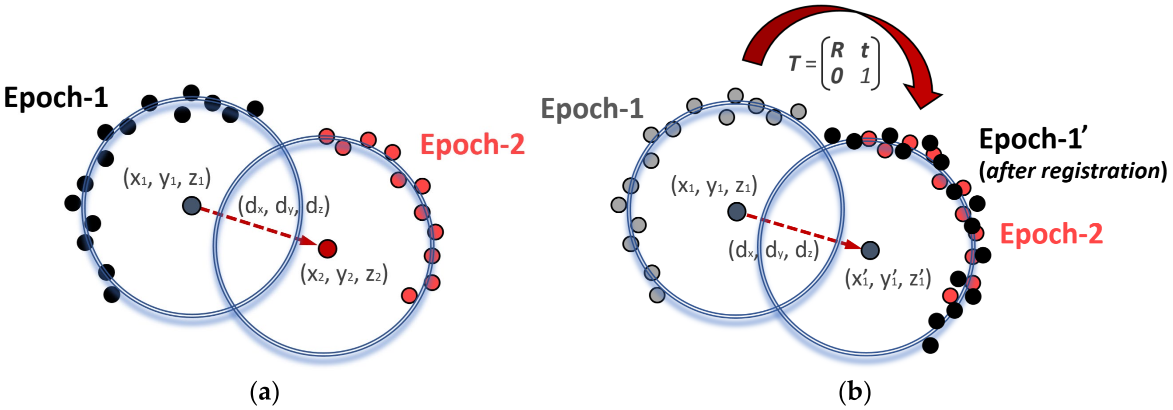
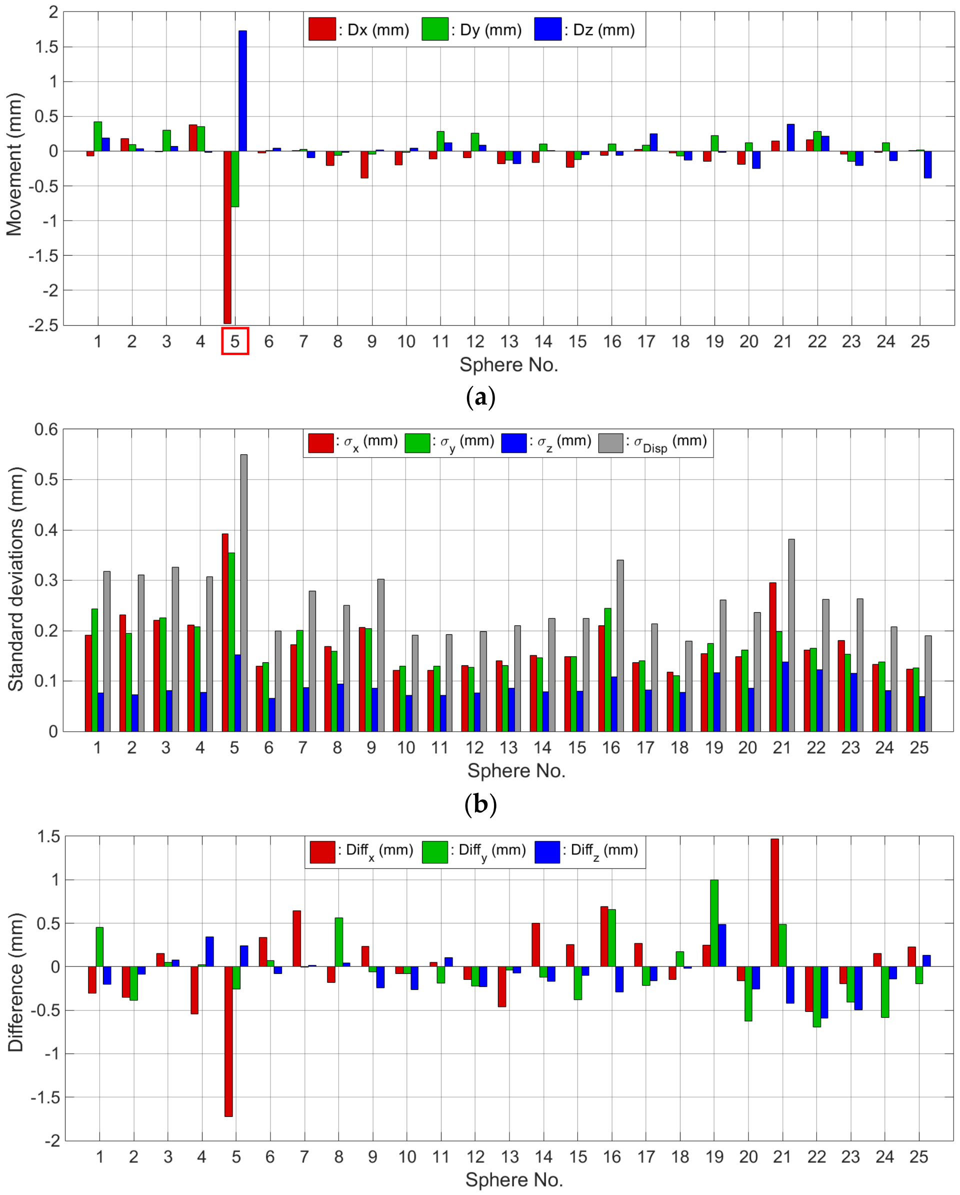


Figure 5.50: Difference between geometric primitive-based and local registration-based methods for deriving the displacement of a sphere center.

To evaluate the feasibility of the local registration-based method on real data, displacement vectors are derived on the centers of multiple spheres based on the data of segmented spherical points in Section 5.2.3. For demonstration, the data in epoch-4 (three significantly moved hollow spheres) are re-evaluated by the proposed local registration method and compared to the results by the geometric primitive-based method (see Section 5.2.3.6). Figure 5.51 presents the derived displacements of sphere centers and associated standard deviations by the local registration-based method. The significance test is also performed by the procedure in Section 5.3.2 (significant movements are highlighted by a red frame on the horizontal axis).

Except for Sphere-5 (showing much larger movements), deformations calculated by the local registration-based method are all non-significant (below 0.5 mm in all three orthogonal directions). In Figure 5.51(b), the average and the maximum standard deviations of displacement vectors ($\hat{\sigma}_{Disp}$) are 0.26 mm and 0.55 mm, respectively, which agree with the standard deviations estimated by the geometric primitive-based method (see epoch-4 in Table 5.3). Thus, the standard deviations estimated by the two methods show consistency in this experiment. Besides, standard deviations in horizontal directions ($\hat{\sigma}_{Disp_x}$ and $\hat{\sigma}_{Disp_y}$) are higher than those in the vertical direction ($\hat{\sigma}_{Disp_z}$). This discrepancy is attributed to the horizontal distribution of spherical points. By comparing with Figure 5.40(c), differences of calculated movements in three directions between the two methods are shown in Figure 5.51(c). Except for Sphere-5 and Sphere-21, all difference values are less than 1 mm, which are mainly caused by measurement uncertainties. As the reference values for actual movements of these spheres, which can be used to evaluate the absolute errors of estimated deformations, are currently unavailable, explanations regarding the significant differences in results between the two approaches are not given herein. Further investigations into this matter are left for future work.



(a)

(b)

(c)

Figure 5.51: Displacements of sphere centers (a), associated standard deviations by the local registration-based method (b), and differences to the results by the geometric primitive-based method (c).

5.3.4 Discussion

5.3.4.1 Error Analysis

The accuracy of 3D displacement vectors derived via the local registration-based method relies directly on the registration accuracy. The derived transformation parameters may contain both systematic and random errors. The random errors (i.e., standard deviations) can be estimated by the adjustment process (e.g., using linear least squares depicted in Section 5.3.2). However, the systematic shifts caused by the local optimization in the registration process cannot be detected without other information. Hence, the absolute errors are usually larger than the estimated standard deviations (as presented and analyzed in the experiment with simulated data). Besides, the simplified consideration of the stochastic model of point clouds in estimating transformation parameters (e.g., using an identity matrix in both experiments) may also underestimate the uncertainties of derived deformations. According to the experimental results, the accuracy of displacement vectors or transformation parameters depends strongly on the geometries and measurement uncertainties of the patches (objects), but is less related to the magnitudes of rigid-body movements (including rotation and translations). This can be explained by the fact that the coarse registration step could provide a good initial pose between two patches.

5.3.4.2 Strengths and Limitations

This thesis has extended the existing local ICP-based method to the local registration by introducing a synthetic transformation matrix, thus allowing for the introduction of better-performing registration algorithms to improve the reliability and accuracy of derived displacement vectors. Herein, the proposed targetless registration method (see Chapter 4) is employed to register two quasi-rigid patches, simultaneously identifying the rigid and non-rigid parts. Thus, the influence of non-rigid parts on the registration process can be avoided.

The derived real displacement vectors enable the representation of both out-of-plane and in-plane movements, which the classical point cloud comparison techniques (e.g., C2C, C2M, M3C2, etc.) may not correctly quantify. In view of its simplicity and high accuracy, the local registration-based deformation analysis exhibits potential applications for the deformation monitoring of large rigid bodies like buildings with non-uniform settlements, large industrial equipment, larger boulders, and individual trees in rockslides or landslides, etc. However, practical applications are not presented in this thesis yet due to the lack of real-world datasets with available ground truth for validation and evaluation purposes. Similar approaches (mainly using a local ICP) applied in the monitoring of geological phenomena can be found in Oppikofer *et al.* (2009) and Weidner *et al.* (2021).

Limitations still existing in the local registration-based strategy are summarized as follows:

(1) Efficient segmentation of quasi-rigid patches from continuous surfaces is still challenging without prior knowledge, especially in a scene containing highly irregular boundaries of rigid areas (e.g., landslide).
(2) Local optimization in the registration process induced by insufficient surface variations or symmetric structures cannot always be avoided, thus more robust and reliable coarse registration should be considered.
(3) The stochastic model (i.e., VCM) of point clouds is simplified to an identity matrix in the estimation of transformation parameters and their associated uncertainties herein. Possible improvements can be achieved by introducing a more realistic stochastic model (see Section 6.2).

5.4 Method Choice of Point Cloud-based Deformation Analysis

The surface geometry and its representation by captured 3D point clouds of the real world vary widely, and there is, to the best of the author's knowledge, no universal method of point cloud-based deformation analysis to cope with all scenarios perfectly and meet all requirements. The complexity of the deformation process leads to the diversity of relevant methods as well. In this thesis, three proposed methods for point cloud-based deformation analysis are all on the basis of patch-based segmentation under local rigidity and/or local consistency assumptions. Each method exhibits particular advantages in dealing with specific applicable scenes. Benefiting from the utilization of segmented rigid patches, the influence of measurement noise on quantifying deformations is reduced, yielding lower uncertainties. Besides, statistical significance tests can be performed on derived deformation values based on the empirically estimated stochastic information from these rigid patches.

The main characteristics and applicable scenarios of three rigid patch-based methods are listed in Table 5.6. The planar patch (supervoxel)-based method (i.e., PB-M3C2) computes surface distances along defined directions, whereas it is not sensitive to detect in-plane movements. The local registration-based method is able to output real displacement vectors and reveal all rigid-body movements. For the geometric primitive-based method, the type of estimated deformations depends on defined parameters. For instance, if the parameters are with respect to positions, shapes, or orientations, the derived deformations can be regarded as displacement vectors (e.g., the center of a sphere), volumetric changes (e.g., the radius of a cylinder), and orientation changes (e.g., the normal vector of a plane), respectively. Both PB-M3C2 and the local registration-based method can generate high-resolution results, while the resolution of derived deformations relies on the number of associated parameters in the geometric primitive-based method.

Table 5.6: Main characteristics and applicable scenes of three proposed deformation analysis methods based on rigid patches.

Methods	Rigid patches	Type of correspondence	Type of derived deformations	Resolution of derived deformations	Applicable scenes	Limitations
PB-M3C2	Small planar patches (e.g., supervoxels)	Positions along defined directions	Surface distances along defined directions	High	Any kind of surface with deformations and changes in complex topographies (e.g., landslide surfaces)	Insensitive to in-plane movements
Geometric primitive-based	Geometric primitives (e.g., planes, spheres)	Defined parameters	Parametric deformations like movements, orientation changes	Low	Scenarios with geometric primitives, or parametric changes are required (e.g., buildings, industrial facilities, etc.)	Prior knowledge of primitive information is required
Local registration-based	Quasi-rigid areas	Positions connected by rigid transformations	Dense displacement vectors	High	Locally rigid surfaces or individual objects with rich surface variations (e.g., boulders on landslides, vehicles on roads, etc.)	Infeasible to fully deformed (distorted) surfaces

The choice of point cloud-based deformation analysis methods for specific scenarios should consider the following aspects:

(1) The surface geometries of monitored objects (e.g., dimension, shape, roughness, etc.),
(2) the characteristics and quality of acquired point clouds (e.g., point density, spatial resolution, coverage, accuracy, etc.),
(3) the deformation process (e.g., deformation types (see Figure 2.4), directions (in-plane or out of plane) and magnitudes), and
(4) the requirements of the monitoring task (e.g., the areas of interest, whether surface distances or displacement vectors are needed, the temporal and spatial resolution of deformation representation, etc.).

When monitoring the displacements of a bridge, for example, the geometric primitive-based method can be effectively applied if well-distributed geometric primitives, such as planes and paraboloids, are prevalent on the bridge's surface. However, if certain areas of the surface are challenging to extract and parameterize, the local registration-based method is a recommended approach, especially when prior knowledge suggests that the areas of interest exhibit partial rigidity. Furthermore, in cases where some regions of the bridge's surface become quite rough due to human activities or environmental factors, such as erosion or the attachment of foreign objects, PB-M3C2 offers the capability to directly detect and quantify these surface changes.

Complex scenes or tasks with specific requirements may benefit from combining different methods proposed in this thesis. For instance, PB-M3C2 can be applied to detect changes in the underlying surfaces of a landslide, while the local registration-based method is employed to calculate the movements of trees or boulders situated on the landslide. These monitored objects in the scene can be separated by point cloud classification techniques prior to conducting deformation analysis.

6 Conclusions and Outlook

6.1 Summary

Laser scanning technology has facilitated the efficient acquisition of high-precision and high-resolution area-wise spatial data, allowing for a better revelation of complex geometric changes and the deformation process of the dynamic 3D world. Despite the rapid development and wide application of TLS in areal geodetic monitoring, several challenges are not solved satisfactorily yet in the processing chain of TLS-based deformation monitoring. Registration and deformation analysis of TLS point clouds, as two crucial parts, have been intensively investigated and developed in the community of engineering geodesy, aiming to achieve better representation and interpretation of the deformations of monitored objects. In this context, the thesis specifically proposed novel methods to improve the performance of targetless registration and point cloud-based deformation analysis by utilizing the segmented rigid patches from 3D point clouds.

Differences between multi-epoch 3D point clouds are the direct reflection of geometric surface changes in the real world. Conversely, mechanical phenomena during the deformation process are to a certain extent manifested in captured point clouds. In this thesis, local rigidity and local consistency are assumed in the deformation process, taking into account the physical properties of deformed objects, and thus are transferred to associated 3D point clouds. Based on these properties and assumptions, three typical rigid patches that potentially exist in deformed point clouds are introduced, including small planar patches, geometric primitives, and quasi-rigid areas. Relevant segmentation techniques designed to efficiently generate these patches from point clouds are then presented, laying the groundwork for subsequent point cloud registration and deformation analysis.

For the targetless registration of deformed point clouds, correct and complete identification of stable areas is crucial for unregistered scans. To solve this challenge, point clouds are first divided into small and locally planar patches, i.e., supervoxels, and the stability of each supervoxel is analyzed by comparing to a piece-wise distance threshold. Then point-to-plane ICP is employed to register the two deformed point clouds accompanied by a progressive classification of the stable and unstable areas in an iterative way. The proposed pipeline was evaluated by two TLS point cloud datasets, showing that it outperforms other targetless registration approaches (two robust ICP variants and the voxel-based method) with respect to accuracy and robustness against complex deformations. Specifically, compared with the voxel-based method (as the state-of-the-art), the absolute registration accuracy has been improved from the millimeter level to the submillimeter level in an indoor scene and from the decimeter level to the centimeter level in a natural scene. Thereby, the first objective of this thesis is achieved: **Patch-based segmentation can improve the performance of targetless registration for deformed point clouds**.

Based on the three kinds of rigid patches, three relevant point cloud-based deformation analysis methods are presented to tackle the current challenges, including detecting small deformations under complex topographies and deriving high-accuracy and high-resolution displacement vectors in rigid areas. Specifically,

(1) PB-M3C2 employs small planar patches to construct prisms instead of fixed-size cylinders in standard M3C2, so as to separate actual surface variations and measurement uncertainties. The accuracy of calculated distances is improved while associated empirical uncertainties are reduced, thus enabling the detection of smaller deformations between complex surfaces. Specifically, compared to standard M3C2 (as the state-of-the-art), the median of uncertainties has been reduced by approx. 75% in an indoor scene with rough surfaces and 78% in a rock cliff surface, while the mean of uncertainties has been reduced by approx. 30% and 78% in the two scenes, respectively.

(2) The geometric primitive-based method uses a coarse-to-fine segmentation framework to extract multiple geometric primitives from point clouds and performs rigorous parameter estimation and associated significance tests individually. High-precision parametric deformations that fit the requirements of monitoring tasks are derived. This workflow has been successfully applied to monitor the hollow spheres in the production process of graded concrete components with submillimeter-level accuracy.

(3) Corresponding quasi-rigid areas are established across epochs by defining areal PFH descriptors for segmented patches. Then the proposed supervoxel-based registration method is locally applied between corresponding patches to derive dense 3D displacement vectors. The feasibility and accuracy of this local registration-based method are validated by simulated data with defined rigid-body movements. The absolute and relative accuracy of derived displacement vectors can achieve a submillimeter level in the simulated data if no local optimization in the local registration occurs.

Hence, the satisfactory experimental results of three proposed deformation analysis methods show that the second objective of this thesis is achieved: **Patch-based segmentation can improve the performance of point cloud-based deformation analysis**. Essentially, using segmented patches rather than individual measured points to perform deformation analysis is to explicitly or implicitly convert multiple measurements (points) into a few parameters or a model that are further used to derive deformations. For example, measurements are converted into individual patch planes in PB-M3C2, and identified rigid points are converted into the synthetic transformation matrix by local registration. This kind of local parameterization of point clouds could reduce the uncertainties of estimated deformation values, thus improving the capabilities of detecting small deformations from high-noise measurements. The rigid patches proposed in this thesis just simplify the parametrization process of real-world point clouds with complex geometries, making the methods more applicable.

Based on the three parts summarized above, it is clear to summarize the relationships and connections among patch-based segmentation (in Chapter 3), targetless registration (in Chapter 4), and deformation analysis (in Chapter 5) of TLS point clouds, which are schematically presented in Figure 6.1. Rigid patches are generated by specific segmentation techniques, which can be utilized to improve the performance of targetless registration and point cloud-based deformation analysis. Accurate registration is the prerequisite for deformation analysis, and the local registration based on rigid patches can be used for deformation analysis (deriving displacement vectors). In contrast, deformation analysis can be employed for identifying stable or unstable areas in the registration of deformed point clouds. Therefore, as the intermedium, it can be concluded that rigid patches obtained by point cloud segmentation could connect the targetless registration and deformation analysis of point clouds and enhance the performance of the two crucial parts in the processing chain of TLS-based deformation monitoring.

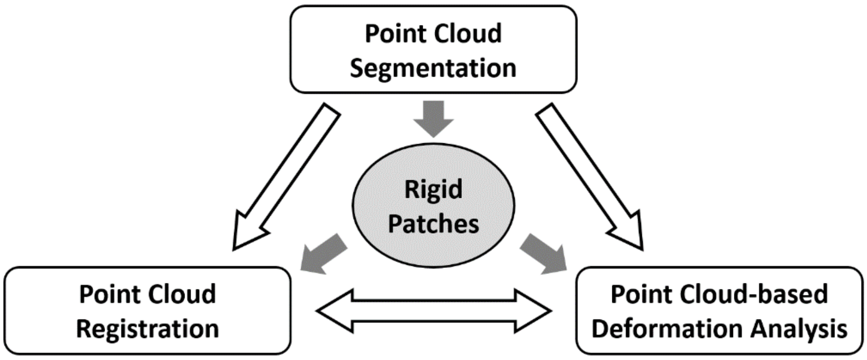


Figure 6.1: The relationship among segmentation, targetless registration, and deformation analysis of TLS point clouds.

6.2 Limitations and Outlook

This contribution has presented the improvements in targetless registration and deformation analysis of TLS point clouds with complex geometries and deformations by using patch-based segmentation, validated and evaluated by experiments with simulated and/or real data. Although these novel methods exhibit satisfactory results and even outperform the state-of-the-art in certain aspects, some issues and limitations are not fully considered in the thesis while could be investigated and tackled in the future.

Limitations of proposed targetless registration and deformation analysis methods have been specifically discussed in previous chapters (e.g., see Section 4.3, Section 5.1.3, Section 5.2.4, and Section 5.3.4). Here the common limitations, as well as potential improvements of all methods, are listed as follows.

Realistic stochastic model: In both the targetless registration method and deformation analysis methods, the stochastic model of point clouds herein is empirically estimated by considering the scanner's specifications or the residuals from local modeling, thus yielding a simple diagonal or even identity VCM without considering it. Correlations are partially considered and empirically incorporated by introducing the effective number of measurements. However, the realistic uncertainties and correlations among measured points are normally inhomogeneous and related to many error sources that are challenging to be completely considered. Hence, the realistic standard deviations may not agree with the empirical estimates, resulting in inappropriate parameter estimation and deformation analysis. Therefore, a more realistic stochastic model with a fully populated VCM of the point cloud can be obtained by, e.g., the elementary error model (Kauker and Schwieger, 2017; Kerekes and Schwieger, 2020; Kerekes and Schwieger, 2021) and integrated into the local modeling of rigid patches. Exemplary applications of fully populated VCM in parameterization and deformation analysis are presented by Raschhofer *et al.* (2021) and Kerekes *et al.* (2022). Besides, other empirical stochastic models (Wujanz *et al.*, 2017; Schmitz *et al.*, 2021) can be utilized to improve the modeling accuracy of patches as well.

Imperfect patch-based segmentation: Due to complex surface geometries, high measurement noise, and algorithmic imperfection, rigid patches may not be perfectly segmented. For example, boundaries are not completely separated in generated supervoxels, or outliers are still contained in segmented geometric primitives. In these cases, systematic errors may arise in the estimated parameters based on these imperfect rigid patches, leading to incorrect deformation analysis. Two strategies can be adopted to avoid these cases. The first strategy is performing strict validations on segmented patches regarding, e.g., their planarity (for small planar patches), inlier ratio (for geometric primitives), and rigid ratio (for quasi-rigid areas). Unqualified patches will not be involved in the following registration or deformation analysis. The second strategy is to employ better-performing segmentation techniques, especially deep learning-based methods when annotated training dataset incorporating ground truth is available (Landrieu and Boussaha, 2019; Wang *et al.*, 2018; Zhang *et al.*, 2020).

Absence of the radiometric data: The methods developed in this thesis are solely based on the geometric information (i.e., 3D coordinates) of point clouds. In spite of their generality and applicability on ubiquitous point cloud data captured from different sensors or platforms, as mentioned in Gojcic (2021), correspondences of rigid patches between epochs may not always be correctly established merely relying on geometric properties. For instance, correspondences might be ambiguous in the case of repetitive structures. Therefore, radiometric information like RGB colors or reflectance intensity along with the point coordinates can be exploited to help enhance the distinction (i.e., the descriptiveness of features) of associated segmented patches (Gojcic, 2021). On the one hand, measured points with short spatial distances as well as similar RGB colors or reflectance values may indicate their consistency regarding the type of materials of object surfaces. On the other hand, a rigid local area basically belongs to the same material, especially in natural scenes like boulders in a landslide. Hence, radiometric data can also be utilized in patch generation to increase the ratio of rigid points. In particular, integrating radiometric values with specified weight in Equation (3.1) has the potential to improve the quality of generated supervoxels. Moreover, by taking both geometric and radiometric information into account for instance segmentation, more complete and cleaner geometric primitives or individual objects can be extracted from complex scenarios.

Since most experimental evaluations of the proposed methods are conducted on simulated datasets or the point cloud data with simulated deformations for the purpose of comparison with known ground truth (reference), practical applications that lack a known ground truth are yet to be demonstrated. Nevertheless, it should be emphasized that the methods and algorithms presented in this dissertation hold great potential for real-world applications in areal geodetic monitoring, including natural hazard scenarios such as landslides and rockfalls, as well as civil constructions and industrial facilities like buildings and dams.

Bibliography

Abellán A, Vilaplana JM, Calvet J, García-Sellés D, Asensio E. Rockfall monitoring by terrestrial laser scanning – Case study of the basaltic rock face at Castellfollit de la Roca (Catalonia, Spain). *Natural Hazards and Earth System Sciences*. **2011**; 11(3):829–841.

Abuzaina A, Nixon MS, Carter JN. Sphere detection in Kinect point clouds via the 3D Hough transform. In: *International Conference on Computer Analysis of Images and Patterns*. **2013**; York, UK, pp. 290–297.

Adams R, Bischof L. Seeded region growing. *IEEE Transactions on Pattern Analysis and Machine Intelligence.* **1994**; 16(6):641–647.

Aiger D, Mitra NJ, Cohen-Or D. 4-points congruent sets for robust pairwise surface registration. *ACM Transactions on Graphics*. **2008**; 27(3):1–10.

Anders K, Lindenbergh RC, Vos SE, Mara H, De Vries S, Höfle B. High-frequency 3D geomorphic observation using hourly terrestrial laser scanning data of a sandy beach. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*. **2019**; 4(2/W5):317–324.

Aspert N, Santa-Cruz D, Ebrahimi T. Mesh: Measuring errors between surfaces using the Hausdorff distance. In: *Proceedings of the IEEE International Conference on Multimedia and Expo*. **2002**; 1:705–708.

Attene M, Patanè G. Hierarchical structure recovery of point-sampled surfaces. In: *Computer Graphics Forum*. **2010**; 29:1905–1920.

Balangé L, Zhang L, Schwieger V. First step towards the technical quality concept for integrative computational design and construction. In: *International Conferences on Engineering Surveying*. **2021**; pp. 118–127.

Balta H, Velagic J, Bosschaerts W, De Cubber G, Siciliano B. Fast statistical outlier removal based method for large 3D point clouds of outdoor environments. *IFAC-PapersOnLine*. **2018**; 51(22):348–353.

Barbarella M, Fiani M, Lugli A. Landslide monitoring using multitemporal terrestrial laser scanning for ground displacement analysis. *Geomatics, Natural Hazards and Risk*. **2015**; 6:398–418.

Belton D, Lichti DD. Classification and segmentation of terrestrial laser scanner point clouds using local variance information. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*. **2006**; 36(5):44–49.

Besl PJ, McKay ND. A method for registration of 3-D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. **1992**; 14(2):239–256.

Blandini L, Sobek W. Das Institut für Leichtbau Entwerfen und Konstruieren ILEK: Über 100 Jahre Forschung im Stahlbetonbau. *Beton-und Stahlbetonbau*. **2020**; 115(8):626–631.

Burger M. Synergetic segmentation and registration of TLS point clouds using geometric and radiometric information. *Ph.D. Thesis*. **2022**; Berlin, Germany: Technische Universität Berlin.

Bustos AP, Chin TJ. Guaranteed outlier removal for point cloud registration with correspondences. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. **2017**; 40(12):2868–2882.

Camurri M, Vezzani R, Cucchiara R. 3D Hough transform for sphere recognition on point clouds. *Machine Vision and Applications*. **2014**; 25(7):1877–1891.

Chen S, Nan L, Xia R, Zhao J, Wonka P. PLADE: A plane-based descriptor for point cloud registration with small overlap. *IEEE Transactions on Geoscience and Remote Sensing*. **2020**; 58(4):2530–2540.

Cheng L, Chen S, Liu X, Xu H, Wu Y, Li M, Chen Y. Registration of laser scanning point clouds: A review. *Sensors*. **2018**; 18(5):1641.

Chetverikov D, Stepanov D, Krsek P. Robust Euclidean alignment of 3D point sets: the trimmed iterative closest point algorithm. *Image and Vision Computing*. **2005**; 23(3):299–309.

Choi S, Kim T, Yu W. Performance evaluation of RANSAC family. In: *Proceedings of the British Machine Vision Conference (BMVC)*. **2009**; pp. 1–12.

Cignoni P, Rocchini C, Scopigno R. Metro: measuring error on simplified surfaces. In: *Computer Graphics Forum*. **1998**; 17(2):167–174.

Connolly C. Cumulative generation of octree models from range data. In: *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*. **1984**; pp. 25–32.

DiFrancesco PM, Bonneau D, Hutchinson DJ. The implications of M3C2 projection diameter on 3D semi-automated rockfall extraction from sequential terrestrial laser scanning point clouds. *Remote Sensing*. **2020**; 12(11):1885.

Dyke RM, Lai YK, Rosin PL, Tam GK. Non-rigid registration under anisotropic deformations. *Computer Aided Geometric Design*. **2019**; 71:142–56.

Eitel JU, Höfle B, Vierling LA, Abellán A, Asner GP, Deems JS, Glennie CL, Joerg PC, LeWinter AL, Magney TS, Mandlburger G. Beyond 3-D: The new spectrum of LiDAR applications for earth and ecological sciences. *Remote Sensing of Environment*. **2016**; 186:372–392.

Ester M, Kriegel HP, Sander J, Xu X. A density-based algorithm for discovering clusters in large spatial databases with noise. In: *Kdd*. **1996**; 96(34):226–231.

Fan L, Smethurst JA, Atkinson PM, Powrie W. Error in target-based georeferencing and registration in terrestrial laser scanning. *Computers & Geosciences*. **2015**; 83:54–64.

Fischler MA, Bolles RC. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*. **1981**; 24(6):381–395.

Friedli E, Wieser A. Identification of stable surfaces within point clouds for areal deformation monitoring. In: *Proceedings of 3rd Joint International Symposium on Deformation Monitoring (JISDM)*. **2016**; Vienna, Austria.

Friedli E. Point cloud registration and mitigation of refraction effects for geomonitoring using long-range terrestrial laser scanning. *Ph.D. Thesis*. **2020**; Switzerland: ETH Zurich.

Fu W, Zhang H. Reverse engineering design based on Geomagic Studio software. *Tool Engineering*. **2007**; 41:54–57.

Girardeau-Montaut D, Roux M, Marc R, Thibault G. Change detection on points cloud data acquired with a ground laser scanner. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences.* **2005**; 36:30–35.

Glira P, Briese C, Kamp N, Pfeifer N. Simultaneous relative and absolute orientation of point clouds with" TLS radomes". In: E*GU General Assembly Conference Abstracts*. **2013**; pp. 13116.

Gojcic Z, Schmid L, Wieser A. Dense 3D displacement vector fields for point cloud-based landslide monitoring. *Landslides*. **2021**; 18(12):3821–3832.

Gojcic Z, Zhou C, Wegner JD, Guibas LJ, Birdal T. Learning multiview 3D point cloud registration. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. **2020**a; Virtual.

Gojcic Z, Zhou C, Wieser A. F2S3: Robustified determination of 3D displacement vector fields using deep learning. *Journal of Applied Geodesy*. **2020**b; 14(2):177–189.

Gojcic Z. Benefiting from local rigidity in 3D point cloud processing. *Ph.D. Thesis*. **2021**; Switzerland: ETH Zurich.

Gruen A, Akca D. Least squares 3D surface and curve matching. I*SPRS Journal of Photogrammetry and Remote Sensing*. **2005**; 59(3):151–174.

Grussenmeyer P, Landes T, Voegtle T, Ringle K. Comparison methods of terrestrial laser scanning, photogrammetry and tacheometry data for recording of cultural heritage buildings. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*. **2008**; 37:213–218.

Han XF, Jin JS, Wang MJ, Jiang W, Gao L, Xiao L. A review of algorithms for filtering the 3D point cloud. *Signal Processing: Image Communication*. **2017**; 57:103–112.

Harmening C, Hobmaier C, Neuner H. Laser Scanner–Based Deformation Analysis Using Approximating B-Spline Surfaces. *Remote Sensing*. **2021**; 13(18):3551.

Harmening C, Neuner H. A constraint-based parameterization technique for B-spline surfaces. *Journal of Applied Geodesy*. **2015**; 9(3):143–161.

Harmening C. Spatio-temporal deformation analysis using enhanced B-spline models of laser scanning point clouds. *Ph.D. Thesis*. **2020**; Vienna, Austria: Technische Universität Wien.

Hassan A, Zhang L, Kerekes G, Schwieger V. Fusion of inhomogeneous geodetic data for rock cliff monitoring: A case study of the Lianziya cliff in Three Gorges National Geological Park in China. In *Proceedings of the XXVII FIG Congress*. **2022**; pp. 11340.

Heunecke O, Kuhlmann H, Welsch W. Auswertung geodätischer Überwachungsmessungen. Handbuch Ingenieurgeodäsie. **2013**; Wichmann, Berlin, 2nd edition.

Himmelsbach M, Hundelshausen FV, Wuensche HJ. Fast segmentation of 3D point clouds for ground vehicles. In: *IEEE Intelligent Vehicles Symposium*. **2010**; pp. 560–565.

Holst C, Janßen J, Schmitz B, Blome M, Dercks M, Schoch-Baumann A, Blöthe J, Schrott L, Kuhlmann H, Medic T. Increasing spatio-temporal resolution for monitoring alpine solifluction using terrestrial laser scanners and 3D vector fields. *Remote Sensing*. **2021**; 13(6):1192.

Holst C, Kuhlmann H. Challenges and present fields of action at laser scanner based deformation analyses. *Journal of Applied Geodesy*. **2016**; 10(1):17–25.

Holst C, Nothnagel A, Haas R, Kuhlmann H. Investigating the gravitational stability of a radio telescope's reference point using a terrestrial laser scanner: Case study at the Onsala Space Observatory 20-m radio telescope. *ISPRS Journal of Photogrammetry and Remote Sensing*. **2019**; 149:67–76.

Holz D, Ichim AE, Tombari F, Rusu RB, Behnke S. Registration with the point cloud library: A modular framework for aligning in 3-D. *IEEE Robotics & Automation Magazine*. **2015**; 22(4):110–124.

Honti R, Erdélyi J, Kopáčik A. Semi-automated segmentation of geometric shapes from point clouds. *Remote Sensing*. **2022**; 14(18):4591.

Huang QX, Adams B, Wicke M, Guibas LJ. Non-rigid registration under isometric deformations. *Computer Graphics Forum*. **2008**; 27(5):1449–1457.

Huang X, Li S, Zuo Y, Fang Y, Zhang J, Zhao X. Unsupervised point cloud registration by learning unified Gaussian mixture models. *IEEE Robotics and Automation Letters*. **2022**; 7(3):7028–7035.

Hui L, Yuan J, Cheng M, Xie J, Zhang X, Yang J. Superpoint network for point cloud oversegmentation. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. **2021**; pp. 5510–5519.

Jagannathan A, Miller EL. Three-dimensional surface mesh segmentation using curvedness-based region growing approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. **2007**; 29(12):2195–2204.

Janßen J, Kuhlmann H, Holst C. Target-based terrestrial laser scan registration extended by target orientation. *Journal of Applied Geodesy*. **2022**; 16(2):91–106.

Janßen J, Medic T, Kuhlmann H, Holst C. Decreasing the uncertainty of the target center estimation at terrestrial laser scanning by choosing the best algorithm and by improving the target design. *Remote Sensing*. **2019**; 11(7):845.

Kabsch W. A solution for the best rotation to relate two sets of vectors. *Acta Crystallographica Section A: Crystal Physics, Diffraction, Theoretical and General Crystallography*. **1976**; 32(5): 922–923.

Kauker S, Holst C, Schwieger V, Kuhlmann H, Schön S. Spatio-temporal correlations of terrestrial laser scanning. *Allgemeine Vermessungs-Nachrichten: AVN*. **2016**; 123(6):170–182.

Kauker S, Schwieger V. A synthetic covariance matrix for monitoring by terrestrial laser scanning. *Journal of Applied Geodesy*. **2017**; 11(2):77–87.

Kazhdan M, Bolitho M, Hoppe H. Poisson surface reconstruction. In: *Proceedings of the 4th Eurographics Symposium on Geometry Processing*. **2006**; 7.

Kerekes G, Raschhofer J, Harmening C, Schwieger V, Neuner H. Two-epoch TLS deformation analysis of a double curved wooden structure using B-spline surfaces. In: *Proceedings of the 5th Joint International Symposium on Deformation Monitoring (JISDM)*. **2022**; Valencia, Spain.

Kerekes G, Schwieger V. Elementary error model applied to terrestrial laser scanning measurements: Study case arch dam kops. *Mathematics*. **2020**; 8(4):593.

Kerekes G, Schwieger V. Determining variance-covariance matrices for terrestrial laser scans: a case study of the arch dam kops. In: *International Conferences on Engineering Surveying*. **2021**; pp. 57–68.

Kermarrec G, Kargoll B, Alkhatib H. Deformation analysis using b-spline surface with correlated terrestrial laser scanner observations—A bridge under load. *Remote Sensing*. **2020**; 12(5):829.

Klasing K, Althoff D, Wollherr D, Buss M. Comparison of surface normal estimation methods for range sensing applications. In: *IEEE International Conference on Robotics and Automation (ICRA)*. **2009**; Kobe, Japan, pp. 3206–3211.

Klingbeil L, Eling C, Heinz E, Wieland M, Kuhlmann H. Direct georeferencing for portable mapping systems: In the air and on the ground. *Journal of Surveying Engineering*. **2017**; 143(4):04017010.

Kovaleva D, Gericke O, Kappes J, Tomovic I, Sobek W. Rosenstein Pavilion: Design and structural analysis of a functionally graded concrete shell. *Structures*. **2019**; 18:91–101.

Kuhlmann H, Schwieger V, Wieser A, Niemeier W. Engineering geodesy – Definition and core competencies. *Journal of Applied Geodesy*. **2014**; 8(4):327–334.

Lague D, Brodu N, Leroux J. Accurate 3D comparison of complex topography with terrestrial laser scanner: Application to the Rangitikei canyon (NZ). *ISPRS Journal of Photogrammetry and Remote Sensing*. **2013**; 82:10–26.

Landrieu L, Boussaha M. Point cloud oversegmentation with graph-structured deep metric learning. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. **2019**; pp. 7440–7449.

Lane SN, Westaway RM, Murray Hicks D. Estimation of erosion and deposition volumes in a large, gravel-bed, braided river using synoptic remote sensing. *Earth Surface Processes And Landforms*. **2003**; 28(3):249–271.

Lavoué G, Dupont F, Baskurt A. A new CAD mesh segmentation method, based on curvature tensor analysis. *Computer-Aided Design*. **2005**; 37(10):975–987.

Lenzmann L, Lenzmann E. Strenge Auswertung des nichtlinearen Gauß-Helmert-Modells. *Allgemeine Vermessungs Nachrichten (AVN)*. **2004**; 111:68–73.

Li J, Ayiguli A, Zheng D. Automatically identifying uncertain spherical targets in 3D point clouds of complex scenario. *Journal of Computer Aided Design & Computer Graphics*. **2013**; 25(10):1489–1495.

Li L, Sung M, Dubrovina A, Yi L, Guibas LJ. Supervised fitting of geometric primitives to 3D point clouds. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. **2019**; pp. 2652–2660.

Li ZW, Yang ZF, Zhu JJ, Hu J, Wang YJ, Li PX, Chen GL. Retrieving three-dimensional displacement fields of mining areas from a single InSAR pair. *Journal of Geodesy*. **2015**; 89(1):17–32.

Lin Y, Wang C, Zhai D, Li W, Li J. Toward better boundary preserved supervoxel segmentation for 3D point clouds. *ISPRS Journal of Photogrammetry and Remote Sensing*. **2018**; 143:39–47.

Lindenbergh R, Pfeifer N, Rabbani T. Accuracy analysis of the Leica HDS3000 and feasibility of tunnel deformation monitoring. In: *Proceedings of the ISPRS Workshop on Laser scanning*. **2005**; Enschede, The Netherlands.

Low KL. Linear least-squares optimization for point-to-plane ICP surface registration. *Chapel Hill, University of North Carolina*. **2004**; 4(10):1–3.

Luo MR. CIELAB. In: Luo, R. (eds) Encyclopedia of Color Science and Technology. Springer, Berlin, Heidelberg. **2015**. https://doi.org/10.1007/978-3-642-27851-8_11-1

Medic T, Ruttner P, Holst C, Wieser A. Keypoint-based deformation monitoring using a terrestrial laser scanner from a single station: Case study of a bridge pier. In: *Proceedings of the 5th Joint International Symposium on Deformation Monitoring (JISDM)*. **2022**; Valencia, Spain.

Meng X, Wang H, Liu B. A robust vehicle localization approach based on gnss/imu/dmi/lidar sensor fusion for autonomous vehicles. *Sensors*. **2017**; 17(9):2140.

Moore JF. Monitoring building structures. 1st Edition. Glasgow and London: Blackie; **1992**.

Mukupa W, Roberts GW, Hancock CM, Al-Manasir K. A review of the use of terrestrial laser scanning application for change detection and deformation monitoring of structures. *Survey Review*. **2017**; 49(353):99–116.

Neuner H, Holst C, Kuhlmann H. Overview on current modelling strategies of point clouds for deformation analysis. *Allgemeine Vermessungs-Nachrichten: AVN*. **2016**; 123:328–339.

Niemeier W. Ausgleichungsrechnung: Statistische Auswertemethoden. *Walter de Gruyter*. **2008**.

Ning X, Li F, Tian G, Wang Y. An efficient outlier removal method for scattered point cloud data. *PloS One*. **2018**; 13(8):e0201280.

Nurunnabi A, Belton D, West G. Robust segmentation in laser scanning 3D point cloud data. In: *International Conference on Digital Image Computing Techniques and Applications (DICTA)*. **2012**; Fremantle, Australia, pp. 1–8.

Ohlmann-Lauber J, Schäfer T. Ansätze zur Ableitung von Deformationen aus TLS-Daten. In: *Schriftenreihe DVW, 66 (Terrestrisches Laserscanning)*. **2011**; Wißner, Augsburg, Germany, pp. 147–157.

Oppikofer T, Jaboyedoff M, Blikra L, Derron MH, Metzger R. Characterization and monitoring of the Åknes rockslide using terrestrial laser scanning. *Natural Hazards and Earth System Sciences*. **2009**; 9(3):1003–1019.

Paffenholz JA. Direct geo-referencing of 3D point clouds with 3D positioning sensors. *Ph.D. Thesis*. **2012**; Hannover, Germany: Gottfried Wilhelm Leibniz Universität Hannover.

Pandžić J, Pejić M, Božić B, Erić V. Error model of direct georeferencing procedure of terrestrial laser scanning. *Automation in Construction*. **2017**; 78:13–23.

Papon J, Abramov A, Schoeler M, Worgotter F. Voxel cloud connectivity segmentation-supervoxels for point clouds. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. **2013**; Portland, Oregon, USA.

Park SY, Subbarao M. An accurate and fast point-to-plane registration technique. *Pattern Recognition Letters*. **2003**; 24(16):2967–2976.

Pauly M, Gross M, Kobbelt LP. Efficient simplification of point-sampled surfaces. In: *Proceedings of IEEE Visualization*. **2002**; Boston, MA, USA, pp. 163–170.

Pfeiffer J, Wujanz D, Zieher T, Rutzinger M, Scaioni M, Höfle B, Lindenbergh R, Oude Elberink S, Pirotti F, Bremer M, Hämmerle M. Terrestrial laser scanning data of the Nesslrinna landslide close to Obergurgl, Austria acquired during the 2019 Innsbruck Summer School of Alpine Research. *University of Innsbruck, PANGAEA*. **2019**; Available from: https://doi.org/10.1594/PANGAEA.901293 [accessed on 31 January 2023].

Pfeiffer J, Zieher T, Bremer M, Wichmann V, Rutzinger M. Derivation of three-dimensional displacement vectors from multi-temporal long-range terrestrial laser scanning at the Reissenschuh landslide (Tyrol, Austria). *Remote Sensing*. **2018**; 10(11):1688.

Poiesi F, Boscaini D. Distinctive 3D local deep descriptors. In: *25th International Conference on Pattern Recognition (ICPR)*. **2021**; Milan, Italy.

Qin H, Guan H, Yu Y, Zhong L. A voxel-based filtering algorithm for mobile lidar data. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences.* **2018**; 42(3):1433–1438.

Qin R, Tian J, Reinartz P. 3D change detection–approaches and applications. *ISPRS Journal of Photogrammetry and Remote Sensing*. **2016**; 122:41–56.

Rabbani T, Van Den Heuvel F, Vosselmann G. Segmentation of point clouds using smoothness constraint. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*. **2006**; 36(5):248–253.

Rachakonda P, Muralikrishnan B, Cournoyer L, Cheok G, Lee V, Shilling M, Sawyer D. Methods and considerations to determine sphere center from terrestrial laser scanner point cloud data. *Measurement Science and Technology*. **2017**; 28(10):105001.

Raffl L, Holst C. Including virtual target points from laser scanning into the point-wise rigorous deformation analysis at geo-monitoring applications. In: *Proceedings of the 5th Joint International Symposium on Deformation Monitoring (JISDM)*. **2022**; Valencia, Spain.

Raguram R, Chum O, Pollefeys M, Matas J, Frahm JM. USAC: A universal framework for random sample consensus. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. **2012**; 35(8):2022–2038.

Raschhofer J, Kerekes G, Harmening C, Neuner H, Schwieger V. Estimating control points for B-Spline surfaces using fully populated synthetic variance–covariance matrices for TLS point clouds. *Remote Sensing*. **2021**; 13(16):3124.

Rashidi M, Mohammadi M, Sadeghlou Kivi S, Abdolvand MM, Truong-Hong L, Samali B. A decade of modern bridge monitoring using terrestrial laser scanning: Review and future directions. *Remote Sensing*. **2020**; 12(22):3796.

Rusu RB, Blodow N, Beetz M. Fast point feature histograms (FPFH) for 3D registration. In: *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*. **2009**; Kobe, Japan.

Rusu RB, Marton ZC, Blodow N, Dolha M, Beetz M. Towards 3D point cloud based object maps for household environments. *Robotics and Autonomous Systems*. **2008**; 56(11):927–941.

Rusu RB. Semantic 3D object maps for everyday manipulation in human living environments. *Ph.D. Thesis*. **2009**; Munich, Germany: Technische Universität Müchen.

Scaioni M. Direct georeferencing of TLS in surveying of complex sites. In: *Proceedings of the ISPRS Working Group*. **2005**; 4:22–24.

Schmeer D, Sobek W. Gradientenbeton. In: *Beton Kalender 2019: Parkbauten Geotechnik und Eurocode 7*. **2019**; 108:456–476.

Schmitz B, Kuhlmann H, Holst C. Investigating the resolution capability of terrestrial laser scanners and its impact on the effective number of measurements. *ISPRS Journal of Photogrammetry and Remote Sensing*. **2020**; 159:41–52.

Schnabel R, Wahl R, Klein R. Efficient RANSAC for point-cloud shape detection. *Computer Graphics Forum*. **2007**; 26:214–226.

Schweitzer J, Schwieger V. Modeling of quality for engineering geodesy processes in civil engineering. *Journal of Applied Geodesy*. **2011**; 5(1):13–22.

Schäfer T, Weber T, Kyrinovic P, Zámecniková M. Deformation measurement using terrestrial laser scanning at the hydropower station of Gabcikovo. In: *INGEO 2004 and FIG Regional Central and Eastern European Conference on Engineering Surveying*. **2004**; Bratislava, Slovakia.

Segal A, Haehnel D, Thrun S. Generalized-ICP. *In Robotics: Science and Systems*. **2009**; 2(4):435.

Shan J, Toth CK. Topographic laser ranging and scanning: principles and processing. CRC press, 2nd edition, **2018**.

Sobek W. Die Zukunft des Leichtbaus: Herausforderungen und mögliche Entwicklungen. *Bautechnik*. **2015**; 92(12):879–882.

Sobek W. Über die Gestaltung der Bauteilinnenräume. Festschrift zu Ehren von Prof. Dr.-Ing. Dr.-Ing. Eh Manfred Curbach, Dresden: Institut für Massivbau der TU Dresden. **2016**; pp. 62–76.

Stilla U, Xu Y. Change detection of urban objects using 3D point clouds: A review. *ISPRS Journal of Photogrammetry and Remote Sensing*. **2023**; 197:228–255.

Suchocki C, Błaszczak-Bąk W. Down-sampling of point clouds for the technical diagnostics of buildings and structures. *Geosciences*. **2019**; 9(2):70.

Taubenheim J. Statistische Auswertung geophysikalischer und meteorologischer Daten. Leipzig, Germany: *Akad. Verlagsgesellschaft Geest & Portig K. G*. **1969**.

Teza G, Galgaro A, Zaltron N, Genevois R. Terrestrial laser scanner to detect landslide displacement fields: a new approach. *International Journal of Remote Sensing*. **2007**; 28(16):3425–3446.

Theiler PW, Wegner JD, Schindler K. Keypoint-based 4-points congruent sets–automated marker-less registration of laser scans. *ISPRS Journal of Photogrammetry and Remote Sensing*. **2014**; 96:149–163.

Tournas E, Tsakiri M. Deformation monitoring based on terrestrial laser scanner point cloud registration. In: *13th FIG Symposium on Deformation Measurement and Analysis*. **2008**; LNEC Lisbon.

Tran TT, Cao VT, Laurendeau D. eSphere: extracting spheres from unorganized point clouds. *The Visual Computer*. **2016**; 32(10):1205–1222.

Travelletti J, Malet JP, Delacourt C. Image-based correlation of laser scanning point cloud time series for landslide monitoring. *International Journal of Applied Earth Observation and Geoinformation*. **2014**; 32:1–8.

Vieira M, Shimada K. Surface mesh segmentation and smooth surface extraction through region growing. *Computer Aided Geometric Design*. **2005**; 22(8):771–792.

Vo AV, Truong-Hong L, Laefer DF, Bertolotto M. Octree-based region growing for point cloud segmentation. *ISPRS Journal of Photogrammetry and Remote Sensing*. **2015**; 104:88–100.

Vosselman G, Maas H.-G. Airborne and terrestrial laser scanning. Boca Raton, FL, USA: CRC press. **2010**.

Wagner A, Wiedemann W, Wunderlich T. Fusion of laser-scan and image data for deformation monitoring–concept and perspective. In: *Proceedings of the 7th International Conference on Engineering Surveying (INGEO)*. **2017**; pp. 157–164.

Walicka A, Pfeifer N. Automatic segmentation of individual grains from a terrestrial laser scanning point cloud of a mountain river bed. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*. **2022**; 15:1389–1410.

Wang H, Guan X, Wu H. A hybrid parallel spatial interpolation algorithm for massive LiDAR point clouds on heterogeneous CPU-GPU systems. *ISPRS International Journal of Geo-Information*. **2017**; 6(11):363.

Wang L, Bao J. Initial registration algorithm for PCA point cloud data with additional main direction decision. *Journal of Geomatics*. **2021**; 46(6):59–62,66.

Wang W, Yu R, Huang Q, Neumann U. SGPN: Similarity group proposal network for 3D point cloud instance segmentation. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. **2018**; pp. 2569–2578.

Wang Y, Feng HY. Effects of scanning orientation on outlier formation in 3D laser scanning of reflective surfaces. *Optics and Lasers in Engineering*. **2016**; 81:35–45.

Wang Y, Shi H, Zhang Y, Zhang D. Automatic registration of laser point cloud using precisely located sphere targets. *Journal of Applied Remote Sensing*. **2014**; 8(1):083588.

Wang Y, Wang J, Chen X, Chu T, Liu M, Yang T. Feature surface extraction and reconstruction from industrial components using multistep segmentation and optimization. *Remote Sensing*. **2018**; 10(7):1073.

Watson GA. Computing Helmert transformations. *Journal of Computational and Applied Mathematics*. **2006**; 197(2):387–394.

Wei F. Reverse engineering design based on Geomagic Studio software. *Tool Engineering*. **2007**; 41(11):54–57.

Wei S. Building boundary extraction based on LiDAR point clouds data. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences.* **2008**; 37:157–161.

Weidner L, van Veen M, Lato M, Walton G. An algorithm for measuring landslide deformation in terrestrial lidar point clouds using trees. *Landslides*. **2021**; 18:3547–3558.

Weinmann M, Jutzi B, Hinz S, Mallet C. Semantic point cloud interpretation based on optimal neighborhoods, relevant features and efficient classifiers. *ISPRS Journal of Photogrammetry and Remote Sensing*. **2015**; 105:286–304.

Williams JG, Anders K, Winiwarter L, Zahs V, Höfle B. Multi-directional change detection between point clouds. *ISPRS Journal of Photogrammetry and Remote Sensing*. **2021**; 172:95–113.

Winiwarter L, Anders K, Höfle B. M3C2-EP: Pushing the limits of 3D topographic point cloud change detection by error propagation. *ISPRS Journal of Photogrammetry and Remote Sensing*. **2021**; 178:240–258.

Wong KY, Zhang G, Chen Z. A stratified approach for camera calibration using spheres. *IEEE Transactions on Image Processing*. **2011**; 20(2):305–316.

Wörner M, Schmeer D, Schuler B, Pfinder J, Garrecht H, Sawodny O, Sobek W. Gradientenbetontechnologie: von der Mischungsentwicklung über den Bauteilentwurf bis zur automatisierten Herstellung. *Beton- und Stahlbetonbau*. **2016**; 111(12):794–805.

Wujanz D, Burger M, Mettenleiter M, Neitzel F. An intensity-based stochastic model for terrestrial laser scanners. *ISPRS Journal of Photogrammetry and Remote Sensing*. **2017**; 125:146–155.

Wujanz D, Krueger D, Neitzel F. Identification of stable areas in unreferenced laser scans for deformation measurement. *The Photogrammetric Record*. **2016**; 31(155):261–280.

Wujanz D. Terrestrial laser scanning for geodetic deformation monitoring. *Ph.D. Thesis*. **2016**; Berlin, Germany: Technische Universität Berlin.

Wunderlich T, Niemeier W, Wujanz D, Holst C, Neitzel F, Kuhlmann H. Areal deformation analysis from TLS point clouds—The challenge. *Allgemeine Vermessungs Nachrichten (AVN)*. **2016**; 123:340–351.

Xia S, Chen D, Wang R, Li J, Zhang X. Geometric primitives in LiDAR point clouds: A review. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*. **2020**; 13:685–707.

Xiao Y, Chen Z, Lin Z, Cao J, Zhang YJ, Lin Y, Wang C. Merge-swap optimization framework for supervoxel generation from three-dimensional point clouds. *Remote Sensing*. **2020**; 12(3):473.

Xie Y, Tian J, Zhu XX. Linking points with labels in 3D: A review of point cloud semantic segmentation. *IEEE Geoscience and Remote Sensing Magazine*. **2020**; 8(4):38–59.

Xiong L, Wang G, Bao Y, Zhou X, Sun X, Zhao R. Detectability of repeated airborne laser scanning for mountain landslide monitoring. *Geosciences*. **2018**; 8(12):469.

Xu Y, Tuttas S, Hoegner L, Stilla U. Geometric primitive extraction from point clouds of construction sites using VGS. *IEEE Geoscience and Remote Sensing Letters*. **2017**; 14(3):424–428.

Yan L, Wei P, Xie H, Dai J, Wu H, Huang M. A new outlier removal strategy based on reliability of correspondence graph for fast point cloud registration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. **2023**; 45(7):7986–8002.

Yang B, Dong Z, Liang F, Liu Y. Automatic registration of large-scale urban scene point clouds based on semantic feature points. *ISPRS Journal of Photogrammetry and Remote Sensing*. **2016**a; 113:43–58.

Yang B, Zang Y, Dong Z, Huang R. An automated method to register airborne and terrestrial laser scanning point clouds. *ISPRS Journal of Photogrammetry and Remote Sensing*. **2015**; 109:62–76.

Yang J, Cao Z, Zhang Q. A fast and robust local descriptor for 3D point cloud registration. *Information Sciences*. **2016**b; 346:163–179.

Yang Y, Balangé L, Gericke O, Schmeer D, Zhang L, Sobek W, Schwieger V. Monitoring of the production process of graded concrete component using terrestrial laser scanning. *Remote Sensing*. **2021**; 13(9):1622.

Yang Y, Schwieger V. Patch-based M3C2: Towards lower-uncertainty and higher-resolution deformation analysis of 3D point clouds. *International Journal of Applied Earth Observation and Geoinformation*. **2023**; 125:103535.

Yang Y, Schwieger V. Supervoxel-based targetless registration and identification of stable areas for deformed point clouds. *Journal of Applied Geodesy*. **2022**; 17(2):161–170.

Zahs V, Winiwarter L, Anders K, Williams JG, Rutzinger M, Höfle B. Correspondence-driven plane-based M3C2 for lower uncertainty in 3D topographic change quantification. *ISPRS Journal of Photogrammetry and Remote Sensing*. **2022**; 183:541–559.

Zhang F, Guan C, Fang J, Bai S, Yang R, Torr PH, Prisacariu V. Instance segmentation of LiDAR point clouds. In: *IEEE International Conference on Robotics and Automation (ICRA)*. **2020**; pp. 9448–9455.

Zhao X, Kargoll B, Omidalizarandi M, Xu X, Alkhatib H. Model selection for parametric surfaces approximating 3D point clouds for deformation analysis. *Remote Sensing*. **2018**; 10(4):634.

Zhong W, Zhang T, Chen J, Shang J, Wang S, Mu C, Fan C. Seasonal deformation monitoring over thermokarst landforms using terrestrial laser scanning in Northeastern Qinghai-Tibetan Plateau. *International Journal of Applied Earth Observation and Geoinformation*. **2021**; 103:102501.

Appendix

A.1 3D Region Growing

The region growing algorithm was first proposed for the intensity image segmentation by Adams and Bischof (1994). The inputs of this region-based algorithm are the intensity image and a set of seeds which are individual points or connected components. By gradually expanding the range of seed points provided a small difference in gray values, the algorithm can identify and cluster individual regions to be segmented in the image.

The region growing algorithm introduced to 3D point cloud segmentation starts from one or more points (i.e., seed points) featuring specific characteristics and then grows around neighboring points with similar characteristics, such as surface orientation, curvature, etc. (Grilli *et al.*, 2017; Jagannathan and Miller 2007). The constraints of surface normal vectors and curvatures were widely used to find the smoothly connected areas that would be clustered as specific regions (Rabbani *et al.*, 2006; Nurunnabi *et al.*, 2012; Belton and Lichti, 2006; Klasing *et al.*, 2009).

The general steps of standard 3D region growing algorithm for point cloud segmentation can be summarized as follows:

(1) The unlabeled points in the point cloud are sorted according to their curvatures, and the point with minimum curvature will be added into the seed point set;
(2) The set of neighbor points of each seed point is obtained by *k*-nearest search based on a *k*d-tree structure;
(3) Points that pass the following two tests will be removed from the original point cloud:

(i) The angles between normal vectors of each neighboring point and the current seed point are calculated. If the difference is less than the angle threshold, the neighbor point will be clustered into the current region and further tested in step (ii), or this point will not be considered;
(ii) If the curvature of the point that has passed step (i) is less than the curvature threshold, the point will be clustered and added to the seed point set as well, or it will only be clustered.

(4) If the number of some clustered point set is between the threshold of minimum and maximum point number, this cluster will be segmented and labeled as one of the final results;
(5) Repeat steps (1) to (4). The algorithm stops until the point clusters generated in the remaining points cannot meet the threshold of minimum number of points.

The detailed process of standard 3D region growing algorithm can be described in pseudocode in **Algorithm 1** (Rabbani *et al.* 2006):

Algorithm 1. Standard 3D Region Growing

```
Inputs : point cloud = $\{P\}$, normal vectors of points $\{N\}$, curvatures $\{C\}$,
         finding function of neighbor points $\Omega(\cdot)$, angle threshold $\theta_{th}$, curvature threshold $C_{th}$.
Initialize : region list $\{R\} \leftarrow \varphi$, available points list $\{A\} \leftarrow \{1 \cdots P_{count}\}$.
while $\{A\}$ is not empty do
  Current region $\{R_c\} \leftarrow \varphi$; Current seeds $\{S_c\} \leftarrow \varphi$
  Point with minimum curvature in $\{A\} \rightarrow P_{min}$
  $P_{\min} \xrightarrow{insert} \{S_c\} \,\&\, \{R_c\}$
  $P_{\min} \xrightarrow{remove} \{A\}$
  for $i = 0$ to size ($\{S_c\}$) do
    Find $k$-nearest neighbors of current seed point $\{B_c\} \leftarrow \Omega(S_c\{i\})$
    for $j = 0$ to size ($\{B_c\}$) do
      Current neighbor point $P_j \leftarrow B_c\{j\}$
      if $\{A\}$ contains $P_j$ and $\cos^{-1}(\left|\langle N\{S_c\{i\}\}, N\{P_j\}\rangle\right|) < \theta_{th}$ then
        $P_j \overset{insert}{\rightarrow} \{R_c\}$, $P_j \overset{remove}{\rightarrow} \{A\}$
        if $r\{P_j\} < C_{th}$ then
          $P_j \overset{insert}{\rightarrow} \{S_c\}$
        end if
      end if
    end for
  end for
  Add current region to global segment list $\{R_c\} \overset{insert}{\rightarrow} \{R\}$
end while
Sort $\{R_c\}$ according to the size of the regions.
Return $\{R_c\}$
```

A.2 RANSAC

The RANdom SAmple Consensus (RANSAC) algorithm was firstly introduced by Fischler and Bolles (1981) as a general framework for model fitting in the presence of outliers. Unlike the least squares, RANSAC is an iterative method to estimate the parameters of a known mathematical model by using measured points. The measured data includes inliers and outliers, while the outliers are not involved in the model fitting. RANSAC is a non-deterministic algorithm because it can only obtain a reasonable result under a certain probability. The probability will increase with the increase of the proportion of inlier values and the number of iterations. Compared with the least squares-based fitting, RANSAC-based fitting is more robust to gross errors (outliers). The whole process of RANSAC algorithm can be summarized as follows: An estimated model is generated by selecting the subspace of the dataset randomly and repeatedly, then the estimated model is tested by the remaining points in the dataset, yielding an associated score. Finally, the estimation with the highest score is output as the model of the whole dataset.

The set of inliers obtained for the fitting model is called the consensus set. RANSAC algorithm repeats the random sampling and fitting iteratively until the obtained consensus set in certain iterations has enough inliers and achieves its goal by repeating the following steps:

(1) Selecting a random subset of the original data, which is called hypothetical inliers;
(2) Fitting the set of hypothetical inliers into an expected model;
(3) All other data are tested against the fitted model. Those points that fit the estimated model within a set threshold are considered as part of the consensus set;
(4) The estimated model is considered to be good, provided that the majority of points have been classified as part of the consensus set sufficiently;
(5) The model will be improved by re-estimating it using all members of the consensus set.

The detailed process of standard RANSAC is presented as follows (Raguram *et al*. 2012):

Algorithm 2. Standard RANSAC

```
Input: $U, \eta_0, k_{max}, t$
Output: $\theta^*, I^*$
$k = 0, \quad I_{max} = 0$
while $k < k_{max}$ do
  1. Hypothesis generation
    Randomly sample minimal subset of $m$ points
    Estimate model parameters $\theta_k$
  2. Hypothesis verification
    Calculate the support set $I_k$
    if $|I_k| > I_{max}$ then
      $\theta^* = \theta_k, \; I_{max} = I^* = I_k$
      Re-compute $k_{max}$ from Equation (A.3) using $\varepsilon = |I^*| \div N$
    end if
    $k = k + 1$
end while
```

where a set of measurements U contains N data points, k and k_{max} are the current and maximum iterations, θ and I are model parameters and inlier point set respectively, η_0 is the confidence level, t is the set threshold and ε is the probability of inlier points.

The goal of RANSAC is to explore the space of model parameters efficiently so as to maximize some objective function C (Raguram *et al.* 2012). The objective function C in the standard RANSAC is to maximize the number of data points from all measurements U whose residual errors are smaller than the predefined threshold t when given a model with parameters θ_k. RANSAC attempts to maximize the cardinality of the consensus set. Thus, C is defined as

$$C = \sum_i \rho(e_i^2) , \tag{A.1}$$

where e_i^2 is the squared error between sampling points and the estimated model, and the cost function $\rho(\cdot)$ is defined as

$$\rho(e_i^2) = \begin{cases} 1 & for \;\; e_i^2 \le t^2 \\ 0 & for \;\; e_i^2 > t^2 \end{cases} , \tag{A.2}$$

In order to maximize the objective function C, RANSAC operates in a hypothesize-and-verify loop, repeatedly sampling subsets of the data to hypothesize the model parameters, and then verifying their support against all data points (Raguram *et al.* 2012). Each subset is a minimal sample with size m which is defined by the minimum number of points that can compute the model parameters uniquely. For example, three points are needed to determine a plane and four points are needed to determine a sphere. Thereby, RANSAC uses only as little data as possible to estimate model parameters compared with traditional regression methods that use all available data from the dataset (e.g., least squares).

The iteration of RANSAC algorithm should be limited under specific stopping criterion, thus a minimum number of iterations k can be defined to ensure that at least one outlier-free set of m points is sampled in RANSAC with confidence level η_0.

Here ε is assumed to be the probability of being an inlier point in the dataset when selecting one point. m is the minimum sampling subset size, then ε^m is the probability that m points are all inliers. So the probability that at least one of m points is an outlier (sampling failure) can be defined as $(1-\varepsilon^m)^k$. Therefore, the probability of success at least once in k-times sampling (confidence) is $\eta_0 = 1-(1-\varepsilon^m)^k$, which could be transferred into

$$k = \frac{\log(1-\eta_0)}{\log(1-\varepsilon^m)} . \tag{A.3}$$

The confidence η_0 is set to 0.95 or 0.99 typically. In fact, since ε is unknown in most cases, it is possible to use a worst-case estimate of ε to pre-compute the number of RANSAC trials (Torr *et al.*, 1998). However, it is more efficient to use an adaptive stopping criterion where $\mathcal{E}$ is initialized with a worst-case assumption (normally using zero), and updated based on the size of the maximum consensus set found as stated in **Algorithm 2**.

A.3 Iterative Closest Point (ICP)

The Iterative Closest Point (ICP) algorithm proposed by Besl and McKay (1992) is a classical method for rigid registration. It alternates between the closest point query in the target set and the minimization of the distance between corresponding points, and is guaranteed to converge to a locally optimal alignment. The general steps of standard point-to-point ICP algorithm for point cloud registration can be summarized as follows:

(1) For each point (from the whole set of vertices usually referred to as dense or a selection of pairs of vertices from each model) in the source point cloud, match the closest point in the target (reference) point cloud (or a selected set);
(2) Estimate the combination of rotation and translation (i.e., transformation parameters) using a root mean square point-to-point distance metric minimization technique that best aligns each source point to its match found in the previous step. This step may also involve weighting points and rejecting outliers prior to alignment;
(3) Transform the source point cloud using the estimated transformation parameters;
(4) Iterate steps (1) to (3) (re-construct the correspondences) until the improvement of estimated transformation parameters or the change of correspondence distances is smaller than a threshold.

The framework of the point-to-plane ICP is the similar to the point-to-point ICP, except that the distance of the correspondence is defined as the projection of the point-to-point distance onto the normal vector of the point in the target (reference) point cloud. The process of registering the source point cloud ($\boldsymbol{P}=\{\boldsymbol{p}_i\}_{i=1}^{n}$) to the target point cloud ($\boldsymbol{Q}=\{\boldsymbol{q}_j\}_{j=1}^{m}$) with an optimally estimated transformation matrix ($\hat{\boldsymbol{T}}$) incorporating the rotation matrix $\boldsymbol{R}$ and translation vector $\boldsymbol{t}$ by minimizing all point-to-plane distances (l_i) is illustrated in Figure A.1. Thus, the objective function can be defined as (Low, 2004)

$$\underset{4\times4}{\hat{\boldsymbol{T}}} = \begin{bmatrix} \underset{3\times3}{\hat{\boldsymbol{R}}} & \underset{3\times1}{\hat{\boldsymbol{t}}} \\ \underset{1\times3}{\boldsymbol{0}} & 1 \end{bmatrix} = \underset{\boldsymbol{R},\boldsymbol{t}}{\operatorname{argmin}} \sum_{i=1}^{N} \left\| (\boldsymbol{R}\boldsymbol{p}_i + \boldsymbol{t} - \boldsymbol{q}_i)\cdot \boldsymbol{n}_{q_i} \right\|^2, \tag{A.4}$$

where N is the number of correspondences and $\boldsymbol{n}_{q_i} = (n_{ix}, n_{iy}, n_{iz}, 0)^T$ is the unit normal vector at $\boldsymbol{q}_i = (q_{ix}, q_{iy}, q_{iz}, 1)^T$.

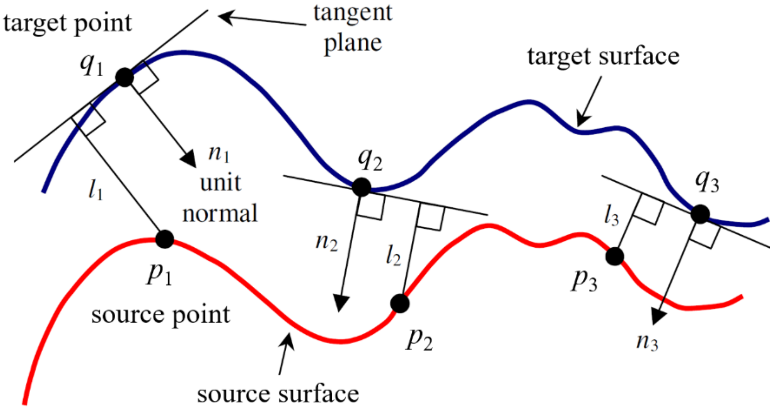


Figure A.1: The point-to-plane distance between two point cloud surfaces.

The solution of $\hat{\boldsymbol{T}}$ is a non-linear least-squares problem. However, as mentioned in Section 2.2, the rotation matrix $\boldsymbol{R}$ (see Equation (2.6)) can be approximated by

$$\boldsymbol{R} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \approx \begin{bmatrix} 1 & -R_z & R_y \\ R_z & 1 & -R_x \\ -R_y & R_x & 1 \end{bmatrix}, \tag{A.5}$$

when the rotation angles are very small (R_x, R_y, $R_z \approx 0$) (e.g., under a good coarse registration). Hence the optimal $\hat{\boldsymbol{T}}$ to be estimated can be written as

$$\hat{\boldsymbol{T}} = \begin{bmatrix} 1 & -R_z & R_y & t_x \\ R_z & 1 & -R_x & t_y \\ -R_y & R_x & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{A.6}$$

Then Equation (A.4) can be written as a linear expression of the six parameters (R_x, R_y, R_z, t_x, t_y, and t_z) as shown in Equation (A.7).

$$\begin{aligned} \boldsymbol{V} = \boldsymbol{AX} - \boldsymbol{L} = (\hat{\boldsymbol{T}} \cdot \boldsymbol{p} - \boldsymbol{q}) \cdot \boldsymbol{n}_q &= \left(\begin{pmatrix} 1 & -R_z & R_y & t_x \\ R_z & 1 & -R_x & t_y \\ -R_y & R_x & 1 & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} p_{ix} \\ p_{iy} \\ p_{iz} \\ 1 \end{pmatrix} - \begin{pmatrix} q_{ix} \\ q_{iy} \\ q_{iz} \\ 1 \end{pmatrix} \right) \cdot \begin{pmatrix} n_{ix} \\ n_{iy} \\ n_{iz} \\ 0 \end{pmatrix}^T \\ &= \underbrace{\begin{pmatrix} n_{1z}p_{1y} - n_{1y}p_{1z} & n_{1x}p_{1z} - n_{1z}p_{1x} & n_{1y}p_{1x} - n_{1x}p_{1y} & n_{1x} & n_{1y} & n_{1z} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ n_{Nz}p_{Ny} - n_{Ny}p_{Nz} & n_{Nx}p_{Nz} - n_{Nz}p_{Nx} & n_{Ny}p_{Nx} - n_{Nx}p_{Ny} & n_{Nx} & n_{Ny} & n_{Nz} \end{pmatrix}}_{\boldsymbol{A}\ (N\times 6)} \cdot \underbrace{\begin{pmatrix} R_x \\ R_y \\ R_z \\ t_x \\ t_y \\ t_z \end{pmatrix}}_{\boldsymbol{X}\ (6\times 1)} \\ &\quad - \underbrace{\begin{pmatrix} n_{1x}(q_{1x} - p_{1x}) + n_{1y}(q_{1y} - p_{1y}) + n_{1z}(q_{1z} - p_{1z}) \\ \vdots \\ n_{Nx}(q_{Nx} - p_{Nx}) + n_{Ny}(q_{Ny} - p_{Ny}) + n_{Nz}(q_{Nz} - p_{Nz}) \end{pmatrix}}_{\boldsymbol{L}\ (N\times 1)} \end{aligned} \tag{A.7}$$

Thus the transformation parameters (i.e., $\boldsymbol{X}$ in Equation (A.7)) can be solved by linear least squares using Equation (5.24) (i.e., the Gauss-Markov model).

Practically, this linear approximation method could be used even when the relative orientation between two point clouds is quite large (sometimes as large as 30°). However, this is strongly dependent on the geometry and the overlapping ratio between the two surfaces (Low, 2004).

A.4 Curvature Distribution of Spherical Clusters

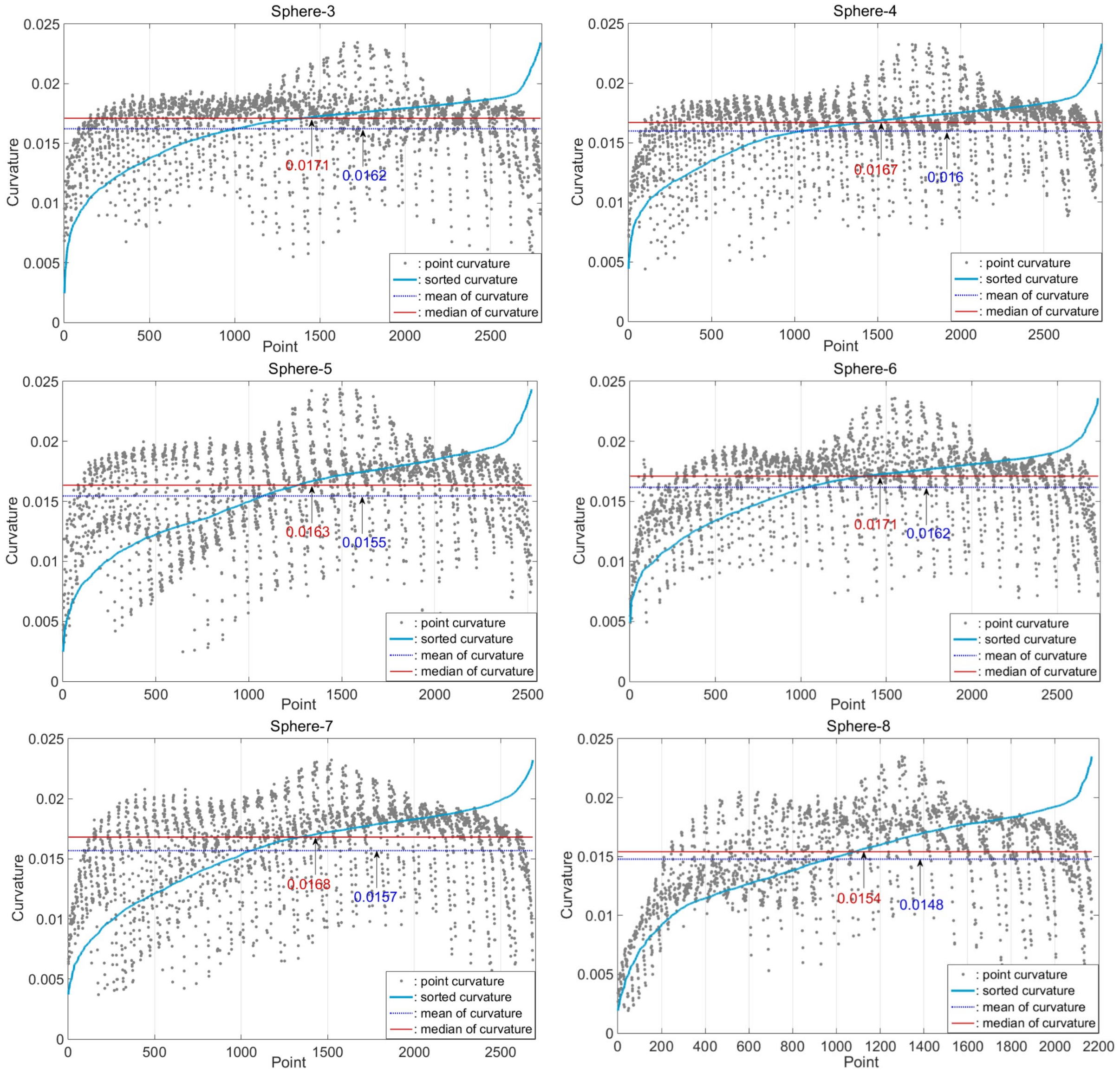
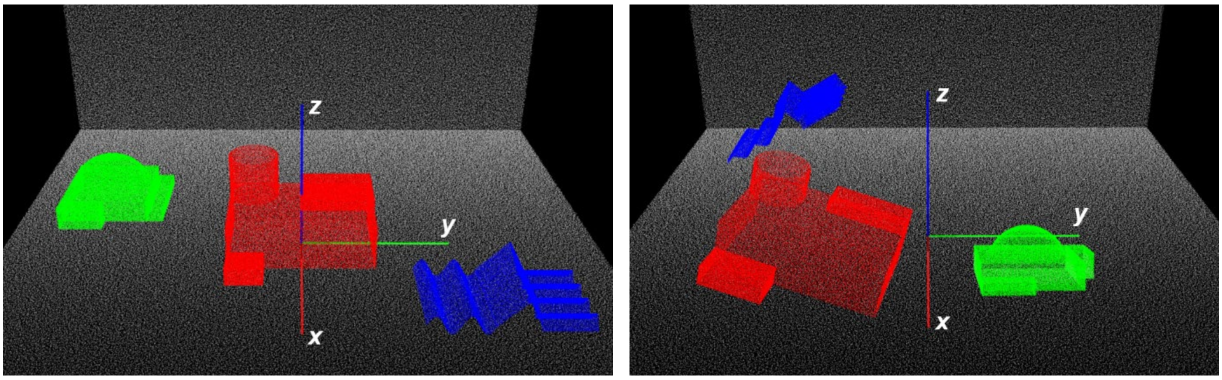


Figure A.2: The curvature distributions of the rest spherical clusters in the scanning test in Section 5.2.3.3.

A.5 Absolute Errors of Displacement Vectors

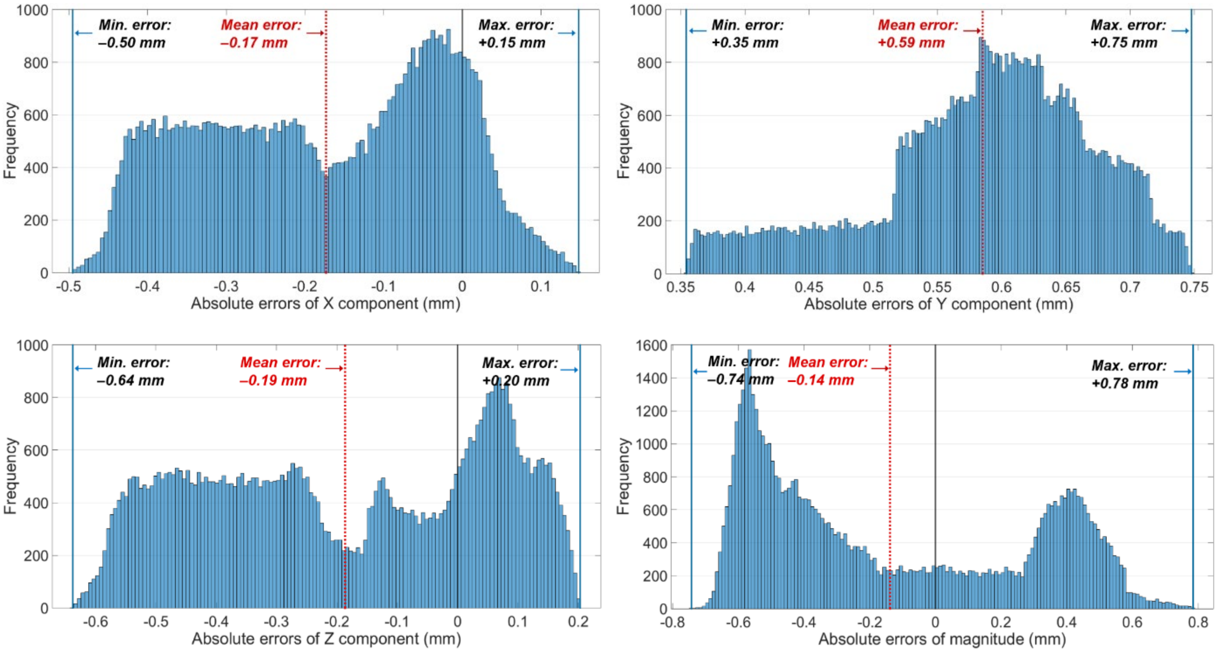


(**a**) Epoch-1 (**b**) Epoch-2

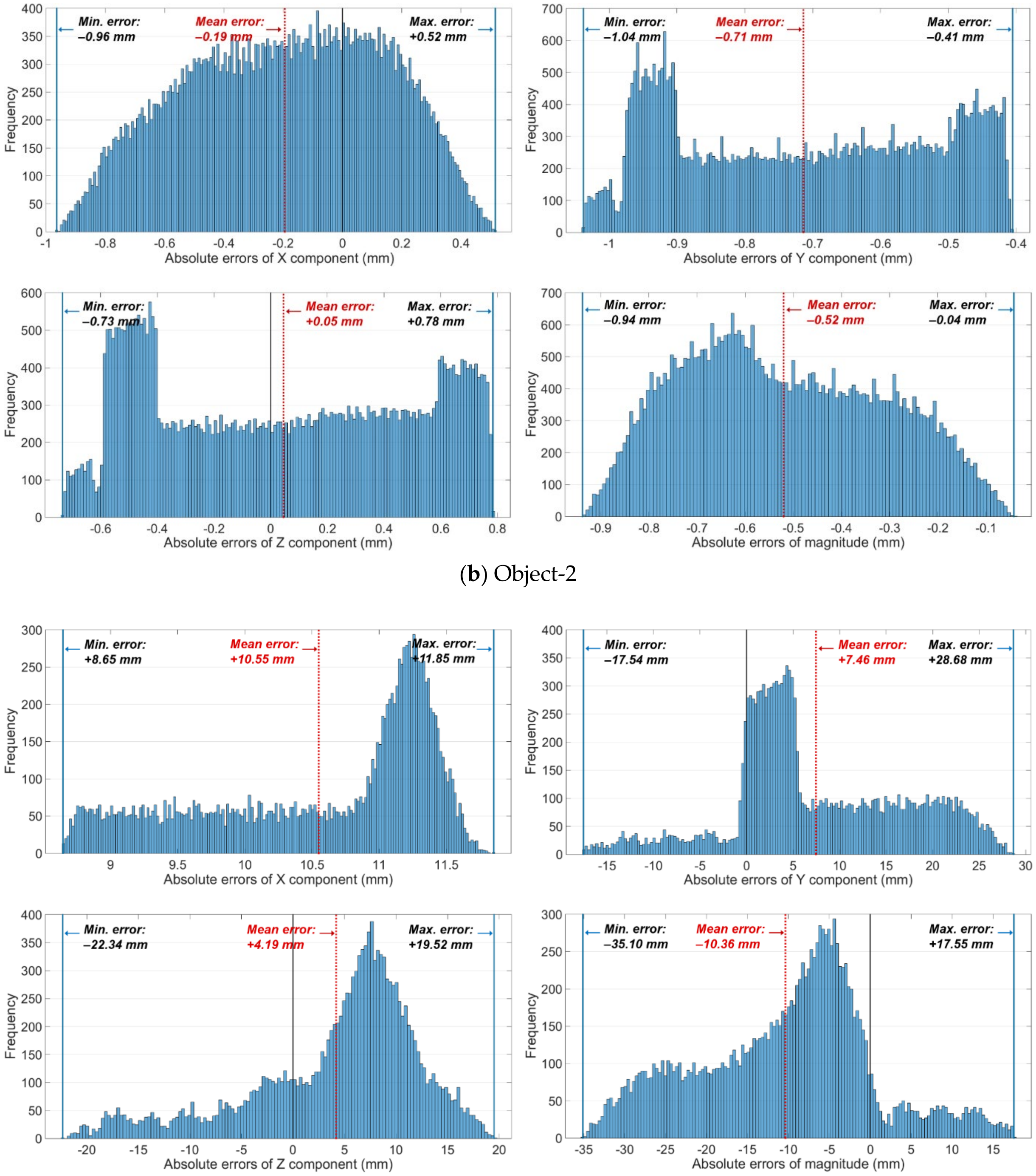
Figure A.3: Three simulated objects in Figure 5.42 are transformed by different rotations and translations in the second epoch.

Table A.1: Absolute errors of the local registration for three objects.

Object	Rotation angle (°)			Translation (mm)		
	ΔR_x	ΔR_y	ΔR_z	Δt_x	Δt_y	Δt_z
1	0.008	0.003	0.004	0.13	0.47	0.25
2	0.001	0.026	0.011	0.17	1.13	1.03
3	0.627	0.106	0.112	12.40	47.64	54.07



(**a**) Object-1



(**b**) Object-2

(**c**) Object-3

Figure A.4: Histograms of absolute errors of displacement vectors of three objects.