Cell-Based Deformation Monitoring via 3D Point Clouds

THÈSE N^{O} 5399 (2012)

PRÉSENTÉE LE 6 SEPTEMBRE 2012
À LA FACULTÉ DE L'ENVIRONNEMENT NATUREL, ARCHITECTURAL ET CONSTRUIT
LABORATOIRE DE TOPOMÉTRIE
PROGRAMME DOCTORAL EN INFORMATIQUE, COMMUNICATIONS ET INFORMATION

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES

PAR

Jing WU

acceptée sur proposition du jury:

Prof. P. Fua, président du jury
Prof. B. Merminod, directeur de thèse
Prof. M. Jaboyedoff, rapporteur
Prof. K. Schindler, rapporteur
Prof. R. Urbanke, rapporteur

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Suisse
2012

Acknowledgements

At the end of my Ph.D. study at EPFL, I would like to sincerely thank many people during my four-year journey.

I would like to express my deepest gratitude to my thesis supervisor Prof. Bertrand Merminod, for offering me the chance to be a member of Geodetic Engineering Laboratory (TOPO) and for funding my four-year Ph.D. work. I really appreciate Prof. Merminod's warm welcome when I arrived at EPFL, and his religious supervision during these four years, including providing scientific comments, proofreading my papers & thesis, etc.

I want to thank all of the jury members of my thesis defense, including Prof. Pascal Fua, Prof. Rüdiger Urbanke, Prof. Michel Jaboyedoff, and Prof. Konrad Schindler. I am appreciative of their time and efforts in reading and assessing this thesis work. Their constructive comments and suggestions are very grateful for me to improve this thesis significantly in last periods.

I express my gratitude to the Swiss Federal Railways, to Mr. Laurent Huguenin (from BBHN SA – a private surveying company), and to Prof. Vincent Barras (from HEIG VD – the Haute Ecole d'Ingénierie et de Gestion du Canton de Vaud) for the provision of interesting LiDAR datasets. These datasets enable me to validate this thesis work with real-life scenarios. In addition, many thanks go to Nicolas and Benoit (from HEIG VD) for one day collaboration on the field work at Valais. It was a completely exhausted day but joyful. I appreciate many fruitful discussions with Dr. Gauvin and Dario (from UNIL) on the development of toolbox and practical experiments using the Sandbox.

Many thanks go to the TOPO members, for their generous supports and kind friendship. A grateful acknowledge goes to Mr. Pierre-Yves Gilliéron, for his kind help and important collaborations on this thesis work. I want to thank Valérie (my first office mate at EPFL), for her kindness and encouragement during my fresh year at EPFL. I appreciate the friendship with Audrey (my second office mate) during the challenging thesis writing period. I also want to show my gratitude to Jan, Yannick and Yosef. A special thank goes to our secretary, Mrs. Véronique Boillat Kireev, who always provides with her warm-hearted help.

Last but definitely not least, I want to thank my beloved whole family: my father Jiyou and mother Guifang, my father-in-law Hanyu, mother-in-law Yunnan, younger-sister Huijun, and of course my husband Zhixian. Their unconditional love and endless supports help me to confront all challenges and to pursue my dream. This doctoral dissertation is dedicated to them.

Abstract

Deformation is one of the most important phenomena in environmental science and engineering. Deformation of artificial and natural objects happens worldwide, such as structural deformation, landslide, subsidence, erosion, and rockfall. Monitoring and assessment of such deformation process is not only scientifically interesting, but also beneficial to hazard/risk control and prediction. In addition, it is also useful for regional planning and development.

Deformation monitoring was driven by geodetic observations in the field of traditional geodetic surveying, based on the measurement of sparse points in a control network. Recently, with the rapid development of terrestrial LiDAR techniques, millions of points with associated three-dimensional coordinates (known as "3D point clouds") can be promptly captured in a few minutes. Compared to traditional surveying, terrestrial LiDAR offers great potential for deformation monitoring, because of various advantages such as *fast data capture*, *high data density*, and *precise 3D object representation*. By analysing 3D point clouds, the objective of this thesis is to provide an effective and efficient approach for deformation monitoring. Towards this goal, this thesis designs a new concept of "*deformation map*" for deformation representation and a novel "*cell-based approach*" for deformation computation. The main outcome of this thesis is a novel and rich approach that is able to automatically and incrementally compute a deformation map that enables a better understanding of structural and natural hazards with heterogeneous deformation characteristics. This work includes several dedicated contributions as follows.

- ***Hybrid Deformation Modelling.*** This thesis firstly provides a comprehensive investigation on the modelling requirements of various deformation phenomena. The requirements concern three main aspects, i.e., *what has deformation* (deformation object), *which type of deformation*, and *how to describe deformation*. Based on this detailed requirement analysis, we propose a rich and hybrid deformation model. This model is composed of *meta-deformation*, *sub-deformation* and *deformation map*, corresponding to deformation for a small cell, for a partial area, and for the whole object, respectively.
- ***Cell-based Deformation Computation.*** In order to automatically and incrementally extract heterogeneous deformation of the whole monitored object, we bring the "cell" concept into deformation monitoring. This thesis builds a cell-based deformation computing framework, which consists of three key steps: *split*, *detect*, and *merge*. Split is to divide the space of the object into many cells (uniform or irregular); detect is to extract the meta-deformation for individual cells by analysing the inside point clouds at two epochs; and merge is to group

adjacent cells with similar deformation together and to form a consistent sub-deformation. As the final result, an informative deformation map is computed for describing the deformation for the whole object.

- ***Evaluation of Cell-based Approach.*** To evaluate such hybrid modelling and cell-based deformation computation, this thesis extensively studies both synthetic and real-life point cloud datasets: (1) by imitating a landslide scenario, we generate *synthetic data* using Matlab programming and practical settings, and compare the cell-based approach with traditional non-cell based geodetic methods; (2) by analysing two *real-life cases* of deformation in Switzerland, we further validate our approach and compare the results with third party sources (e.g., results provided by a surveying company, results computed by using a commercial software like 3DReshaper).
- ***Extension of Cell-based Approach.*** At the last stages of this thesis work, we particularly focus on providing several technical extensions to enhance this cell-based deformation monitoring approach. The main extensions include: (1) supporting *dynamic cells* instead of uniform cells when splitting the entire object space, (2) finding *cell correspondence* for the deformation scenarios that have large deformation like rockfalls, (3) movement tracking with *data-driven cells* which have irregular cell shape that can be automatically determined by the deformation boundary itself, (4) designing an *adaptive modelling* strategy that is able to accordingly select a suitable model for detecting meta-deformation of cells, and (5) computing *deformation evolution* for a monitored object with more than two epochs of point cloud datasets.

Keywords: *deformation monitoring, 3D point clouds, terrestrial LiDAR, hybrid deformation modelling, cell-based computation, split-detect-merge, meta-deformation, deformation map*

Résumé

La déformation d'objets naturels et artificiels est l'un des phénomènes les plus importants en sciences de l'environnement. De telles déformations se déroulent dans le monde entier et provoquent différents types de dangers, tels que les déformations structurelles, les glissements, les effondrements, l'érosion et les éboulements. Le suivi et l'évaluation de tels processus de déformation ne sont pas seulement intéressants scientifiquement, mais sont aussi bénéfiques pour la gestion et la prévision des dangers et des risques, ainsi que pour le développement régional.

Traditionnellement, la surveillance de déformations a été conduite par l'utilisation d'observations géodésiques, c'est-à-dire basée sur la mesure de points dispersés. Avec le développement récent et rapide des techniques de LiDAR terrestre, la saisie de millions de points auxquels sont associées des coordonnées tridimensionnelles (connues sous le nom de "nuage de points 3D") en quelques minutes est devenue possible. Comparé aux méthodes traditionnelles, le LiDAR terrestre constitue une approche prometteuse pour la surveillance des déformations, en raison de ses divers avantages tels que la rapidité de saisie des données, la haute densité des données, et la représentation précise d'objets en 3D. Cette thèse examine la méthodologie et les défis liés à la surveillance de déformations à l'aide de nuages de points 3D saisis par le LiDAR terrestre. L'objectif principal de cette recherche doctorale est de proposer une approche de calcul automatique des déformations à partir de nuages de points 3D. Dans ce but, un nouveau concept pour la représentation de la déformation est proposé: celui de "carte de déformation". De plus, on développe une nouvelle approche, basée sur des cellules pour le calcul de déformation. Le principal résultat de ce travail est une méthodologie novatrice et riche pour la génération de cartes de déformations permettant de mieux comprendre les dangers structurels et naturels à différents niveaux. Ce travail apporte les quatre contributions suivantes.

- ***Modélisation hybride de déformations.*** Nous proposons une étude approfondie des exigences de modélisation en fonction du phénomène lié à la déformation. Ces exigences concernent trois aspects principaux, à savoir, *la présence ou non d'une déformation*, *le type de déformation*, et *la façon de décrire la déformation.* Sur la base de cette analyse détaillée, nous concevons un modèle de déformation hybride, composé de *méta-déformations*, de *sous-déformations* et *d'une carte de déformation*, correspondant respectivement à la description de la déformation pour une cellule, une zone partielle et l'ensemble de l'objet.
- ***Calcul de déformation basé sur des cellules.*** Afin d'extraire automatiquement et progressivement les déformations hétérogènes de l'ensemble de

l'objet surveillé, nous introduisons le concept de "cellules" dans le calcul de la déformation. Ce concept se décline en trois étapes principales: *séparer*, *détecter* et *fusionner.* L'étape de séparation consiste en une division de l'espace de l'objet en cellules uniformes; l'étape de détection extrait la "méta-déformation" pour les cellules en utilisant les nuages de points encapsulés dans chaque cellule; la fusion regroupe les cellules adjacentes ayant des déformations similaires et construit une "sous-déformation" cohérente pour une zone partielle. Ceci mène finalement à une carte de déformation de l'ensemble de l'objet.

- ***Evaluation de la détermination de déformations basée sur des cellules.*** Nous évaluons l'approche basée sur des cellules en utilisant trois types différents de nuages de points: (1) En imitant un scénario de glissement de terrain, nous générons des données synthétiques et comparons l'approche basée sur des cellules avec des méthodes géodésiques traditionnelles; (2) En analysant deux cas réels de déformation en Suisse, nous validons notre approche et la comparons avec les résultats de méthodes standards (par exemple, fournis par une entreprise de surveillance ou calculés à l'aide d'un logiciel commercial comme 3DReshaper).
- ***Extensions de l'approche basée sur des cellules.*** Au cours de la dernière étape de ce travail de thèse, nous nous concentrons particulièrement sur la conception d'applications techniques pour notre approche de surveillance des déformations. Les principales applications comprennent: (1) la prise en charge de cellules dynamiques et la détection de correspondances entre cellules dans le calcul de la déformation, (2) le suivi de déformation à l'aide de cellules dont la forme est détectée automatiquement à partir des contours arbitraires des objets soumis à des déformations, (3) la conception d'une stratégie de sélection de modèles adaptatifs pour améliorer la performance de la méta-déformation, et (4) le calcul de l'évolution de la déformation pour un objet surveillé, à l'aide d'ensembles de données de nuages de points correspondant à plus de deux époques.

Mots clés: *surveillance de déformations, nuages de points 3D, LiDAR terrestre, modélisation hybride de déformations, calcul de déformations basé sur des cellules, division-détection-fusion, méta-déformation, carte de déformation*

Contents

Abstract **iii**

Résumé **v**

Contents **vii**

List of Figures **xi**

List of Tables **xv**

List of Algorithms **xvii**

1 Introduction **1**
- 1.1 Background . . . 1
- 1.2 Motivation . . . 2
- 1.3 Core Issues, Assumptions & Design Strategies . . . 3
- 1.4 Contributions . . . 5
  - 1.4.1 Deformation modelling requirements . . . 5
  - 1.4.2 The hybrid deformation model . . . 6
  - 1.4.3 Automatic deformation computation . . . 7
  - 1.4.4 Evaluation and benchmark . . . 8
  - 1.4.5 Extensions of cell-based approach . . . 9
- 1.5 Thesis Organisation . . . 10

2 State of the Art **13**
- 2.1 Introduction . . . 13
- 2.2 Basics of Deformation Monitoring . . . 13
  - 2.2.1 What is deformation? . . . 14
  - 2.2.2 Deformation monitoring process . . . 14
- 2.3 Deformation Measurement . . . 16
  - 2.3.1 Classical geodetic methods . . . 17
  - 2.3.2 Remote sensing techniques . . . 22
- 2.4 Deformation Analysis . . . 27

2.4.1 Analysis of discrete points 28
2.4.2 Deformation estimation 28
2.5 Terrestrial Laser Scanner 30
2.5.1 Working principle 30
2.5.2 Classification of TLS 30
2.5.3 Calibration of TLS 32
2.5.4 Data acquisition via TLS 33
2.6 Point Cloud Processing 34
2.6.1 Point cloud registration 34
2.6.2 Point cloud segmentation 36
2.6.3 3D object reconstruction via point clouds 37
2.7 Deformation Monitoring via Point Clouds 38
2.7.1 Related work in computer vision and graphics 38
2.7.2 Related work in geomatics and geological fields 42
2.7.3 Comparison between two different types of field 43
2.8 Summary 44

3 Deformation Modelling **45**
3.1 Introduction 45
3.2 Medelling Requirements 45
3.2.1 Deformation object - “What” 46
3.2.2 Deformation type - “Which” 47
3.2.3 Deformation description - “How” 51
3.3 Global Model 55
3.4 Hybrid Model 57
3.4.1 Meta-deformation 59
3.4.2 Sub-deformation 60
3.4.3 Deformation map 61
3.4.4 Discussion 62
3.5 Summary 63

4 Cell-based Deformation Computation **65**
4.1 Introduction 65
4.2 Deformation Computation 65
4.2.1 Problem statement 66
4.2.2 Why cells for deformation computation? 66
4.2.3 Computing framework 69
4.3 Step I - Split 69
4.4 Step II - Detect 72
4.5 Step III - Merge 80
4.6 Summary 83

5 Evaluation of Cell-based Approach 85
5.1 Introduction . . . 85
5.2 Evaluation by Synthetic Data . . . 85
  5.2.1 Data from synthetic scenario . . . 86
  5.2.2 Detection of displacement . . . 87
  5.2.3 Detection of angle variation . . . 92
  5.2.4 Detection of rigid transformation . . . 95
5.3 Case Study I - Landslides in Flamatt . . . 98
  5.3.1 Data description . . . 98
  5.3.2 Results by cell-based approach . . . 98
5.4 Case Study II - Erosion on a Concrete Slope . . . 102
  5.4.1 Data description . . . 102
  5.4.2 Results by cell-based approach . . . 103
5.5 Comparison . . . 106
  5.5.1 Introduction of 3DReshaper . . . 106
  5.5.2 Synthetic data - comparison with ground-truth . . . 110
  5.5.3 Case study I - comparison with the Results from BBHN SA . . . 111
  5.5.4 Case study II - comparison with surface-based approach . . . 112
5.6 Summary . . . 115

6 Extension of Cell-based Approach 117
6.1 Introduction . . . 117
6.2 Dynamic Cells . . . 117
6.3 Cell Correspondence . . . 121
  6.3.1 Practical data collection . . . 123
  6.3.2 Finding cell correspondence . . . 127
6.4 Movement Tracking with Data-driven Cell . . . 130
6.5 Adaptive Model . . . 136
6.6 Evolution of Deformation Map . . . 140
6.7 Summary . . . 145

7 Conclusion and Outlook 147
7.1 Conclusion . . . 147
7.2 Limitations and Outlook . . . 151

Bibliography 153

Curriculum Vitae 175

List of Figures

1.1 Framework of cell-based deformation computation 7

2.1 Categorisation based on deformation object scale 15
2.2 Relative network and absolute network 18
2.3 Absolute and relative network in a monitored region 19
2.4 One unstable terrain object under monitoring using discrete points 27
2.5 Working principle of a Riegl TLS 31
2.6 Coordinate system of point clouds 31

3.1 Rigid deformation types 48
3.2 Non-rigid deformation types 50
3.3 Vertical and horizontal displacement of a point from Epoch I to Epoch II 52
3.4 Rotation of an area from Epoch I to Epoch II 53
3.5 Anisotropic scaling change of an object from Epoch I to Epoch II 53
3.6 Simple shear of a line from Epoch I to Epoch II 54
3.7 Low-sampling datasets of a monitored object 56
3.8 Point cloud datasets of a unstable object 58
3.9 Two corresponding point cloud datasets at Epoch I and Epoch II respectively . . 59
3.10 Deformation map of one monitored object 60
3.11 Data-driven cell 62

4.1 Displacement detection 67
4.2 R-tree structure for spatial access 70
4.3 Point clouds in cells after splitting 72
4.4 Vertical information representation for cells 73
4.5 Vertical displacement detection for cells 74
4.6 Angle representation for cells 76
4.7 Rigid transformation detection 79
4.8 Deformation map generation 80
4.9 Initial deformation map (with assigned labels) 81
4.10 Intermediate deformation map (after merging) 81
4.11 Final deformation map (after removing outlier) 81

5.1 Subsidence a synthetic surface 86
5.2 Two point cloud datasets of the simulated subsidence at Epoch I and Epoch II (sampling density: 5 mm * 5 mm, standard deviation $\sigma = 3$ mm) 87
5.3 Displacement of the synthetic subsidence between Epoch I and Epoch II via different detection methods 88
5.4 Displacement of the imitated subsidence between Epoch I and Epoch II via different detection methods (cell size: 5 cm) 89
5.5 Distribution of displacement detected by min, max, mean and median with the same cell size as 5 cm 91
5.6 Angle variation of the slope between two Epochs 93
5.7 Histogram of the number of the points in each cell at Epoch I (sampling density: 5 mm * 5 mm) 94
5.8 Estimation error with regard to noise level (σ) and cell size (w) 95
5.9 Cell-based ICP approach 97
5.10 Mean squared distance via traditional and cell-based ICP 97
5.11 Mean distance via traditional and cell-based ICP 98
5.12 Monitored slope at Flamatt 99
5.13 Deformation map of the whole slope 99
5.14 Deformation distribution (histogram) by “min” 100
5.15 Deformation distribution with regard to different cell size 100
5.16 Displacement extracted by different deformation estimation 101
5.17 Deformation distribution (histogram) of vertical displacement result 101
5.18 A concrete slope 102
5.19 Deformation map of the point cloud between Epoch 0 and Epoch 2 104
5.20 Distribution of displacement between Epoch 0 and Epoch 2 105
5.21 Distribution of displacement with different detection methods 106
5.22 A point cloud at a chosen epoch (Epoch I) 107
5.23 A mesh at a chosen epoch (Epoch I) 107
5.24 Inspection of the point cloud and its mesh at epoch 0 108
5.25 Displacement inspected by surface-based approach (3DReshaper) 110
5.26 Contour map of the whole slope at Flamatt 112
5.27 Contour map of the deformation for the zone marked in Figure 5.26 – inside the ellipse, which is enlarged by 15 times 112
5.28 Erosion map of one area in the concrete slope by 3DReshaper 113
5.29 Deformation histogram by surface-based approach 113
5.30 Histogram by cell-based and surface-based approach 114

6.1 Split strategy 118
6.2 An example of dynamic cell split 120
6.3 Cell correspondence between Epoch I and Epoch II 122
6.4 An object on a slope 123
6.5 Rotation of an object on a slope 124
6.6 Rotation of an object on a slope (Top view) 124
6.7 Translation of an object on a slope 125

6.8 KONICA MINOLTA VIVID 9i laser scanner 125
6.9 Sandbox 126
6.10 A sand slope in a box 126
6.11 Changes of the slope surface at Epoch I and Epoch II 127
6.12 Practical experiment of a rockfall deformation on a slope (front view) 127
6.13 Split cells of a monitored slope at Epoch 1 and Epoch 3 128
6.14 Object detection at Epoch 1 133
6.15 Object detection using angle variation 134
6.16 Object movements from Epoch 1 to Epoch 4 (only show the moving object) . . . 135
6.17 Rotation of deformation vector around the X-, Y- and Z- axis between Epoch i and Epoch j 135
6.18 Deformation map between Epoch 1 and 2 estimated by a uniform model (either linear or order-2 polynomial) and adaptive models. ($cellsize : 5\ cm$) 138
6.19 Statistical analysis of deformation map between Epoch 1 and 2 estimated by an adaptive model ($\delta = 0.002\ m,\ cellsize = 5\ cm$) 139
6.20 Statistical analysis of deformation map between Epoch 1 and 2 estimated by an adaptive model ($\delta = 0.0012\ m,\ cellsize = 5\ cm$) 140
6.21 Deformation map between Epochs 1, 2, 3 and 4 with cell size 5 cm 141
6.22 Deformation map between multi-epochs with cell size 5 cm 143
6.23 Deformation map between Epochs 1, 2, 3 and 4 with cell size 10 cm 144
6.24 Deformation map between multi-epochs with cell size 10 cm 145

List of Tables

2.1 Comparison of geodetic and non-geodetic measurement 17

3.1 Summary of deformation objects 47
3.2 Summary of deformation parameters 54
3.3 Notations of symbols 62

5.1 Error analysis of deformation estimation methods 91
5.2 Error analysis of deformation detection with different cell sizes 92
5.3 Error analysis of deformation detection from datasets with different noises 92
5.4 Selected results of the estimated parameters, gray parts are rejected estimation examples 94
5.5 Details of the point cloud datasets at four epochs 103
5.6 Number of cells in cell-based approach with different cell sizes 103
5.7 Minimum and maximum of the inspection result by 3DReshaper 109
5.8 Statistical analysis of deformation detection from the same datasets by cell-based and surface-based approach (unit: mm) 111

6.1 Iteration number of ICP (N) between two cells at Epoch 1 and Epoch 3 129
6.2 Mean squared distance between two cells at Epoch 1 and Epoch 3 130
6.3 Rotation angle of deformation vector around the X-, Y- and Z- axis between two neighbouring epochs (from Epoch *1* to Epoch *4*) 136
6.4 Translation of the stone (i.e., irregular cell) along the X-, Y- and Z- axis between two neighbouring epochs (from Epoch *1* to Epoch *4*) 136
6.5 Statistical analysis of deformation map estimated by adaptive model 137
6.6 Statistical results of deformation map with cell size 5 cm ($unit : m$) 142
6.7 Statistical results of deformation map with cell size 10 cm ($unit : m$) 143

List of Algorithms

4.1 Split . . . 71
4.2 planeFit . . . 74
4.3 RANSACplaneFit . . . 75
4.4 getVerticalDisplacement . . . 76
4.5 getAngleDifference . . . 77
4.6 getRigidTransformation . . . 78
4.7 Detect . . . 79
4.8 Merge . . . 82
4.9 expandArea . . . 82

6.1 dynamicCells . . . 119
6.2 cellCorrespondence . . . 122
6.3 MovementTracking . . . 132
6.4 findCellBound . . . 132

Chapter 1

Introduction

Nothing endures but change.

Heraclitus (540 BC-480 BC)

This chapter introduces this thesis in general, including the background in Section 1.1, the motivation and the main objective in Section 1.2, the core research issues, research assumptions and design strategies in Section 1.3, the major contributions of this thesis in Section 1.4, and this dissertation organisation in Section 1.5.

1.1 Background

Deformation monitoring is one of the important topics in environmental science and engineering. Based on the category of the monitored object, deformation monitoring can be divided into *structural deformation monitoring* and *natural hazard assessment*. Structural deformation monitoring focuses on the human-made structures such as dams, bridges, and towers [BHK98, FR00, NPS06, Sch06]; whilst natural hazard assessment analyses the natural surfaces including glaciers, slope, regions of landslides [SKW$^+$08, CFP09, ERH10]. The goal of monitoring deformation of such artificial and natural objects is to assess their deformation status, to possibly predict the damage caused by the deformation of such unstable object, which can ensure appropriate reactions in advance. Deformation can bring substantial damages to nature and human life. Because of the intrinsic complexity and high impact of this phenomenon, deformation monitoring is a challenging engineering task as well as a hot research topic.

As one type of the most frequent deformation scenarios in real-world environment, landslides are the major natural hazards on Earth and result in a loss of life and great property damage. Taking Switzerland for example, during the last 36 years (1972-2007), an average annual financial damage of about 20 millions CHF and an average of 1 death per year are directly caused by landslides [LHRB05, HBH09]. Not only in Switzerland, landslides occur widely in mountainous and hilly regions in Europe, as well as all around the world. In addition, landslides have an increasing trend due to the factors of population growth and expansion, and climate change [FCB$^+$08]. The impact of this dynamic process depends primarily on its *size*, *speed*, and *location*;

and its impact grows significantly when it involves infrastructure and people in the context. For example, deformation can bring a huge economic cost according to its location, e.g., a densely populated area.

During the last decades, deformation measurement has been extensively performed by many classical geodetic surveying techniques like *theodolites*, *photogrammetry*, *levels* and GNSS (*Global Navigation Satellite Systems*) [KP93, MMP99, MDH$^+$02]. These techniques primarily focus on the monitoring of a small set of specific points (e.g., under a geodetic control network) with the time series of coordinates. The density of these points is quite low, usually one or two points per square kilometre to monitor the terrain surface. For example, Kalkan et al. study the deformation on Atatäk dam embankment (height 169 m, length 1,819 m) in Turkey with 200 points [KAB10]. As a result, these surveying techniques with low-density measurements cannot provide an informative description of the changes for an unstable object under monitoring. Therefore, a high density sampling technique like terrestrial LiDAR (Light Detection And Ranging) opens a new potential to reach more informative deformation of the monitored object.

Terrestrial LiDAR is a relatively new and revolutionary surveying technology developed in recent decades. As an advanced technique, LiDAR can provide a rich set of three dimensional data including coordinates along X, Y and Z axis, as well as other information like intensity, colour. Millions of data measurements, which can be acquired very promptly in a few minutes, are commonly denoted as "*3D Point Clouds*". Compared to traditional surveying techniques, terrestrial LiDAR shows great potential for deformation monitoring and has been emerging as an important and attractive research topic, because of its advantages of *fast data capture*, *high data density*, *3D object representation*, as well as *user-friendly operation*. In particular, the high-density 3D points captured by LiDAR provide a chance for identifying detailed and distinctive deformation characteristics in partial areas of a big monitored object. Furthermore, a large amount of such 3D data induce new research challenges such as *huge data management*, *outlier filter*, *hole filling*, *extraction of useful information* (e.g., deformation, interesting areas), and *3D object reconstruction*. In this thesis, we focus on studying huge point cloud data management, and automatic deformation extraction from different epochs of point clouds.

1.2 Motivation

Resulting from the previous background discussion, we observe that: (1) Deformation monitoring of landslides alike natural hazards is a very important issue in environmental science and engineering, and such monitoring is useful for reducing/preventing damages; (2) Classical surveying techniques only provide data measurements with very low sampling, which can not provide an informative or detailed deformation description, particularly for a large monitored object; (3) Terrestrial LiDAR has alternative advantages in capturing high-density 3D point cloud data that opens substantial potential for the applications of deformation monitoring.

Therefore, the motivation of this thesis work is to analyse the huge amount of 3D point cloud datasets from terrestrial LiDAR and to compute deformation for the whole object monitored. More concretely speaking, we aim at building "*a novel and advanced deformation monitoring method via 3D point clouds*" that is able to generate an informative deformation description for a large monitored object, by analysing 3D point cloud data measured on this object using terrestrial LiDAR. Our method is able to automatically identify deformation from point clouds. This concerns with the whole monitored object and requires no needs of *a priori* knowledge of selected regions. Towards this objective, this thesis provides a rich *hybrid deformation modelling* (including *meta-deformation* for a cell, *sub-deformation* for a partial area, and *deformation map* for the whole object) to comprehensively present different levels of deformation description. To utilize this hybrid model, this thesis accordingly designs *a cell-based deformation computation framework* to automatically and incrementally compute different levels of deformation that are defined in the hybrid model. This computation process should be self-contained, and it includes three main steps, i.e., "split", "detect", and "merge".

1.3 Core Issues, Assumptions & Design Strategies

As described, the motivation and objective of this thesis is to study the research topic on *deformation monitoring via 3D point clouds*. We analyse high density 3D point data to automatically compute deformation information. This research topic is very challenging, as there are a couple of research problems and fundamental issues that need to be explored and answered during this thesis work. In this section, we identify the core issues of this thesis, discuss the assumptions of our proposal, and provide the underlying design strategies.

Core Issues. The main research issues in this thesis is on understanding the problem of "*deformation modelling*" as well as building the technique of "*deformation computation*".

- *Issue-1: What is deformation, and how to model deformation?* – We first need to understand the phenomena of deformation, and identify the main modelling requirements to represent a deformation. In particular, we need to identify the deformation object (i.e., what has deformation), and the deformation type (i.e., which form of deformation and how is the deformation granularity). The research focus here is on designing a rich model that is able to comprehensively modelling deformation for a large area/object with heterogeneous deformation characteristics.

- *Issue-2: How to monitor (or automatically compute) deformation?* – In this thesis, we want to build a method that is able to extract the deformation information from dense point clouds recorded at two different time instants. The fundamental question here is how to build an automatic monitoring (or computing) procedure for extracting such deformation information, without given any prior knowledge (e.g., which part should use which model). Of course, this automatic computation process should be able to take good advantage of the rich deformation model that has been mentioned in the *Issue-1*.

Assumptions. To answer the two core issues previously specified, this thesis is aiming at providing a rich deformation model for a large monitored area, together with a supporting computation process to automatically extract the deformation of this monitored area. We have following assumptions (and research focuses) in this thesis.

- *No prior knowledge* – One of the major assumption in this thesis is that we do not require any prior knowledge about the deformation area - e.g., which deformation model for which sub-area; there is no physical model given in advance. The objective is to automatically identify sub-areas with distinctive deformation characteristics.
- *Rigid deformation type* – In modelling deformation, we will identify different types of deformations, such as displacement, angle variation, and the combination of these rigid deformation. In addition, there are non-rigid deformation types like pure shear and simple shear. In this thesis, our research focus is on rigid deformation.
- *"Smooth" deformation in general* – Deformation in general is smooth, which means nearby areas have similar deformations (somehow continuous). In such case, if one small area has significantly different deformation from its neighbouring parts, we mainly consider this is outlier that might be caused by a detection error. Of course, in a real-life case, this might be not true and there is no guarantee to make such claim. Nevertheless, the main objective in this thesis is to automatically discover the major deformation characteristics in a smooth context.
- *3D and 2.5D model* – In this thesis, we study on using 3D point clouds to extract deformation information. In principle, we should model the area in pure 3D space, e.g., using 3D volume cells. However, most of real-life deformation / landslides scenarios (particularly for our two datasets) are interested in analysing the deformation of the 2.5-dimensional surface, e.g., displacement along Z dimension (i.e., subsidence). Therefore, this thesis mainly analyses the model of 2.5D, not 3D, for simplicity.

Design Strategies. Based on the two core issues and relevant assumptions of this deformation computation study using 3D point clouds, we conclude the following main design strategies of this thesis work.

- *Cell-based Approach* – The fundamental design choice in this thesis is to provide a cell-based deformation approach. We build a model to identify the deformation of a small area (e.g., a cell); this is because the cell is reasonably small in general, so that the deformation in a cell is somehow consistent and a single model can cover it. After detecting a cell deformation, we can identify the total deformation of the whole area by using hierarchical merging. The cell in this thesis is mainly about regular shapes, e.g., rectangles in 2.5D and cubes in 3D. In addition, we also study the irregular cells that are determined by deformation boundary (e.g., a non-cell based moving object like a falling stone).
- *A Hybrid Deformation Model* – To provide a rich deformation model for a large monitored area (or a big object), we choose a hybrid method to model such big area. The "hybrid" here has two meanings: (1) it supports different types of deformation in each individual partial area, e.g., displacement, rotation, even non-rigid deformation; (2) it is three levels of abstraction, from meta-model for cells, sub-model for partial areas, and a deformation map for the whole area.

- *Automatic Deformation Computation* – There is one more design strategy about automatic deformation computation without prior knowledge. In some scenarios, geologists or domain experts can provide some very useful prior information (e.g., which part has which type of deformation). However, such manual detection is non-precise and not suitable for a large area. Therefore, an automatic method needs to be designed for computing such heterogeneous deformations at the "hybrid" levels.

1.4 Contributions

Towards the objective of *deformation monitoring via 3D point clouds*, the main research outcomes of this thesis include the novel hybrid deformation modelling and the cell-based deformation computation. The hybrid model consists of a meta-deformation that supports different uniform deformation models for each cell, a sub-deformation model for describing distinctive deformation in each partial area, and a complete deformation description (deformation map) of the whole object. The cell-based deformation computation can incrementally and automatically extract informative and different levels of deformation without *a priori* geomorphology knowledge. More concretely speaking, this thesis is targeting at the following contributions.

1.4.1 Deformation modelling requirements

To propose a rich deformation model and a novel deformation computation method, we need to first understand the intrinsic characteristics of deformation scenarios. Therefore, the first contribution of this thesis is that we study various deformation scenarios and provide a comprehensive deformation modelling requirements analysis. This modelling requirement has three main elements, i.e., "what object has deformation", "which type of deformation", and "how to describe/quantify the deformation".

Deformation object ("What") – Deformation objects describe "*what has deformed*", i.e., the monitored object under the measurement using relevant terrestrial LiDAR equipments. There are two types of objects such as *artificial objects* and *natural objects* under investigation. Artificial objects include the human-built constructions like building, structures, bridges, dams, etc.; natural objects consist of some Earth environmental components in nature such as mountain sides, slopes, and valleys. For most of deformation studies like displacement, it is non-important to make the difference between artificial and natural objects. In this thesis about detailed cell-based approach, we do not make distinguish between artificial and natural objects like much literature.

Deformation type ("Which") – Deformation types concerns with "*which kind of deformation*". Deformation types have rigid deformation and non-rigid deformation. Rigid deformation is used for describing the absolute change of a monitored object from the complete object point of view. The basic elements of rigid deformation are translation (e.g., displacement along X, Y and Z axis) and rotation (i.e., angle variation/displacement). Compared to rigid deformation without changes on size or shape of the monitored object, non-rigid deformation concerns relative changes for the partial areas inside the object. Some deformation parameters such as scaling, shearing and stain are frequently used to represent non-rigid deformation.

Deformation quantification ("How") – After a clear investigation of "what" object has "which" type of deformation, quantification is the next crucial aspect for deformation modelling, and it comprises *quantitative* and *qualitative* methods. Quantitative method is to represent the deformation in term of numeric values like quantifying displacement, magnitude and direction; whilst, qualitative method provides more meaningful terms (e.g., "small", "middle", "big") to indicate the deformation level, which can be determined by domain experts based on empirical knowledge.

The details of our investigation on various deformation scenarios and our proposal on comprehensive deformation modelling requirements are presented in Section 3.2 of Chapter 3.

1.4.2 The hybrid deformation model

After analysing the detailed deformation modelling requirements, this thesis reviews traditional deformation analysis using a small set of observation points from geodetic surveying techniques. We identify that most of these traditional deformation analysis is using a global model, i.e., (1) focusing on a single deformation type (e.g., displacement, rotation, scaling), and more importantly (2) utilising a single model for the whole monitored object. To provide a richer model for deformation analysis using high-density 3D point clouds, we develop a comprehensive and hybrid model including *meta-deformation* for each cell, *sub-deformation* for a partial area and *deformation map* for the whole object to identify and demonstrate informative deformation for a large monitored object.

Meta-deformation – To provide an automatic deformation computation, we firstly divide the space of the monitored object into 3D cells. The meta-deformation is defined to model the deformation of individual cells. Meta-deformation is the building blocks of the complete deformation description for the whole object.

Sub-deformation – Following up the results of meta-deformation, we can merge the adjacent cells together to compose a partial area of the monitored object; of course, the cells merged together should have consistent (similar or even the same) meta-deformation, and share one sub-deformation description.

Deformation map – Finally, combing all of the sub-deformations together, we can generate a complete deformation description of the whole object, like a map indicates detailed deformation in each partial area; therefore, we call this final result "deformation map".

From meta-deformation, to sub-deformation, and finally to the deformation map, we can incrementally achieve an informative deformation description of the monitored object. We can apply this hybrid model in deformation computation, and gradually generate different levels of deformation abstraction. The details of this hybrid deformation modelling approach is discussed in Section 3.4 of Chapter 3.

1.4.3 Automatic deformation computation

Based on *a priori* knowledge of geomorphology, most of existing deformation analysis in the literature preliminarily select some partial areas in the monitored object, and apply a single deformation estimation model to extract the deformation information – namely, these methods focus on global modelling and analysing. Based on our hybrid model, we design a novel cell-based approach to automatically compute the informative deformation description of the whole monitored object, without the need of any given prior knowledge. This cell-based computation has three main steps, i.e., split, detect, and merge (see Figure 1.1).

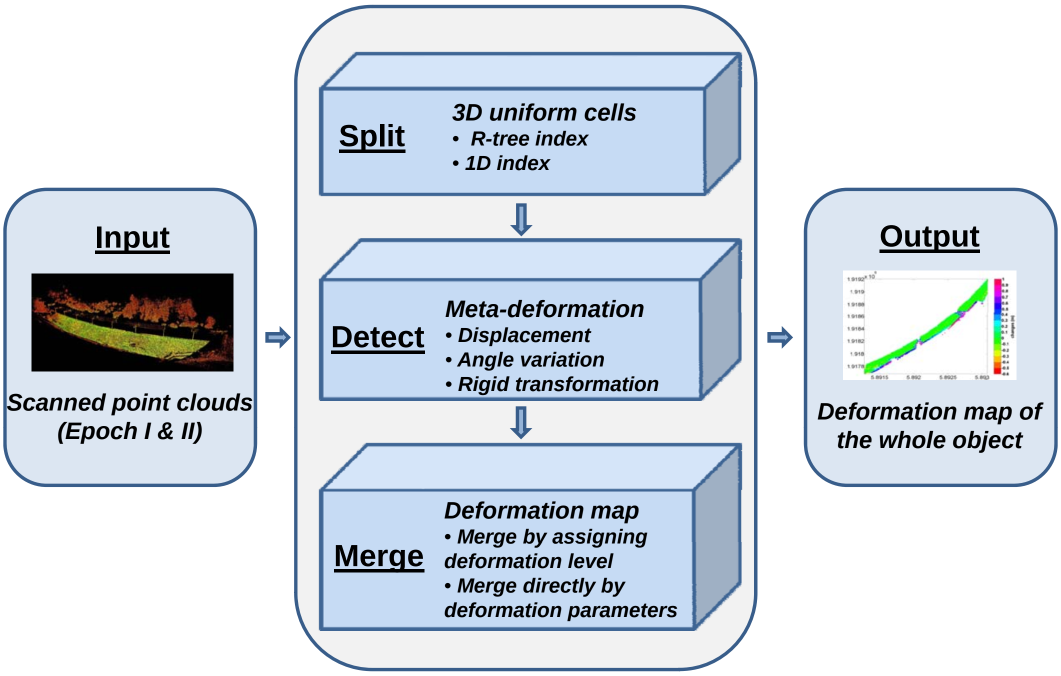


Figure 1.1: Framework of cell-based deformation computation

Split – The first step is to split the 3D point cloud of the whole object into many uniform cells, which provides a primary input to detect the meta-deformation. Another benefit of splitting the space into cells is that we can efficiently and easily manage the point cloud data and compute deformation incrementally, as the complete point cloud is usually very huge which cannot be directly processed at one time. In the split step, we can apply efficient space fast accessing techniques to improve our performance, such as the R-tree spatial indexing structure.

Detect – The second step is to detect the meta-deformation of cells, based on comparing the subset of point clouds in the cell at two different epochs (i.e., Epoch I and Epoch II corresponding to the time before and after the happened deformation event like a landslide). In this step, we focus on studying a couple of uniform deformation analysis models, such as vertical displacement, angle variation, and rigid transformation. This is because the cell is usually reasonably small and uniform deformation is applicable in a small cell.

Merge – After detecting the meta-deformation of cells, we need to merge adjacent cells with similar deformation characteristics together and generate sub-deformation for a partial area. Finally, the complete deformation map can be achieved after combining all of the sub-deformations. In contrast to the top-down split that divides the whole region into small cells, the merge is a bottom-up method that incrementally group cells together to formulate a sub-deformation, and to reach the deformation map finally.

The major advantage of this split-detect-merge procedure of deformation computation is its automation and completeness. There is no prior knowledge needed for selecting partial areas in advance. In addition, we can adopt and test various traditional geodetic surveying methods in detecting the meta-deformations. The details of this contribution of proposing the cell-based deformation computation are presented in Chapter 4.

1.4.4 Evaluation and benchmark

To evaluate our cell-based approach for deformation modelling (i.e., the hybrid model) and computation (i.e., the cell-based computation), we test it using both synthetic and real-life point cloud datasets from terrestrial LiDAR. Additionally, we provide exhaustive experiments in sensitivity analysis on some important parameters (e.g., cell size) during such deformation computation procedure.

Data simulation – We generate a point cloud dataset by a computing program implemented in Matlab to imitate a typical deformation scenario, i.e., landslides of a slope. In this synthetic scenario, a single surface (e.g., a mountain slope) subsides into two surfaces (sub-slopes), where the top-right one is sharper and the bottom-left one is flatter. We introduce the necessity of the cell-based approach, and perform the comparison between the cell-based approach and traditional non-cell based global method. Based on the ground-truth deformation we know from the simulation, we identify better performance from our cell-based approach. We apply three meta-deformation detection methods, including *displacement*, *angle variation* and *rigid transformation* to detect deformation of this slope. Additionally, we add extra noises to the data and analyse their influence in the final results of the computed deformation map, which shows the robustness of our approach.

Practical experiment – In addition to simulate point clouds using Matlab coding, we also design practical experiments to simulate relevant deformation scenarios. The practical experiments are performed under the collaboration with UNIL (University of Lausanne) using their “sandbox” and 3D digitizer. We design two practical simulation scenarios: one is for simulating the rockfalls on a slope; and the other is the erosion/movement of the slope itself. Particularly, the rockfall scenario is later used in our experiment in finding cell correspondence.

Landslides in Flamatt – We study a real-life landslide in Flamatt, Switzerland to evaluate the cell-based approach. This landslide has affected a major railway line. The size of this unstable slope is approximately 200 $m \times 25$ m; and two dense point cloud datasets from this slope are captured before and after the landslide, respectively. We perform the cell-based approach on this unstable slope and analyse deformation of this region by computing its deformation map. Compared to the results provided by BBHN (a Swiss surveying company), our result are consistent with their results in general. In addition, we perform sensitivity analysis of different cell sizes, and identify the optimal cell size for deformation detection.

Erosion on a concrete slope in Valais – We study another real-life erosion case of a concrete slope. This concrete slope locates between a dam and a natural riverbed; and its erosion is mainly caused by water flows. Four point cloud datasets are provided by Haute Ecole d'Ingenierie et de Gestion du Canton de Vaud (HEIG-VD). We compute the deformation maps of this region, and compare our deformation map with the results built by traditional surface-based approach (e.g., a point cloud processing software "3DReshaper" that we apply in this dissertation). Not only from the graphic view but also from the deformation distributions, we identify consistent results from our cell-based approach and from the 3DReshaper, this surface-based approach.

The details of experiments and evaluations of the hybrid modelling and the cell-based deformation computation are provided in Chapter 5.

1.4.5 Extensions of cell-based approach

After modelling deformation and computing it from real-life 3D dense point clouds, we further identify a couple of research directions and challenges to improve this cell-based deformation monitoring approach. At the last period of this thesis work, we provide several extended contributions, such as splitting object with "dynamic cells", finding "cell correspondence" in complex scenarios, tracking "data-driven cells" in movement scenario, supporting "adaptive model" selection in computing meta-deformation, and identify "deformation evolution" with multiple epochs of point cloud measurements.

Dynamic cell split – In the previous contributions, we have tested many sensitivity analysis on applying different cell size. However, the cell size is still uniform and static in the experiments. It is non-trivial to find an optimal cell size: (1) if it is too big, only a few cell exist, and all cells have huge points inside; this is under-splitting, and meta-deformation might be not very accurate. (2) if cell size is too small, there will be too many cells; this is over-splitting, computing meta-deformations for so many cells would be time-consuming. Therefore, as the first major extension, the enhanced cell-based approach is able to design a strategy of using dynamic cells for detecting meta-deformation. We design a top-down split strategy that is able to use an initial model to judge whether the current cell needs to be further split into two sub-cells or not.

Cell correspondence – The previous detecting methods are suitable for small deformation. This is because small deformation can guarantee the assumption that most points in on cell after deformation still stay in the same cell; therefore, we can apply the direct cell correspondence. However, some real cases can not satisfy this assumption because of their large deformation. Therefore, we propose a matching method to find cell correspondence, based on local search optimisation. In addition, we apply this method on a rockfall scenario with our practical experimental datasets, i.e., finding the corresponding cells with the rock at different epochs.

Movement tracking with data-driven cell – We have discussed about "regular 3D cell" and "direct cell correspondence". In these cases, the deformation is restricted to an exact multiple of the cell size (e.g., inside one cell or deformation from one cell to neighbouring cells). However, real-life cases are much more complex. For example, the cell cannot be always regular. Deformation has its own boundary. In addition, deformation could happen from one cell to another cell with overlapping part. Therefore, the exact multiple of the cell size is not always appropriate. To solve this problem, we introduce the "meta-cell" concept into the "movement tracking" extension and utilise the meta-cell to form "a data-driven cell with arbitrary boundary". Furthermore, the meta-cell can assist in finding a better cell correspondence. Now, the moving distance of a searching cell can be not only an exact multiple of the cell size, but also be some non-integer-multiple cell sizes. We apply this algorithm on a rockfall scenario with our experimental datasets, i.e., tracking the deformed parts like a rock with a series of epochs.

Adaptive modelling – In the earlier work of this thesis, only one detection method was applied for deformation computation from each cell. However, the real case is that a monitored object has heterogeneous deformation, which means several deformation models might exist simultaneously in this object. Therefore, we design adaptive model selection to pick the most suitable model for each cell, based on the geometric features of the point clouds inside the cell at two epochs. The algorithm can support the model selection between 1-order linear model and 2-order polynomial model.

Deformation Evolution – Two-epoch deformation analysis is the most general case study in many applications. In addition, multi-epoch point cloud datasets provide the possibility to analyse deformation evolution. In our erosion case, there are four-epoch point cloud datasets. In terms of analysing deformation evolution on this scenario, we recognise that the deformation on this slope mainly happened between Epoch 1 and Epoch 2; afterwards, the deformation is quite non-obvious (from Epoch 2 to Epoch 4).

The detailed contributions of these four main extensions to our cell-based deformation monitoring approach are presented in Chapter 6.

1.5 Thesis Organisation

This dissertation is organised in 7 chapters as follows,

- **Chapter 1** introduces the main background, motivation, and contributions of the "cell-based deformation monitoring via 3D point clouds" research in this thesis.

- **Chapter 2** presents the related work on deformation monitoring and terrestrial LiDAR. It demonstrates the advantages and disadvantages of the geodetic techniques for deformation measurement, as well as the classical deformation parameters/types of deformation analysis. It then introduces the research status and challenges that exist in 3D point cloud processing to identify deformation.

- **Chapter 3** focuses on identifying important modelling requirements of deformation. It provides the formal problem statement of our research challenges and proposes a comprehensive and hybrid model for deformation monitoring (including meta-deformation, sub-deformation, and deformation map), in contrast to traditional global modelling.

- **Chapter 4** explores deformation detection based on the hybrid model. Cell-based approach, a novel deformation computation method, is presented. This method provides a split-detect-merge procedure to automatically and incrementally identify rich deformation information from 3D point clouds of a large monitored region.

- **Chapter 5** describes experimental results with both synthetic data and real-life 3D point cloud data, and provides a benchmark system for testing our algorithms. We compare our cell-based deformation computation results with non-cell based results, ground-truth deformation from simulation, third party results provided by a surveying company, and results computed by a commercial software.

- **Chapter 6** investigates the extensions to enhance our cell-based deformation monitoring. There are four main extensions, including splitting the region with dynamic cells, finding cell correspondence, movement tracking with data-driven cell, building adaptive modelling, and computing deformation evolution.

- **Chapter 7** concludes the main findings of this thesis and points out the limitations of our work. Some interesting directions towards this topic for future work are also discussed.

Chapter 2

State of the Art

> *In the present state of the art*
> *this is all that can be done.*
>
> *Henry H. Suplee (1856-1943)*

2.1 Introduction

In this chapter, we present state-of-the-art about the methodology and technology of deformation monitoring, and describe their advantages and drawbacks. For a better understanding and a clearer organisation, we group these approaches in the literature into the following three main perspectives: *Deformation Measurement*, *Deformation Analysis*, and *Deformation Monitoring via Terrestrial LiDAR*.

This chapter is organised as follows: Section 2.2 provides some basic definitions of deformation; Section 2.3 discusses the techniques of deformation measurement, identifying the geodetic methods and non-geodetic ones; Section 2.4 describes the deformation analysis, focusing on existing statistical methods of estimating deformation parameters; Section 2.5 and Section 2.6 present the application of terrestrial LiDAR – where Section 2.5 describes the LiDAR technique whilst Section 2.6 discusses 3D point clouds processing techniques. Finally, Section 2.7 presents the related work of deformation monitoring in both computer vision/graphics and environmental fields and focuses on its applications in deformation monitoring via 3D point clouds.

2.2 Basics of Deformation Monitoring

This section introduces the fundamental concepts of deformation monitoring. We firstly provide general definitions of *deformation* and identify different types of *monitored objects*. Subsequently, we focus on explaining *deformation monitoring process*, and classify deformation monitoring based on the scale of monitored object or the type of deformation modelling.

2.2.1 What is deformation?

There are a lot of definitions about deformation. It is technically difficult and perhaps impossible to provide a universal definition which will be accepted in all relevant research domains. Here, we only summarise the most well-known and widely cited definitions.

- Deformation is a conception which is original from the theory of *elasticity and continuum mechanics* and can be defined as the alteration of form and shape in a term loosely [DL83].
- The deformation of an object is the result of a process, including not only the geometrical changes of an object but also the incorporation of the causative forces and the physical properties of the object [WH01].
- In materials science, deformation is a change in the shape or size of an object due to an applied force (the deformation energy in this case is transferred through work) or a change in temperature (the deformation energy in this case is transferred through heat) [Dav79, Zar91].

Although these definitions have distinct focuses and application scenarios, they have a common knowledge that deformation is a change of an object, and such change is caused by some intrinsic or external forces. Deformation describes the variation between two different states. For the detailed information about “what” is deformed (i.e., the monitored object) and in “which” sense (i.e., the deformation types generated by intrinsic or external force), different definitions provide different explanations.

The monitored object can be any object affected by natural or man-induced movements. Therefore, the monitored objects are typically classified into two categories: *natural structures* such as ground, slope, valley or mountain sides, and *artificial structures* like dams, bridges and buildings. The detailed deformation process can be presented in different ways, like landslides, rockfall, ground subsidence, erosion, etc. These different types of deformation can result in a huge property damage and even lost of life in real-world scenarios. The impact/damage of a deformation scenario depends primarily on its size, speed and location, especially when crossing infrastructures and involving people.

2.2.2 Deformation monitoring process

Deformation monitoring is the procedure to estimate the deformation of the monitored object, and further provides the qualitative or quantitative description of this natural phenomena. Due to the hazards caused by this phenomenon [LHRB05, HBH09], deformation monitoring has been becoming an active and important research field for many years, and still is a dominant topic in geodetic surveying. According to the force type, deformation monitoring can be categorised into *structural deformation monitoring* and *natural hazard assessment*. Monitoring deformation of artificial and natural objects should be capable of predicting the damage caused by the deformation of an unstable object; such deformation prediction can ensure the appropriate reaction in advance, and reduce the damages. There are several ways to expatiate on and classify deformation monitoring, e.g., depending on *the scale of monitored object* or *the type of deformation modelling.*

- *Scale of Deformed Object.* Deformation monitoring of different scales such as *local*, *regional*, *continental* and *global* are categorised according to the scale of the object under control. Deformation monitoring of a *local* scale is the estimation of the deformation of some artificial structure (e.g., a dam, a bridge and a slope along a highway or a railway line) and natural sites (like a ground subsidence or a bank erosion along a river) [Hud98, TLP06]. Deformation monitoring of a *regional* scale is the investigation of the stability of objects such as a valley and a mountain side [Kaa02, Riz02]. Deformation monitoring of a *continental* scale is the achievement of the deformation from the observations of national and continental geodetic network [RS05]. Deformation monitoring of a *global* scale is the study of the deformation of the earth as a whole like polar motion, variation of the earth's rotation, etc.; monitoring such *global* deformation typically applies data from geodetic space techniques, including *Satellite Laser Ranging (SLR), Very Long Baseline Interferometry (VLBI)*, and *Interferometric Synthetic Aperture Radar (InSAR)*, and so on [Nat81].

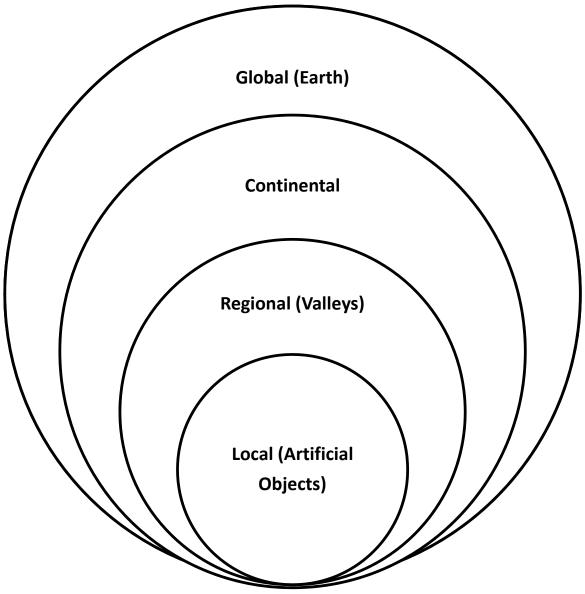


Figure 2.1: Categorisation based on deformation object scale

- *Type of Deformation Modelling.* Deformation modelling needs to consider *space* (where is the deformation), *time* (when is the deformation) and *causative forces* (by which and how is the deformation caused). Accordingly, the type of deformation modelling can be classified into *static model*, *kinematic model* and *dynamic model* [WH01]. Static model is the identification of the geometrical deformation (e.g., displacement, rigid transformation and volume deformation, etc.) of an object at two epochs or multi-epochs [Cas88]. Kinematic model is the description of the trend of the geometrical changes with a time series (e.g, velocity and accelerations, etc.) [PP83]. Dynamic model is the complete investigation

> on deformation process of an object including not only geometrical changes of the object in space and time domain, but also the causative factors of deformation and even the materials of the object [CF65].

Deformation monitoring is an interdisciplinary topic across the sciences of geomatics, geology and mechanics, as well as computer science. It contains studies from distinctive perspectives because of the complicated nature of this phenomenon. For example, (1) the scale of the monitored object can vary from global (e.g., earth) to local (e.g., an unstable slope along a railway line); and (2) the complicated state of deformation can be in different domains, including space, time and causative forces. In this thesis, from a perspective of geomatics, we focus on deformation computation from the geodetic observations (i.e., 3D point clouds) on the monitored object in spatial and temporal domains; we focus on identifying "what" objects have "which" types of deformation happened at "when", but not the "why" this happens by "which" forces. In addition, we analyse the *local* and *regional* scale deformation, and deformation of objects at the continental or global scales is out of the scope of this thesis.

The process of deformation monitoring consists of measurement and analysis phases [AOA+06]. Deformation measurement is the acquisition of the relevant observations of the changes that are collected by several surveying techniques at a time series (including at least two epochs); and deformation analysis investigates the geometrical changes of the object from the multi-temporal measurements, and provides statistical assessment of the derived deformation model and characteristics. The explicit concepts of these two parts are presented in Section 2.3 and Section 2.4, respectively. Recently, with the rapid terrestrial LiDAR techniques, 3D point clouds datasets can be easily captured and effectively used for deformation monitoring; and relevant techniques are summarised in Section 2.5 and Section 2.6.

2.3 Deformation Measurement

Deformation measurement is an essential prerequisite procedure to acquire the periodic observations of the monitored object. These observations offer the possibility for the subsequent deformation analysis such as deformation modelling, computation and statistical assessment. There are a numerous amount of techniques employed in deformation measurement, which can be mainly grouped into *geodetic* and *non-geodetic* methods [Chr86]. The geodetic methods of deformation monitoring capture the observations (e.g., angle and distance measurement) of the sampled points on the monitored object [Tor01]. Based on these observations, this method describes a global deformation of the monitored object. Non-geodetic methods apply some specialised equipments (like tiltmetres, strainmeters, extensometers, join-meters, plumb lines, micrometers) to measure the changes automatically and continuously [EEA04, KAB10]. Compared to geodetic methods giving the absolute changes on some points, non-geodetic methods provide the relative deformation inside the object under control. Detailed comparison between the geodetic and non-geodetic measurement is summarised in Table 2.1.

There are a numerous amount of geodetic techniques being applied for deformation measurement. We divide these geodetic techniques into two groups, i.e., *classical geodetic methods*,

Table 2.1: Comparison of geodetic and non-geodetic measurement

Geodetic method	**Non-Geodetic Method**
Global information on the behaviour of the monitored object	Local information inside the object
Intensive labour and requirement of skilled observers	Automatic and continuous monitoring
The absolute displacement determination of the selected points	Relative deformation identification within the monitored object and its surroundings

and recent measurement methods like *remote sensing*. Many classical surveying techniques for angle and distance measurements have been used to detect the movements. These methods include levels [CCRS86], theodolites [MMC02], Electronic Distance Measurement (EDM) [RRSJACBG02], total station and recent Global Navigation Satellite System (GNSS) techniques [BHK98, SDA05, BCK08]. Additionally, remote sensing techniques provide numerous kinds of datasets (e.g., satellite images, SAR and InSAR images, Photogrammetric and LiDAR datasets) for deformation measurement. Recently, advanced measurement technique like Light Detection And Ranging (LiDAR) shows great potential to track the geometrical changes of the monitored object [FM04, TGZG07, Lem10]. As a relatively new and even revolutionary surveying method, LiDAR technique can perform "surface measurement" on the monitored object, which is different from the former "single point measurement" using classical geodetic methods. The appearance of this technique provides a better understanding of the monitored object and offers the possibility of more detailed deformation computation or extraction [GSC$^+$06, JOA$^+$10]. For real-life deformation monitoring, the choice of the surveying techniques depends on a couple of deformation aspects, such as the condition of the monitored object (e.g, location of the object, safety considering of the operators), the requirement of the deformation (like accuracy, the scale of the object), the frequency of measurement and the economical factor, and so on.

2.3.1 Classical geodetic methods

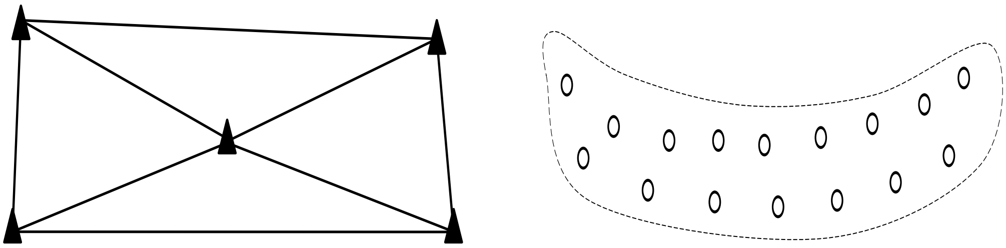
Traditional surveying techniques such as theodolite, EDM and total station are mainly applied for angle or distance measurement of the sampled points marked in the monitored objects [GW08]; GNSS (Global Navigation Satellite System) techniques can directly achieve the positions ($\langle longitude,\ latitude,\ altitude \rangle$, or simply noted as the $\langle X,\ Y,\ Z \rangle$ coordinates) of the control points on the object [GCR00]. These traditional surveying techniques and GNSS can be entitled "single point measurement" technique. For such technique, low sampling density points are deployed on the object for investigation, and need to be measured by these classical geodetic methods. These sampling points can form a geodetic network inside or around the object. The observations of this control network represent the object under investigation. Therefore, choosing the sampling points to design a geodetic control network is a crucial step in these classical geodetic methods for deformation measurement.

Design of Geodetic Network - Design of the geodetic network is the exploration of the network's structure and the determination of the connection between the observations by different types of measurements (e.g., direction measurement and distance measurement) [KF88]. The objectives of optimising deformation geodetic network are: (1) to achieve the required accuracy level of detected deformation; (2) to develop a reliable mathematical model; (3) to produce adequate sensitivity with regard to the testability of deformation; and (4) to design the practical observation program under a limited financial constraint [Kua96].

As a consequence, designing geodetic network needs to consider several aspects, such as *accuracy*, *reliability*, *sensitivity* and *cost* [Cas88]. *Accuracy* is usually expressed by the variance-covariance matrix of the estimated deformation parameters, and is a representation of random error propagation (one aspect of the geodetic network). *Reliability* of the geodetic network can describe the influence of the random and systematic errors on the deformation results [See01] and can be increased by additional observations and repeated measurements between the neighbouring points. *Sensitivity* is an indication of the deformation detection ability from the geodetic network. *Cost* is the financial constraint which cannot be overlooked as the designed network should be economically feasible.

Geodetic Control Network - For all of these classical geodetic methods used for deformation monitoring (e.g., *levels*, *theodolites*, *GPS* and *GNSS*), we can identify two fundamental types of control networks in the geodetic methods, i.e., *absolute network* and *relative network* [Cas88].

- *Absolute Network.* All of the points (or at least some of the points) of the network are assumed to be outside the monitored object, e.g., in the foundation or surrounding terrain beyond the area [Chr81, BHK98]. These points are defined as "*reference points*" and are not affected by the deformation on the object, as shown in Figure 2.2(a). These points need subsequently periodic observations to assess the absolute data for detecting long-term deformation [EEA04]. From these observations, we determine the absolute movement of the object. The following analysis of this network is to confirm the stability of the reference points and to identify the external force that caused the single point displacement.



(a) Absolute network (reference points) (b) Relative network (object points)

Figure 2.2: Relative network and absolute network [KAB10]

- *Relative Network.* All of the points in relative network are assumed to be located inside the monitored object [Che83]. These points are entitled "*object points*" and usually need to be monitored frequently to assess the behaviour of the monitored object in a short term, as shown in Figure 2.2(b). Compared to applying the absolute network, the distortion assessment using a relative network is more complicated, because all of the points are influenced by strains in the object and by the geometric changes of distinctive parts in the object with rigid translations and rotations [NMV+03]. The consecutive analysis of these object points is to generate the deformation model for identifying the cause of the deformation, for example, *extension* or *shearing strains*.

Absolute network and relative network can be combined together to offer both global and local information for detecting the geometric changes of the object under investigation. Figure 2.3 shows an absolute network (red line) and a relative network (blue line) on on a simulated deforming-embankment along a railway line. In order to build these two networks, reference points (red triangles) and object points (blue triangles) are settled separately outside and inside the monitored parts. These networks need to be designed and located before the deformation measurement; and points deployment is a labour intensive work with high cost [CRH09], especially for the deployment of reference points. For example, reference points placed on the earth are better installed on the solid rocks underground. The reference points should be above ground for certain height (e.g., 1.5 m) [WS09b].

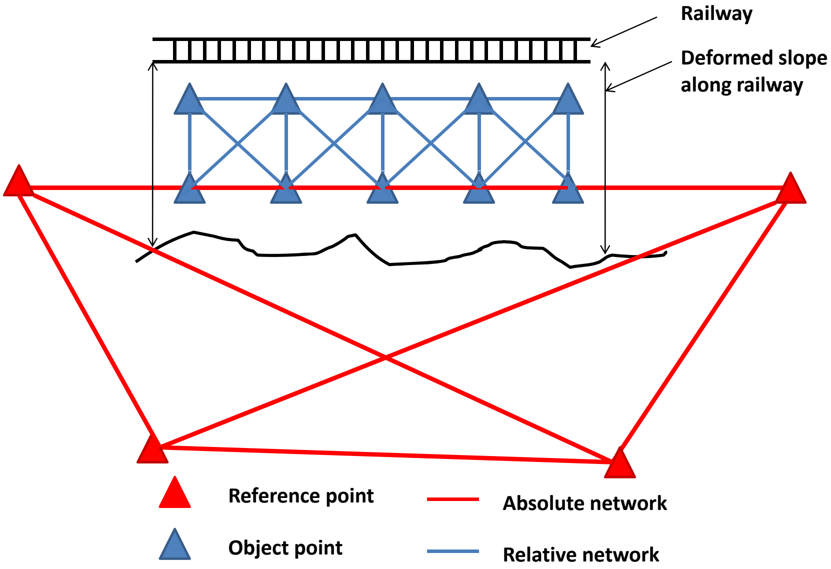


Figure 2.3: Absolute and relative network in a monitored region

Classical Geodetic Methods Summary - After the monitoring networks have been designed and set up, classical geodetic methods are applied to acquire the observations of the monitoring networks; such observations can be angle and distance measurement, and point coordinates, etc. Based on the comparison of the observations and the statistical evaluation of their quality, the geometrical changes can be estimated. With the development of the geodetic techniques, there are different types of instruments available for deformation measurement. We present a short summary for each geodetic method.

- *Theodolites and Electronic Distance Measurements* (*EDM*). Theodolites and EDM were frequently used as the primary surveying methods. Theodolites are specially designed for measuring angles in the horizontal and the vertical planes [WP97], while EDM can measure distance with relatively high accuracy [R9̈6]. Theodolites and EDM have been widely used in slope measurement [RRSJACBG02]. These methods can identify the magnitude and the direction of distance changes for each point in the geodetic control network; and additionally they plot these displacement vectors on maps to locate the deformation [HW86, Rob79]. A three-dimensional coordinate system can be built by two or more electronic theodolites with a computer together; and such system can be used to calculate the coordinates in real time. This system can be applied for the highest precision positioning and deformation monitoring surveys over small areas [SF90, Sav03].

- *Total Station.* Total station can be considered as an electronic theodolite combined with an EDD, to provide more convenient angle and distance measurement for operators [Fie04]. With the advantage of combining angle and distance measurement, electronic total stations have been largely applied in many surveying applications. Therefore, total station became a quite efficient alternative of theodolites and EDM instruments. For the measurement with theodolite and total station, the high-quality visibility are required between the stations on the monitoring points [SM86]. Of course, such restrict visibility requirement affects the location of the reference or object points, which in turn limits the geometry of the geodetic control network.

- *Global Navigation Satellite System* (*GNSS*). GNSS is a satellite system for providing geo-spatial positioning in a global coverage [GGE09]. Current GNSS systems include GPS ("Global Positioning System") from the United States [ER02], GLONASS ("Globalnaya navigatsionnaya sputnikovaya sistema or Global Navigation Satellite System") from Russian [MVM09], Galileo positioning system from the European Union [NLSH12] and Compass navigation system (extended from Beidou navigation system) from China [Bha10]. GNSS surveying technique can perform automatic and continuous measurement (7 days a week, 24 hours per day), under various weather conditions. The accuracy and reliability of GNSS positioning mainly depend on the number and geometric distribution of visible satellites [Wan02]. Compared to classical geodetic methods, GNSS technique does not require high inter-visibility between the points of the network; therefore, GNSS allows more flexibility in the optimisation and design of the monitoring network [Ver02, EEA04]. In addition, the development of various efficient real-life GNSS products (e.g., modern receivers with the ability to use GPS and GLONASS satellite) further extends the flexibility of geodetic network design.

Accuracy Analysis of Different Geodetic Techniques - For different geodetic measurement methods, the observation accuracy offered by these techniques is one of the most important aspects for evaluation of the quality in deformation monitoring.

- *Accuracy of Theodolites and EDM.* In the last thirty years, the development of angle measurement by theodolites depends on the automation of the readout systems, particularly on the horizontal and the vertical circles. From the accuracy point of view, the electronic theodolites have no significant improvement compared to optical theodolites. Regarding the distance measurement by using EDM instruments, the accuracy can be expressed in terms of standard deviation in a general form (see Equation 2.1).

$$\sigma_{EDM} = a + b * S \tag{2.1}$$

where "a" consists of zero correction (including errors of the phase measurement and calibration errors), and the additive constant of the EDM instrument and the reflector; "b" is a scale factor, due to the uncertainty in the determination of the refractive index and errors in the calibration of the modulation frequency. Typically the value of "a" ranges from 3 mm to 5 mm. For the most precise EDM instruments with a high modulation frequency and high resolution (excluding the more recent laser trackers), the value of "a" can reach from 0.2 mm to 0.5 mm [U.S02].

- *Accuracy of Total Station.* Total stations with different models can have different accuracy, range, capability to support automatic data collection and online processing [Lut02]. The combination of a digital theodolite, an EDM and a microprocessor records angle and distance measurements directly, which can reduce many reading errors. These reading errors exist in the operation procedure of optical theodolites and traditional EDM. Modern total stations (e.g., Trimble S8 total station) can obtain $1''$ angle accuracy and 2500 m distance measurement with 1 cm accuracy assisted by long-range functionality [Tri12].

- *Accuracy of Global Navigation Satellite System (GNSS).* GNSS can have different types of errors. For example, the GPS accuracy can be affected by many sources, such as signal propagation errors, receiver related errors, receiver system noise, and satellite related errors. The accuracy of GPS positioning can be expressed by using the variance of the horizontal components and the vertical components. Usually, the accuracy of the vertical components is 1.5 or 2.5 times worse than the horizontal components [Sav03].

To summarise, many classical geodetic techniques for high precision measurement are available for deformation measurement alone or combined together. Several advantages of these classical geodetic techniques can be concluded as follows: (a) offering high accuracy positioning of discrete monitoring points; (b) supplying redundant observations to evaluate the quality of measurement; and (c) providing a global information of the status of the deformation.

On the other hand, these classical geodetic methods also have some obvious drawbacks. For example, (a) they require a lot of redundant works and hereinafter they are expensive for continuous monitoring; (b) the location and condition of the monitored object limit the application of these techniques. These related instruments are deployed on the object with possible deformation, and the deformation could induce potential safety hazards to the operators. In some cases, the objects under control are even inaccessible because of the high risk level of these sites; (c) these methods are based on the "single point measurement" technique. They only provide low-density sampling data (e.g., discrete points) of the object under investigation.

2.3.2 Remote sensing techniques

Besides classical geodetic methods, remote sensing techniques also play an important role in deformation monitoring. The remote sensing source datasets used for deformation monitoring can be classified into *satellite images*, *SAR/InSAR images* (Synthetic Aperture Radar/Interferometric Synthetic Aperture Radar), *photogrammetry datasets*, and *LiDAR datasets*, according to remote sensor types [MHG05]. The following subsection discusses the remote sensing techniques for deformation measurement from four perspectives based on this sensor type.

Satellite Images - Satellite images are the datasets collected by optical or infrared sensors, which have become one kind of important datasets to determine Earth surface changes. With the development of high resolution geometric sensors and even very high resolution geometric sensors panchromatic images, there are increasing studies on satellite imagery for deformation monitoring. These studies are typically for some concrete applications such as landslide monitoring and natural hazard assessment.

- *Optical/infrared sensors.* At the beginning, satellite images did not work successfully in landslide monitoring because of the insufficient spatial resolution imaged from spaceborne earth observation systems [SVW96, MSVW96]. For example, satellite images with a spatial resolution from 10 m to 30 m can only be used for determination and identification of some terrain information related to landslides, e.g., vegetation types, land use and soil humidity [CWC04, HC91, LLHY02, MSVW96, ZLLX02]. Afterwards, the development of sensors on satellites can produce high resolution images. Some earth observation satellite such as Ikonos, Quickbird, SPOT-5, IRS CartoSat-1, WorldView can provide good quality images with spatial resolution less than 1 m. Some examples are listed as follows: (1) Accurate digital surface models are generated using IKONOS stereo imagery and are applied in tectonic deformation modelling [ZCF07]; (2) Several large landslide cases, located in the North Island of New Zealand, are mapped from Quickbird images with 0.6 m spatial resolution [VLL09]; and (3) Identification of landslides during 2005 northern Pakistan earthquake in the northwestern part of the Himalayas is studied using SPOT 5 stereo images with a spatial resolution of 2.5 m [SHF$^+$06].
- *Deformation Measurement by Satellite Imagery.* Multi-temporal and high resolution satellite imagery have been widely considered as an effective complementary dataset to traditional observations [Her03]. Many researchers have studied a large amount of applications of satellite images for deformation monitoring. Nagarajan et al. presented a method to integrate spatial and temporal multi-layered information for landslide hazard assessment [NMRK98]. In this study, temporal data from Indian Remote Sensing Satellite (IRS) are used to interpret land coverage and land use information. In addition, Delacourt et al. demonstrated a method that combines aerial photographs and Quickbird images for deformation measurement [DACV04]. This investigation is mainly used to monitor landslide displacement.

SAR and InSAR Images - SAR & InSAR Images are another kind of datasets acquired by active remote sensing techniques. In the following paragraphs, we firstly introduce the SAR technique to capture images; then present the applications of SAR and InSAR techniques for deformation monitoring. Finally, we summarise the pros and cons of SAR and InSAR.

- *Introduction of SAR/InSAR.* SAR is a form of radar device for recording the phase (time delay) and amplitude (energy intensity) of radio waves reflected by the Earth's surface [GVG96]. SAR can be implemented as follows: an antenna is mounted on a moving platform such as a spacecraft or an aircraft, which transmits periodically radio waves at wavelengths from a metre down to millimetres. A part of the radio waves are backscattered from the surface of the Earth and received successively at the different antenna positions. These waveforms echoes are coherently detected and recorded to construct the SAR images [Hei04]. Traditional SAR remote sensing only utilises the amplitude information of a single SAR image [KM98]. Interferometric SAR is a technique for measuring phases from successive SAR images to infer range changes of the same surface. This technique can be used for subtle changes detection with reasonable scale, accuracy and reliability [CFN$^+$03]. Therefore, InSAR provides the chances to displacement detection of the Earth's surface. It is worth noting that short-baseline (zero to tens of metres) interferograms are suitable for displacement detection, while long-baseline (up to several hundred metres) interferograms are utilised for constructing Digital Elevation Models. More explicit explanations are presented in [Smi02].

- *Deformation Measurement by SAR/InSAR.* InSAR technique has the advantage of detecting large areas with centimetre-scale displacement. Such areas can be tens of kilometres. Therefore, InSAR technique has been demonstrated successfully in a lot of applications of deformation monitoring such as subsidence, landslides and erosion. For subsidence detection, InSAR has been successfully applied in many cases like flat and dry terrains. For example, a map of subsidence in the Belridge and Lost Hills oil fields (100 km west of Bakersfield, California) is generated from InSAR pairs [DK97]. Similar results such as subsidence rate as high as 400^+ $mm/year$ is also detected in the same area [FBG98]. More examples of subsidence detection are presented in [AGB$^+$99, CCDMW$^+$08, AZGK09]. For landslide monitoring, the La Clapière landslide, located in Southern France on the left bank of the Tinée river, is monitored by six different interferograms driven from ERS-1 SAR images. This study demonstrats a downhill movement from the top to the bottom of the landslide up to 30 mm/day [FAD96]. The same area is also studied using the same ERS-1 datasets in [CMK96]. Another case study of Itaya landslide in Japan is analysed by three interferograms constructed from JERS-1 SAR images [KY00]. These SAR data are collected from June to October, 1995 and are processed to show the extension of the displacement field. However, good geometrical features in this case can not be recovered. In addition, the La Valette landslide, located in the Ubaye valley (southern French Alps), is analysed by 15 differential interferograms realized from ERS-1 and ERS-2 satellite radar images. Investigation using SAR interferograms indicated that a sector in the eastern part of the landslide had small velocity ($<$ 10 mm/day) [SDA03]. More case studies of landslide monitoring are introduced in [CW04, CCB$^+$05, GMA$^+$09, CSR10, COCC$^+$12]. The limit of InSAR technique is that it can observe a phase difference caused by the ground motion parallel to the satellite line of sight vector, which means that the components of vertical motion and horizontal motion parallel to the plane of the line of sight cannot be separately resolved.

- *The Pros and Cons.* As a promising deformation technique, SAR and InSAR have advantages as follows [Rot04, Pag04, Sin02]: (1)High resolution images (e.g., less than three metre resolution stereo imaged from C-band SAR) can provide more geomorphologic information on the slope. This can produce higher quality landslide maps. (2) The usage of techniques such as Permanent Scatterers (PS), DInSAR and InSAR can increase the accuracy of slope stability detection. (3) Long term access of SAR data can assist in temporal analysis of slope movement. Nevertheless, the disadvantages of this remote sensing technique are also quite obvious, as listed in the following paragraph [CMK96, CEO01, CCE+04, Rot04]: (1) Limit types of displacement can be measured, e.g., InSAR only detects displacement in direction of the radar illumination (in the satellite's line of sight). (2) Interferometry can be affected by vegetation situation on the surface, especially with dense vegetation. (3) Rapid landslide deformation can cause loss of radar coherence, especially during successive data collection. (4) Detection ability of InSAR is constrained by the location and speed of landslide. For instance, rapid slope movement like rock/debris fall on a steep slope or narrow valleys can not be detected. (5) SAR sensor technique issues can cause the problems of data unavailability.

Photogrammetric Datasets - Photogrammetric methods have been extensively used for deformation monitoring of terrain surface and structures during the past three decades [Coo84, Maa98, KÖ0, MH06]. Generally speaking, photogrammetry is one kind of methods for measuring an object from two or more known points by a camera with known interior and exterior orientations. Coordinates of some points on this object are calculated according to the geometrical relationships between the optical rays from the image to the object points [KHK07]. Based on the platform where cameras are mounted, photogrammetry can be classified into *aerial photogrammetry* and *terrestrial photogrammetry.* In the following paragraphs, we present photogrammetry for deformation measurement from these two aspects:

- *Aerial Photogrammetry.* Aerial photographs for topographic mapping was considered as a standard technique in land surveying; and became a proper approach for the measurement of Earth surface mass movement [CC88, MG03, CDBA03, DGK12]. The possibility of high-precision and multi-temporal image analysis for the prediction of glacial hazards is shown by real case studies in the Swiss Alps, e.g., detection of the increasing risk of late-outbursts and related debris-flows from a 25-year monitoring observation series [KHG97, KÖ0]. Additionally, aerial photogrammetry has been applied into the studies of ground movements in large areas [MBC+03, ZBB+06]. For example, the evolution of a deep-seated gravitational slope deformation, located in the Northern Apennine range (Italy), is investigated by photogrammetry combined with GPS observations, for detecting the displacement vectors of 293 points in the landslide [BCFZ08].

- *Terrestrial Photogrammetry.* Terrestrial photogrammetry shows great potential to solve measurement tasks in structural deformation monitoring [Maa98, SPT$^+$03, MH06]. Four different photogrammetric methods are identified and compared respectively; they are a single camera, controlled stereomodels, resection/intersection procedures, and the bundle adjustment approach [Coo84]. The first three methods are used when the detected deformation is larger compared to the photogrammetry accuracy; the last one is applicable when the deformation to be measured are of the same magnitude as the accuracy of the photogrammetric observation. Many case studies of deformation monitoring using terrestrial photogrammetry are conducted and some examples are listed as follows: (1) a pilot study of dam monitoring is performed by a photogrammetric method with a digital high-resolution still video camera Kodak DCS200 and with a metric camera Wild P31 [KM95]; (2) a case study in monitoring slope displacements is applied in an open lignite mine, and a series of products such as maps, orthoimages and volume calculations are presented [SPT$^+$03]; (3) a case study of deformation measurement during construction material testing is illustrated, and the image-based algorithms can integrate and even replace traditional sensors currently used in several laboratories [BS11].

There are several advantages in using photogrammetry for deformation measurement. For example, this technique can significantly reduce the time of fieldwork, and becomes a good candidate for urgent monitoring work. Moreover, this technique provides three-dimensional coordinates simultaneously with an unlimited number of points on the monitored object. Some digital products such as Digital Terrain Models, vertical terrain changes, and orthoimages can be created by photogrammetry. With the progress of photogrammetric techniques, the accuracy improvement in determining point positions can enhance their capability in deformation measurement.

LiDAR Datasets - LiDAR data is yet another kind of important datasets that provide potential for deformation monitoring. Its applications recently become a hot research and engineering topic. From image point of view, LiDAR data can be considered as an extension of 2D images, i.e., the depth images. Compared with two-dimensional images, points of the LiDAR data have three components of positional information such as $\langle x, y, z \rangle$ in the coordinate framework. From the remote sensing perspective, LiDAR can be considered as an active remote sensing technique that uses laser lights at different frequency. It measures the temporal difference between emitted pulses and their received reflections, and reconstructs the location, height, speed and direction for an object [Pop09]. The history of using laser for remote sensing can trace back to the 1960s. After LiDAR becomes an effective geodetic mapping tool, applications of LiDAR for deformation measurement attract increasing interests recently. According to the mounted platform of LiDAR sensors, we present LiDAR technique from the following two aspects:

- *Airborne LiDAR.* In the initial sate, many LiDAR sensors are mounted on airborne platforms. LiDAR system can be categorised into two types: waveform LiDAR systems and discrete-return systems [She08]. Waveform LiDAR systems continuously acquire the returned waveform periodically in a large footprint (e.g., 10-25 m), while discrete-return

LiDAR systems collect only one or a few discrete echoes from a small footprint (e.g., 5-30 *cm*) [MHG05, She08]. LiDAR footprint is a crucial parameter to describe the size of laser sampling area, which is influenced by the scanning geometry and the scanned local topographic surface. LiDAR systems with large footprint, recording an average value to approximate a larger region, can be applied for forest mapping [MAH$^+$99, LCPH02, LTW$^+$03]. Whereas, LiDAR systems with small footprints can be used for high-resolution terrain surface mapping [ZEC$^+$03, WHBD08]. Airborne LiDAR can provide digital elevation models or digital surface models of large regions; and this is a revolutionary change for terrain analysis. However, for some specific area, Airborne LiDAR technique can not provide very high resolution data like millimetre level.

- *Terrestrial LiDAR.* Compared to airborne LiDAR, terrestrial LiDAR is a relatively new technique for geodetic mapping and measurement. Terrestrial LiDAR is initially used for detailed surveying of complex objects, e.g., design and manufacture of automobiles in industrial engineering [BVT92, CBS00, BR02]. During the last decade, terrestrial LiDAR is applied successfully in cultural heritage documentation which is essential for the protection and restoration of cultural heritage status [BHM01, KTL$^+$04, GVEHG04, PKA$^+$07, Yas07]. The advantages of terrestrial LiDAR (such as huge datasets with high accuracy and resolution) can assist in completing the recording of cultural heritage in three dimensions. It also affects the aspects in terms of new digital datasets management, representation and reproduction. Afterwards, applications in heritage digital recording attract attentions of researchers and specialists such as geologists and surveyors. Considering recent studies, applications of terrestrial LiDAR have expanded to many new areas and topics, such as 3D city modelling [Lem08], analysis of traffic accidents and road safety [Lem11], deformation monitoring [GS04, TGZG07, MC08, LB09, JOA$^+$10] and civil engineering setting [Mur08]. Development of terrestrial LiDAR, driven by resolution, accuracy, speed and operational requirement, shows great potential for deformation monitoring.

Complementary to traditional geodetic methods like EDM and total stations, remote sensing is new effective techniques for deformation measurement. Traditional geodetic methods can be considered as “single point measurement” whilst LiDAR is “surface measurement”. Additionally, photogrammetry can also produce 3D object surface measurement. From a point of view of hardware, novel digital large format aerial cameras produce high quality and overlapping images. From a point of view of software, computer vision has developed innovative algorithms in multi-image matching and can also utilise Graphics Processing Unit (GPU) to make the complex image matching algorithms more practical. Driven by both kinds of innovations, 3D photogrammetric point clouds are produced at sub-pixel accuracy at very dense intervals [LIP$^+$10]. Compared to these remote sensing techniques (e.g., space-borne sensors), LiDAR can be used for extra scenarios such as corridor mapping and precise measurement of concrete structures (e.g., steep cliffs). These measurements are quite complicated and are hard to be collected by other remote sensing techniques. The density of LiDAR points is quite high and good for deformation monitoring.

In this thesis, we propose a novel approach for deformation monitoring using point clouds theoretically, which can be photogrammetric or LiDAR point cloud datasets; whilst in the following experiments, we utilise terrestrial LiDAR for deformation measurement and focus on analysing the point clouds captured by terrestrial laser scanner. We provide more detailed explanations of terrestrial laser scanner in Section 2.5.

2.4 Deformation Analysis

Geometrical analysis of the datasets resulting from deformation measurement has been a hot research topic for a long time [Chr81, Cas88, HR01]; such geometrical analysis to estimate deformation is called "deformation analysis". The deformation analysis of an unstable object in essence is to estimate geometrical changes of this object during a given time period. These changes can be estimated by comparing the datasets sampled on this object between different epochs and identifying the differences of this object between two epochs. In terms of deformation measurement, the representation of the object is mainly the observations of the discrete data points (very low sampling data) captured by classical geodetic methods. Classical geodetic methods like GNSS set up some limited control points on/around this monitored object. The observations of these control points are considered as the sampling datasets; each sampling dataset is corresponding to one complete observation at one epoch; and these datasets are used to estimate the deformation of this object from one epoch to another. Therefore, *geodetic network design* consisting of the discrete points is a preliminary and important work to fulfil the requirements of the anticipated deformation, as already discussed in Section 2.3.1 of deformation measurement.

To give a concrete example, Figure 2.4 shows an unstable object *Obj* under monitored. To estimate the deformation of *Obj* between Epoch I and Epoch II, we build n control points (black triangles, i.e., from L_1 to L_n) which form a relative network on this object; based on this network, we can acquire the sets of observations, i.e., $\mathbf{L}^I$ at Epoch I and $\mathbf{L}^{II}$ at Epoch II.

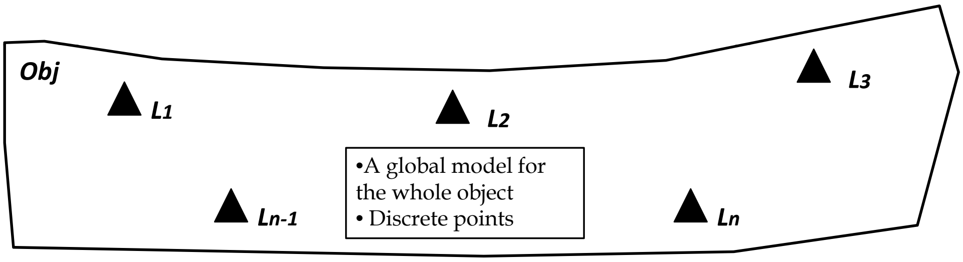


Figure 2.4: One unstable terrain object under monitoring using discrete points

Deformation analysis based on these discrete points can be summarised as follows: (1) *Analysis of the discrete points* is the examination of the observations in the geodetic network at each epoch; therefore this is also an essential step that can void illustrating the undetected errors which might cause error propagation in the subsequent procedures. (2) With the prior step of error cleaning via discrete point analysis, *deformation estimation* is the next step and the main focus in deformation analysis. Deformation estimation is to determine the geometrical changes from the geodetic network with global or local coverage at different epochs, and even develops a unified theoretical system for the geometrical analysis of the observations [Chr81].

2.4.1 Analysis of discrete points

Analysis of discrete points at each epoch is the adjustment of the geodetic network formed by the discrete points, including *network adjustment* and *statistical test of adjustment.* Based on the raw observations, the objective is to estimate the required information for the subsequent deformation detection and to identify the stochastic model of the adjusted observations.

Network Adjustment - The raw observations from geodetic methods might include different kinds of measurements such as *distances*, *directions*, *azimuths* and *coordinates.* In addition, these observations contain measurement errors including *systematic errors*, *random errors* and *mistakes.* The three types of errors are mixed together and affected the residuals in deformation estimation. In order to check the outliers in the measurement and assess the precision of the adjusted quantities, redundant observations are usually designed and obtained during the surveying procedure. Analysis of discrete points from the redundant measurements at each epoch is the preliminary task to remove the gross errors or outliers in the observations and to build the stochastic model of the adjusted quantities.

Statistical Test of Adjustment - After the adjustment of the geodetic network, the statistical test is performed to analyse the quality of the observations. The estimation residuals are a mixture of all error types, including systematic errors, random errors and mistakes [CCS90]. Therefore, a global model test is required to avoid the possibility that these undetected errors are interpreted as deformations in the subsequent procedures. Additionally, this global test can also evaluate the model assumption and to check whether the model confirms with reality.

2.4.2 Deformation estimation

After statistical test of geodetic network adjustment, the adjusted observations of the discrete points (e.g., usually the coordinates of these points) are obtained from the raw deformation measurements. Based on the adjusted observations $\hat{L}^I$ at Epoch I and $\hat{L}^{II}$ at Epoch II, a global deformation model $M(X)$ can be built to estimate the deformation for the whole object

$$\hat{L}^I - \hat{L}^{II} = M(X) + V \tag{2.2}$$

where $\hat{L}^I = (\hat{l}_1^I, \cdots, \hat{l}_n^I)$ is the adjusted coordinates of these discrete points at Epoch I, $\hat{L}^{II} = (\hat{l}_1^{II}, \cdots, \hat{l}_n^{II})$ is the adjusted coordinates at Epoch II, X is u unknown deformation parameters $X = (x_1, \cdots, x_u)$, and V is the residuals $V = (v_1, \cdots, v_n)$. $M(X)$ is the global model of this whole object to estimate the unknown parameters X, which offers a global picture of this object under monitoring and provides a global and uniform estimation.

In order to get the optimal estimation of parameters X, this can be reformulated as the following *optimisation* problem,

$$\begin{aligned} &\underset{x_1,\cdots,x_u}{argmin} \quad f(V) \\ &subject\ to \quad \hat{L}^I - \hat{L}^{II} = M(X) + V \end{aligned} \tag{2.3}$$

where V is residuals between the real deformation ($\hat{L}^I - \hat{L}^{II}$) and the deformation estimated by the model $M(X)$ with the parameters X; the cost function $f(V)$ is determined by residuals.

- The cost function $f(V)$ can be defined by several estimation methods. *Least-squares estimator* is one of the well known estimators and defines cost as the sum of squared residuals, i.e., $f(V) = \sum_{i=1}^{n} v_i^2$ [Cas88]. For least-square estimation, a Gaussian distribution is assumed for the observation errors. Outlier are not considered in this distribution, so least-square estimation method is very sensitive to gross errors. *Re-weighting functions* $W_{i+1} = W_i\, f(V)$ is one way to reduce the effects of outliers by computing new weights from the residuals of the previous iteration. Modification of weights in each iteration is different from conventional lest-squares estimation. More *residuals models* are introduced to reformulate the cost function. For instance, the cost function $f(v) = \rho(v_i)$ is an M-estimator that reduce the influence of outliers such as Tukey estimator and Huber estimator [LF05].

- The deformation model $M(X)$ can be defined based on the observation differences or based on displacement approach. There are three general deformation models of $M(X)$ such as *polynomial approach*, *rigid transformation*, and *strain model*. Polynomial approach is a well-applied displacement function and provides the approximation of the displacement field [HR01]. Rigid transformation model is another widely-used representation of the whole object's displacement. Strain model is also applied in numerous applications of deformation analysis, especially in crustal movement studies. In strain model, deformation is assumed to be continuous over the entire object. Detailed formal descriptions of these global models are presented in Section 3.3.

Traditional deformation analysis is mainly limited to low sampling datasets such as discrete points scatted on the monitored region. Based on such sparse data, it is possible to detect the deformation parameters of the whole monitored object approximately using a global model. However, this is not sufficient to obey the real nature that the monitored object usually has a homogeneous mixture of deformation. In real-life cases, different partial areas of the object under investigation have distinctive deformation. Driven by the development of equipments, terrestrial LiDAR provides the high density datasets, entitled 3D point clouds, for the monitored object. Compared to deformation analysis based on discrete points (e.g., around 20 points on the whole object), 3D point clouds with the high spatial resolution as several millimetres provide new directions and challenges for deformation analysis.

2.5 Terrestrial Laser Scanner

Compared to "single point measurement" techniques, LiDAR (Light Detection And Ranging) technique initiates a new spatial dimension for data acquisition by surveying technique. By acquiring a large number of points (entitled "point clouds") on the surface of an object, laser scanning technique provides the abundant geometrical information of the surface, which offers the powerful ability in the application of architecture and archaeology [Lev99]. With the development of LiDAR technology driven by hardware resolution, it becomes one of the promising surveying techniques for field measurement and monitoring [AJOV09]

2.5.1 Working principle

Terrestrial laser scanner (TLS) incorporates a range measurement system, which requires deflection mechanism for pointing the laser beam in two directions [KHK07]. This is because a TLS is usually stationary during one scan of the object. To better understand working principle of terrestrial LiDAR, we take one kind of Riegl TLS as a concrete example. This equipment contains two main components, i.e., "a range finder electronics unit" and "a polygonal mirror element", marked as ① and ③ in Figure 2.5. Firstly, the range finder electronics unit emits a pulsed laser beam to the polygonal mirror element. Secondly, this laser beam is reflected off the mirror surfaces and scan through a vertical angle ζ, because the mirror element rotates at a relatively high speed. These two steps compose a scan with a vertical angle ζ, called *one profile*. Afterwards, the upper part of the laser scanner, marked as ④ in Figure 2.5, rotates through a small angle $\Delta\alpha$ in order to sample the neighbouring ζ profile, until a full horizontal circle has been covered. An external Laptop (or PC) with specific software is usually connected with the TLS for its operation and data storage.

Coordinates of all of the points in a point cloud are calculated by the measuring the deflection of the rotating mirrors. Intensity of each point is achieved by recording the strength of the return signal, which is the reflectance measure and the spectral characteristic of the object. Colour information of the object can be acquired and registered automatically by the laser scanner with a camera embedded inside. For a laser scanner with a camera mounted outside, the registration between the images and the point clouds is performed additionally.

Usually, the coordinate frame of the point clouds is a relative frame based on the terrestrial laser scanner shown by Figure 2.6. The origin of this frame is the centre of laser scanner and the coordinate frame is a right-hand coordinate system. The point clouds of one object, captured at different stations, need to be registered together to represent this object. Some referenced points can be used to transform the relative coordinate frame into a local coordinate system or a user-defined coordinate system.

2.5.2 Classification of TLS

Various terrestrial laser scanners are available in current market. However, categorisation of TLS is non-trivial. Until now, there is no general criterion to classify TLS. Basically, there is no one laser scanner suitable for all applications. Different TLS have distinctive characters and

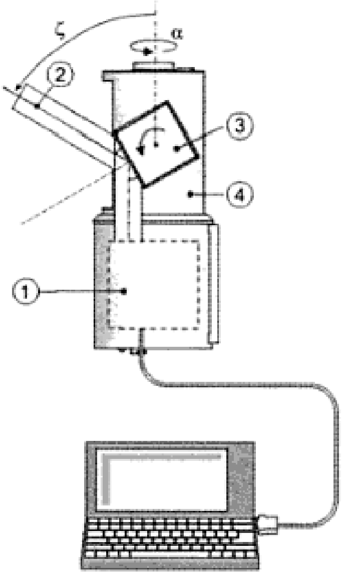


Figure 2.5: Working principle of a Riegl TLS [KHK07]

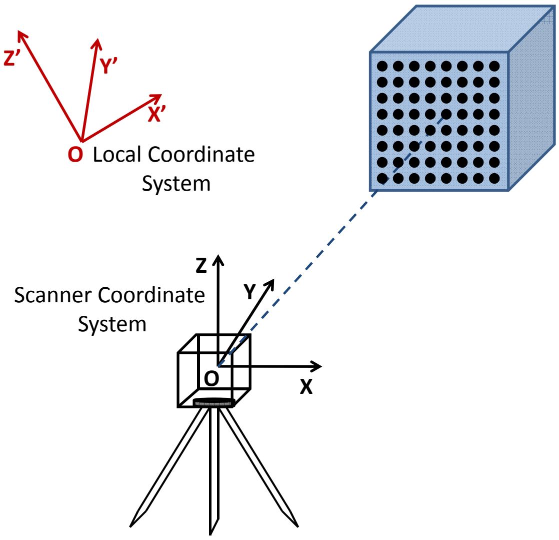


Figure 2.6: Coordinate system of point clouds

have their specific utilisation in different domains. A summary of the classification of TLS is concluded as below, according to the former work done in [FM04, SI04, Sta03, Lic10].

Classification Based on Working Principle - Based on the principles of distance measurement system, terrestrial laser scanners can be classified into three types - based on *time of flight*, *phase measurement* and *optical triangulation*, respectively [SI04].

- *Time of flight.* This is the most popular measurement system of TLS. Based on this technique, TLS can achieve the maximum range up to several hundred metres, even beyond one kilometre.

- *Phase measurement.* This is another common technique for measuring medium range. Using this technique, the maximum range achieved by TLS is about one hundred metres.
- *Optical triangulation.* This is the complementary technique used for close range laser scanners. This kind of instruments can range up to several metres, which are mainly used indoor like reverse engineering and archaeology.

Classification Based on the Technical Specifications - Besides the classification based on the measurement principle, the technical specifications of individual systems are also interesting. For instance, *the field of view* is chosen as a criterion for TLS categorisation[Sta03]:

- *Camera scanner.* Both the vertical and horizontal view of camera scanner are limited, e.g. $60°(\textit{horizontal}) \times 60°(\textit{vertical})$. This type is suitable for a view from outside onto the object.
- *Hybrid scanner.* Only the vertical view of hybrid scanner is restricted for the use of mirrors, e.g. $60°(\textit{vertical})$. The horizontal view has no limitation and can achieve 360°.
- *Panorama scanner.* The tripod of the instrument bounds the field of view of panorama scanner. This type can be used for indoor applications, like the digitisation of chemical plant.

In addition to the field of view, TLS can be classified according to several other important technical performances, such as *the range of TLS*, *the deflection system*, *spacial resolution* and *the combination with other devices* [FM04].

2.5.3 Calibration of TLS

Terrestrial laser scanner calibration is an important procedure to guarantee quality of data captured by a TLS. Therefore, a TLS needs to undergo a strict calibration schedule prior to operation. Self-calibration of TLS is investigated recently by many researchers [Lic07, CTL08, Sch09, MPR$^+$09, Lic10, Res10]. It is to identify the systematic errors inherent to modern TLS instruments which can be modelled mathematically to improve both the precision and the accuracy of the acquired point clouds [Sch09, Lic10]. This procedure is a user self-calibration routine, which can avoid the laboratory calibration by manufacturer, without high-cost and long-time consumption. Self-calibration of TLS demands a large set of observations of a TLS from different positions and various orientations. Depending on these observations, calibration parameters are estimated, including the scanner position/angular/orientation elements and the target parameters. Based on the type of target used, self-calibration approaches can be categorised into signalised point based method and planar feature based method [CLG11].

- *Calibration Parameters*. The calibration parameters can be categorised into two groups - physical parameters and empirical parameters [Lic07]. The physical parameters includes the zero and scale errors of the laser rangefinder, the collimation and horizontal axis errors, and the vertical circle index errors caused when the TLS can not get the theoretical state. The physical parameters have a physical interpretation based on the properties of TLS, and this is similar to total stations. For example, TLS systems have three fundamental axes comprising *trunnion axis*, *vertical axis* and *collimation axis*. Theoretically, these three axes should intersect at a unique point and be orthogonal with each other. Therefore, physical parameters (as systematic errors) can be removed. For empirical parameters, they may stay in the observational residuals, as they do not have such properties in the physical parameters.

- *Single Point Based Approach* For single point based TLS self-calibration, the underlying mathematical model is the rigid object transformation from the object coordinate space to the scanner coordinate space [Lic07, Res10, Sch09]. In this model, exterior orientation parameters (EOPs) are used to describe the orientation of scanner in object space. Lichti presents a free network adjustment method to estimate the exterior orientation parameters, object point coordinates and calibration parameters simultaneously [Lic07]. This method is applied to calibrate several TLSes including Faro 880 [Lic07], Trimble GS200 [CTL08], Trimble GX [CLT10]. The results show the significant improvement achieved by this method. To further improve the quality of TLS self-calibration, a unified method is proposed to decrease the parameter correlation in least-squares adjustment [Res10].

- *Planar Target Based Approach*. Compared to single point based TLS self-calibration which needs to measure a large number of single points, planar target based approach is less labour intensive. By using planar targets, extraction of these features can be highly automated. Similar as point based approach, exterior parameters, plane and calibration parameters are estimated by a combined least-squares method that minimise the distance between points and their corresponding planes with a constraint condition [BL07]. This approach has been used to test FARO 880 with simulated and real data, and demonstrates the improvement of observational precision of the self-calibration residuals' standard deviation. A continuous piecewise linear correction function is suggested by Molnár to replace the harmonic and polynomial functions [MPR$^+$09]. This correction function does not require assumptions on the shape of the calibraiton However, it requires more parameters in the formulation. More experiments of planar based approach are in [GRG04, DNPM08].

2.5.4 Data acquisition via TLS

There is no standard rule for data acquisition using TLS, especially in the environmental field. Different procedures of data acquisition are established according to the situation of the site under control [GS04]. Nevertheless, the general workflow of data acquisition via TLS includes the following main technical aspects: *accuracy requirement*, *preliminary site investigation*, *positioning of TLS* and *data collection*.

- *Accuracy requirement*. The original point clouds collected by TLS with certain accuracy and resolution are required to monitor a given environmental site. In experiences, the resolution of the original data is much higher than the resolution of the final derived topographic product. The data redundancy is frequently used to guarantee the proper quality as user expected. The defined accuracy and resolutions decide the setting of sampling parameters for TLS in data collection.

- *Preliminary site investigation.* The collection of related documents about the monitored site is *a priori* knowledge. Field exploration is yet another basic step to design the survey plan and process some preliminary operational issues. For example, some ground control points are designed and set up either inside or outside the monitored site according to its underlying environmental condition.
- *Positioning of TLS*. After the investigation of accuracy requirement and the investigation of the monitored site, we need to explore and fixe several issues about TLS positioning, e.g., where to set up TLS and how many scans for each station. Several issues need to be concerned here. For example, the distance between TLS and the monitored object affects the sampling resolution in data capture, and the scan region needs to be specifically selected for the following data registration from different stations.
- *Data collection.* Based on the previous design hypotheses (e.g., accuracy requirements, TLS positioning), some pre-scans can be applied to check the scanning coverage and resolution of point clouds. This can be further used to validate the previous design hypotheses, and maybe hypotheses need to be further re-designed. Based on a set of practical tests, we find the most suitable design hypotheses, and collect the final point cloud dataset.

2.6 Point Cloud Processing

After point cloud datasets collection, point could processing is the fundamental procedure to extract useful geo-information as well as deformation. Point cloud processing has recently become a hot research topic in many domains, not only in environmental fields such as geomatics, structural engineering and civil engineering, but also in in computer science like computer vision, and computer graphics and geometry. In this section, we firstly provide related works about point cloud processing such as *point cloud registration*, *point cloud segmentation*, and *3D object reconstruction via point clouds*; secondly, we present the application of point clouds in deformation monitoring.

2.6.1 Point cloud registration

For one environmental site under control, one station of TLS is usually not sufficient to cover the whole monitored region. Therefore, several stations are used frequently to capture the whole region. In such case, registration of 3D point clouds is a fundamental and preliminary process to put the point clouds from different stations with different coordinate frames into the same frame [GS04]. To speed up registration, relevant point cloud indexing methods can be used.

- *Coordinate frames.* Like the coordinate frame in traditional photogrammetric approach [STH80, Bro92], the coordinate frame of point clouds is a basic and fundamental aspect to represent and understand point cloud datasets. There are two types of coordinate frames (or called "coordinate systems"), i.e., *a scanner coordinate system* and *a local coordinate system* (see Figure 2.6). The point clouds captured by each scan are usually calculated based on a coordinate system defined by a TLS station, and this is a scan coordinate system. Different TLS stations have their own scan coordinate systems. These coordinate systems need to be registered into the same coordinate system, e.g., a local coordinate system, which is usually pre-defined according to the monitored object or is the particular coordinate system requested for the monitoring objective. In the literatures like [GS04], three kinds of coordinate/reference systems are explicitly defined, i.e, the intrinsic reference system (IRS) of each scan, the project reference system (PRS) that is shared between more than one scan, and the ground reference system (GRS) for sharing all scans. Therefore, IRS is a scanner coordinate system, PRS can be the shared local coordinate system between two scans, while GRS is the local coordinate system for all scans.

- *Data registration methods.* Data registration is to match different coordinate frames, or to transfer one scan frame to a local/global frame. Data registration methods mainly have two types, i.e., *registration based on ground control points* and *registration based on overlap points.* The first type of methods need some fixed points on the monitored site and these points are called "ground control points". For registration of point clouds from two scans, four common ground control points in two scans are required to compute the transformation parameters like *translation* and *rotation.* These control points are usually pre-designed in the procedure of TLS positioning. The second type of registration methods are based on the point clouds of the overlap areas from two scans. Well-known second type registration methods include Iterative Closest Point (ICP) and Least Squares 3D Surface Matching (LS3D). ICP is based on the search for pairs of nearest points in two datasets and estimates the rigid object transformation that aligns the data [BM92, Zha94]. LS3D is also used to calculate the transformation parameters of one search surface with respect to a template surface by minimising the sum of squares of Euclidean distances between the surfaces [Akc07]. Both ICP and LS3D methods belong to the automatic registration methods of point clouds.

- *Point cloud indexing.* In point cloud registration using ICP, it optimises the mean-square distance objective function and finds the minimum solution. Such process requires expensive computation and leads to slow convergence, especially for nearest neighbour searching in each iteration. Therefore, some point cloud indexing techniques are requested to improve the nearest point searching between two point cloud datasets. K-d tree (k-dimensional tree) is a special case of binary space partitioning tree [Ben75] and has been well applied in ICP method to improve searching efficiency [BM92, GG01, NLH07]. K-d tree can be used to accelerate searching computation by searching binary tree, instead of closest point linear search [Sim96]. Many enhanced k-d tree methods are developed. For example, approximated k-d tree is presented using depth-first search of the k-d tree structure, replacing a backtracking searching [GY03]. This extended method improves the computation efficiency but with the cost of reducing the accuracy of correspondence searching. Another modified method is cached k-d tree, which applies caching techniques to contain most temporarily nearest tree nodes [NLH07]. Octree is another tree structure frequently used to partition a three dimensional space by recursively splitting this space into eight elements [JT80, Sze96, SED07, RKR$^+$08]. R-tree is yet another spatial access technique and is a tree data structure for indexing multi-dimensional spatial information [BKSS90, MNT05, SSV07, GZZ$^+$12].

2.6.2 Point cloud segmentation

After point cloud registration, all of the point cloud datasets are aligned in the same coordinate system. Then, point cloud segmentation can be used to assist in extracting 3D interesting spatial features from point clouds. Usually, a scanned object has several components with different geometrical features. Generally speaking, point cloud segmentation is the classification of point clouds with different geometrical features, by grouping adjacent points with uniform characteristics together [Wan11]. There are a large amount of point cloud segmentation methods been developed recently, and can be categorised into three groups: *edge-based methods*, *region growing methods* and *hybrid methods* [WKW$^+$02, LX08, Sam10].

- *Edge-based methods.* The objective of edge-based point cloud segmentation is to detect the boundaries of components from the scanned point clouds and to form partial regions using detected edges [WB94, HM01]. This can be achieved by distinguishing discontinuities of the object using geometric properties, e.g., normals and curvatures [FMN87, YL99]. Yang and Lee present an automatic edge-based method to identify edge points from the estimated local surface curvatures, based on a parametric surface approximation [YL99]. Sappa and Devy introduce a fast edge-based segmentation method with two steps: firstly, a binary edge map is generated only considering two orthogonal scan lines; secondly, boundaries are obtained from the built binary edge map using a strategy of contour detection [SD01]. Ding et al. use RHT (Randomised Hough Transform) based plane detection which is robust to noise [DPHW05]. Wang and Shan segment point clouds directly based on similarity measures, supported by fast nearest neighbourhood search. This method is utilised to segment the building boundary and roof planes [WS09a].

- *Region growing methods.* Region growing methods are mainly used for grouping neighbourhood points with similar geometric features together. The general procedure of region growing methods can be summarised as follows: (1) start from a selected point; (2) calculate the geometric features of this point; (3) compare these features with adjacent points; (4) form regions by grouping points around the seed point based on similarity measures, e.g., proximity, slope, curvature, and surface normals [Sam10]. Rabbani et al. segment point clouds using a smoothness constraint by searching smoothly connected areas from point clouds [RVDHV06]. In this method, however, few parameters are required to find a suitable segmentation, balancing under-segmentation and over-segmentation. Chen and Chen cluster point cloud datasets based on normals and point locations [CC08]. Confidence rates of points are used to enforce fast clustering and to identify planar regions, which are affected by point location. Confidence rate might be low when points locate at the discontinuity of surfaces, but high for points in a smooth surface. Ning et al. use the residual of plane fitting to separate a building from architectural point clouds, as well as concrete architectural elements like walls and windows [NZWJ09]. In summary, the main challenges in region growing are (1) how to choose seed points, (2) how to control grow process, and (3) how is the algorithm robustness to noise and parameters.

- *Hybrid methods.* Hybrid segmentation methods are the combination of edge-based and region growing approaches [YL89]. Ghosal et al. present an approach consisting of two stages: range growing and edge-based method [Gho93]. Firstly, region growing segmentation is obtained by a surface feature-based clustering of point clouds. For the over-segmented areas produced in the first stage, the second stage is to use edge information to merge neighbourhood areas together as the final results. Checchin et al. propose a similar dual-step segmentation approach [CTA97]. This approach includes over-segmentation based on RAG (Region Adjacent Graph) and the refinement of over-segmented regions using a surface-based merging technique. Compared to previous methods in [Gho93, CTA97], Zhao and Zhang present an inverse segmentation procedure [ZZ97]. As an integration of edge and region information, this method firstly detects edges and critical points to approximate the curves. Triangulation procedure is subsequently used to connect pairs of edge points. Afterwards, small triangular surfaces are extended into large ones to form segmentation, based on the normal of the surface.

2.6.3 3D object reconstruction via point clouds

3D object reconstruction is very challenging task to build a computer or mathematical model to best fit the reality [Rem06]. This has been widely applied in different domains such as geomatics [Sam10], heritage digital documentation [KKOF04, EhBPG04], manufacturing [Sar07], reverse engineering [Zen12] and architecture [LN03].

Based on whether the source datasets are available or not, 3D object reconstruction can be categorised into two types: *reconstruction without object measurement* and *reconstruction based on object observation* [Rem03]. Reconstruction without object measurement is usually performed by animation software for artificial graphics and objects, which are mainly used in movies, video games, and object design (e.g., architecture, automobiles, etc.)[1]. Reconstruction based on object observation is to generate 3D models for real-life objects, from datasets collected by observation techniques. These datasets can be images and point cloud datasets. Object reconstruction from 3D point clouds could be quite challenging because: (1) the point cloud datasets are usually unorganised and very noisy, and (2) the surfaces of the object can be arbitrary (any kind of topological or irregular shape).

Remondino claims that polygons are usually the most flexible representation of 3D point clouds, which can describe the surface accurately and optimally [Rem06]. A good representation of surface have sparse polygons in regular areas, but dense polygons in explicit & irregular areas. The procedure of polygonal surface reconstruction is concluded as follows: (1) prepare

---

[1] http://www.3DLinks.com/

datasets including noise removal, hole filling, and point cloud resampling; (2) determine the global topology of this surface (e.g., adjacent areas) considering breaklines, to keep the principle spatial features; (3) create the polygonal surface to fulfil some quality requirements (e.g., size of mesh elements, no intersection of breaklines, etc.); (4) refine polygonal surface by edit-operations, like edge corrections, polygon editing, hole filling, and so on. Linsen also presents a similar process of point cloud representation [Lin01]. In addition, a multi-resolution strategy is applied to eliminate redundancy and handle huge datasets. Triangular irregular network (TIN) is used for reconstructing polygonal surface [Ede01]. Surface is described as a set of continuous and non-overlapping triangles.

2.7 Deformation Monitoring via Point Clouds

In this thesis, deformation monitoring can utilise very dense and high resolution points sampling on the whole object, rather than only discrete points scattered on/around this object by traditional geodetic surveying techniques. Such point clouds provide rich geometric information of the object under monitored. In the meanwhile, point clouds also bring new challenges in processing such huge datasets for extracting meaningful deformation information. Deformation monitoring via 3D point clouds is a complex process that involves various point cloud processing techniques, such as *object representation* to model point cloud, *object detection* to identify key element from point cloud, and *object tacking* to check deformation/movement. Before the studies of deformation monitoring via point clouds in geomatics domain, relevant techniques have been investigated in the fields of computer vision and computer graphics. In this section, we firstly introduce related works of deformation analysis in computer vision and computer graphics; afterwards, we discuss the development of this topic in the geomatic and geological fields; and finally, we make some comparison between the two different types of field.

2.7.1 Related work in computer vision and graphics

Deformation monitoring is the detection of the changed parts of a monitored object over time from a set of data at different epochs, and the analysis of their movement behaviours. The most related topic in computer vision and computer graphics is object tracking, which is analysing movement/deformation of the physical object with deforming geometry or texture in multiple frames of image (or multiple epochs of point cloud). Regarding the point cloud context, this question can be further formalised as follows: the detection of deforming geometry from a time series of unstructured, noisy point clouds. There are several crucial techniques like *object representation*, *detection* and *tracking* (or *correspondence searching*).

Object Representation - In computer vision/graphics, an object can be represented as anything whose geometry or texture changing with ongoing frames or epochs [YJS06]. Objects are associated with specific topics in different domains, like football players and football tracking in a match, neuronal stacks registration in medical images analysis, prototype shape design in reverse engineering, a spinnaker tracking during the sailing and performance capture in cartoon animation, etc. The tracked objects can be modelled by following three representations:

- *Point-based representation.* The tracking object can be simply modelled using the centroid of the object or a set of sampling points. Compared to the whole monitored region, such point-based object representation is quite sparse as a small subset [VRB01, SKMG04].

- *Boundary-based representation.* The tracking object can be modelled using a simple or complex boundary. Simple boundary can be some regular shapes like rectangles or ellipses containing the tacking object inside, which is suitable for both rigid and non-rigid objects tracking [NSC06, LHN09]. Complex boundary can be the edge contour of the tracking object or the silhouette inside the contour. This complex boundary attempts to simulate the real irregular shape of the object, e.g., hand and body shape, which is more suitable for complex non-rigid shapes in real life [GMV06].

- *Skeletal-based representation.* In addition, the tracking object can be modelled using the skeletal of the object, which is also suitable for complex non-rigid shapes [BWL+09].

Object Detection - The main objective of object detection is to recognise the object that is of interest from each frame or the initial frame with the object's appearance. In addition to the geometric information of the object, other rich description models such as colour and texture are also utilised for object detection. Some well-known methods such as *point detectors* and *segmentation* are highlighted here.

- *Point detectors.* Two main point detectors are using local features, i.e., *Harris and KLT detectors.* Both use the same matrix, i.e., the second moment matrix. The difference is that Harris detector selects an interest point by using the determinant and the trace of the second moment matrix; while KLT detector selects an interest point by the minimum eigenvalue of this matrix [HS88, ST94]. Different from using the second moment matrix, the *Scale Invariant Feature Transform* (*SIFT*) method is developed by constructing scale space [Low04]. SIFT detector selects interest points at different scales and different resolutions based on the difference-of-Gaussians images. Compared to Harris and KLT detectors, the number of interest points generated by SIFT detector are larger. SIFT detector is robust, not only to basic deformation (scale, rotation and translation), but also to affine transformations (changes in scale, rotation, shear, and position) and changes in illumination [MS05].

- *Segmentation.* In past, several segmentation techniques related to object detection are designed. (1) *Mean shift clustering* is an iterative procedure to locate density modes (i.e., the local maxima of its density) of a given distribution [CMM02]. This method does not require the number of clusters but the kernel bandwidth. The iteration procedure contains four steps: firstly, generate seed points and areas of interest around local maximum in difference image; secondly, determine a vector point towards the highest density point called 'mean shift vector'; thirdly, move area of interest to the point that the mean shift vector pointed to, and iterate these procedures until reaching the convergence; fourthly, group the paths that converge to the same mode together [SRPR08]. The drawback of segmentation using mean shift clustering is the further calculations and parameters tuning to get better performance. (2) *Graph cut based segmentation* is an image partition problem to minimise an energy function [BP07]. In this approach, each pixel in the image is regarded as a node in the graph. Nodes have two types, i.e., "object" or "background". Boundary between the object and the background is achieved using min cut (or max cut) algorithms [SFTA08, LLN11]. (3) *Active contour model or snake* is a framework to find a tight contour depicting the object boundary from an image [CKS97, PD00]. This is a minimisation problem of the sum of an internal and external energy of a contour [KC09, JK11].

Object Tracking - Object tracking is to continuously estimate the position of the object that is of interest from a series of datasets (e.g., images or point clouds) at multiple epochs. This can be achieved by detecting the object at each epoch and generating the trajectory of the object over time. However, such method is inefficient as a couple of reasons, e.g., the dynamics of object, learning of scene priors, and evaluation of different hypotheses [NT01, CRM03]. Another strategy of object tracking is maintaining the estimation of the object over time and predicting its location in forthcoming epochs [ARS06]. Compared to trajectory-based method, this approach is more efficient by updating object information estimated from previous epoch. The disadvantage of this method is the requirement of the priori knowledge about the object behaviour, e.g., the object information at the initial epoch. For both kinds of object tracking methods, *correspondence searching* is the crucial problem to build the association between neighbourhood epochs. We summarise object tracking techniques into three types based on its object representation, i.e., *point-based tracking*, *kernel-based tracking* and *contour-based tracking*.

- *Point-based tracking.* This kind of tracking is to track the object with point-based representation by finding the correspondence between the points over time. The motion model is frequently in the form of translation model. In order to predict the status of the moving object (e.g., its position) in the subsequent epoch/frame, the motion constraints can be formulated using object features like location, velocity, and rigidity [SJ87]. There are two main categories for point-based tracking, i.e., *deterministic* and *statistical* methods. (1) Deterministic point correspondence searching can be formalised as a combinatorial optimisation problem to minimise the correspondence cost of object association between the neighbouring epochs using multiple motion constraints. The selection of the one-to-one correspondence from all possible associations can be achieved by using greedy search methods [RS91, IDB97]. (2) Statistical correspondence searching method not only considers the object location in an image, but also takes the model uncertainty into account during object tracking. The widely-used estimation methods for statistical correspondence searching are Kalman filter [BC86, BK99] and particle filter [Rei79, GSS93, AMGC02]. Point-based tracking techniques can be applied to tracking a single point (e.g., a very small object in a image) or multiple points (e.g., larger object).

- *Kernel-based tracking.* Kernel-based tracking method represents the object using a region contained by regular shapes (e.g., rectangles, eclipses) over time from a set of images

[CRM03]. It uses motion models like affine or projective transformations. Template matching technique has been widely used for Kernel-based tracking. The features of the region inside rectangles or eclipses, such as image intensity, colour and gradients information, are used to form templates [Bir98, SBW02]. Additionally, colour histogram and mixture information of the image inside the rectangle or eclipse also can be used [PHVG02]. For example, template matching technique and mean-shift method can be integrated to track the object with a global motion efficiently [MCXL11]. During this procedure, the size of template is accordingly updated, not just fixed. Additionally, stable object feature can be selected from the dense field of displacement vectors computed by the translation of each pixel [HS81, ST94, YD96]. Optical flow information has been empirically demonstrated as hard constraints with benefits in efficiency and robustness. By combining with Kalman filter, such hard constraints can be softened while keeping the beneficial parts [DM00].

- *Contour-based tracking.* Contour-based tracking method is suitable for tracking an object with complex components like human body, hand, etc., which can be accurately described by contour or silhouette inside. Similar as template matching techniques with simplified boundary (e.g., rectangles), silhouette matching method is to find the object silhouette (i.e., a region with a complex boundary) from the images at current epoch. In addition to using the colour information by template matching, the object model can also use the object contour. For example, Hausdorff distance is used to identify the most mismatched edges and emphasise the edges which are not dramatically affected by object motion [HNR93, LCZD01]. Another approach is to find the corresponding object silhouette across the images at multiple epochs. Compared to the point matching, silhouette matching methods use the region inside and the object appearance features, not just location-based features [KCM04]. Besides tracking the silhouette inside, the active contour can be employed to track the complete object [KWT88]. This approach is to achieve a tight contour enclosing the tracking object by minimising the contour energy function. Colour and contour information can be integrated for contour tracking to handle occlusions [YLS04]. Contour-based object representation is suitable for a non-rigid object tracking with the motion model (parametric or nonparametric models).

Based on previous discussions of object tracking from the computer vision and graphics perspective, we identified a couple of rich techniques that related to deformation and movement analysis. From the images, rich information can be employed for object tracking, such as colour and texture information. For 3D shapes like point clouds and triangle meshes, these informative features are lost, while explicit geometry information can be used. Similar as tracking based on images, descriptors (e.g., few sparse points characterising parts of surface) are detected; and modelled by a Hidden Markov model to generate an invariant point signature; then, the similarity matching is performed on point clouds from multiple views [CCFM08]. Another related work is the 3D shape alignment. For example, rigid point clouds alignment is to find the corresponding point from different views. A concrete algorithm like ICP method is presented in Section 2.6.1. In cases of non-rigid alignment when the 3D shapes undergoing deformation, the quantification of estimated transformation needs to be integrated. The object under deformation over time can be aligned by a single global rigid motion. In addition to the correspondence between the overlapped parts, a wrapping function is required to match the deformed parts [LSP08].

2.7.2 Related work in geomatics and geological fields

Compared with the related work in computer vision and graphics, deformation monitoring via point clouds is still in the initial stage. The early studies of deformation monitoring via point clouds are typically based on commercial software to produce simple topographic information [FSC^{+}07]; and such information can help environmental scientists (or domain experts) in assessing the deformation like natural hazards. Recently, the literature start to build deformation models from 3D point clouds. The main research includes *geometrical model* and *deformation extraction*.

Geometrical Model - The rich point clouds of the monitored object provide the possibility to identify the geometrical properties of the object and to build a geometrical model for analysing the deformation. Two types of geometrical models, *regular shape based model* and *irregular shape based model*, are largely applied to extract the deformation.

- *Regular shape based model.* The first group of geometrical models are based on the shape of the monitored object, e.g., plane, cylinder, etc. Therefore, we call them "regular shape based model". Lindenbergh et al. apply segmentation and reconstruction methods in deformation analysis of a tunnel [LPR05]. This tunnel is divided into a series of segments and is represented as *a cylindric model.* After the model is built, the stability of this cylindric model is analysed. Schneider uses a "bending line" model to determine deformation of a television tower by the following steps: (1) cut point clouds of this tower into 15 layers with incremental height; (2) project points of each layer onto 2D planes to form a circle; (3) calculate the centre and the radius of each circle by a circle-fit algorithm; (4) link all centres at different layers to represent the bending line; (5) determine the deformation based on the bending line [Sch06]. The advantage of this kind of shape-based geometric model is using the object's geometric descriptions that can better understand the objects (e.g., tunnels, dams, planar walls of some buildings, etc.) with regular shapes.

- *Irregular shape based model.* In contrast to regular shape based models, the second geometrical models are built as the irregular surfaces such as triangle meshes. For example, DSM (Digital Surface Model) is used to convert 3D point clouds into a continuous surface and then induce comprehensive visualisation of the monitored region. Based on DSM, additional topographic products are derived, such as cross-sections, contour lines, orthophoria and topographic maps; and these products can be used to analyse the deformation [GS04]. Another irregular shape based modelling method is to resample the point clouds to build an two dimensional grid network; and this grid network is used to formalise a reference for point clouds comparison to extract deformation [SWKZ04]. Alba et al. propose a similar surface-based comparison method to extract deformation as well as to locally filter measurement noise [AFP^{+}06]. In this study, two types of meshes such as *a triangular mesh* and *a regular polynomial 3D surface* are interpolated and compared using the shortest distance with resampled point clouds.

Deformation Extraction - Based on the geometric models, we can extract deformation information (e.g., parameters like displacement) by comparing the difference between the models at two epochs of point clouds. Displacement is an important parameter in the assessment of natural hazards like subsidence, rockfall and landslide, etc. It is frequently used to represent the deformation on an object. For example, after building the irregular shape model using grid-network in [SWKZ04], the displacement along Y-axis at different epochs is used to describe the deformation of the gate scenario. In [AJOV09], a simple "nearest neighbour averaging" model is built to enhance the accuracy of displacement detection. In the shape based modelling of the bending line in [Sch06], the deformation is represented by the two parameters, i.e., the displacement between the circle's centre, and the difference between the circle's radius at different epochs.

In addition to extract deformation parameters using the models separately built on the point cloud datasets at different epochs, another kind of approaches directly extract deformation from two epochs of point clouds, without using individual geometric model on each epoch. The well-known methods are the ICP (Iterative Closest Point) and the LS3D (Squares 3D Surface Matching) for extracting rigid-transformation. The two methods directly apply a combined model that covers both epochs of point clouds, and extract deformation parameters like "rotation" and "translation". Monserrat and Crosetto apply LS3D for estimating these deformation parameters by local surface matching [MC08]. Similarly, Teza et al. propose a piecewise application of the ICP algorithm to calculate a displacement field of a landslide [TGZG07].

2.7.3 Comparison between two different types of field

In Section 2.7.1, we present the related research topics such as object representation, detection and tracking from the point of view of computer vision and graphics. In Section 2.7.2, we introduce the current state of the same topic in geomatics and geological fields. We can observe that this topic has been studied deeper in computer vision and graphics than in environmental field. A large amount of theoretical and fundamental works have been proposed and developed for various applications like reverse engineering and animation. Nevertheless, deformation monitoring via 3D point clouds is still a relatively new topic in environmental domain, compared with its development in computer vision and computer graphics. Its applications either depend on the commercial softwares, or are based on some basic processing techniques from computer vision and computer graphics.

There are some features of deformation monitoring via point clouds in environmental fields: 1) The frequency of point clouds collection in environmental domain mainly depends on the requirements of natural hazard monitoring. For some monitoring scenarios like a sudden and quick landslide, point data is collected frequently in a short time; while for other monitoring scenarios such as a structure quality control, the frequency of point data acquisition is much lower, e.g., once per year. In the second cases, the datasets might not be considered as continuous as the datasets in the first scenarios. 2) In many case studies, point clouds provide only the geometry information of the object, not including other rich descriptions based on colour and texture, which is the same as in computer graphics. Therefore, deformed parts of the monitored object need to be detected and tracked based on the distinctive geometric features. 3) Another advantage of point clouds is their the precise scale of the object, not like the camera images. The distance between the camera and the object may cause scale difference and illumination changes of the object.

From the literature on deformation monitoring using LiDAR, we identify the main advantage is the high-density and resolution 3D point clouds, in contrast to the sparse/limited data points in traditional deformation analysis [Ste06]. 3D point clouds provide great potential to build rich deformation for the whole monitored object. Existing works on deformation monitoring using point clouds in environmental fields have built a couple of nice geometrical models and deformation parameters to identify the deformation. The main problem is that the whole area is based on a single model (e.g., surface, or cylinder) or the same parameters (e.g., displacement or rotation). In this thesis, we provide an automatic cell-based deformation computation that support hybrid modelling, in which different cells can apply distinct and the most suitable model separately.

2.8 Summary

This chapter provided a comprehensive overview of the related works of this thesis. We identified that deformation monitoring via 3D point clouds has become an attractive topic in the environmental field, and investigated five main technical aspects. The former three aspects (i.e., *basics of deformation monitoring*, *deformation measurement* and *deformation analysis*) focused on traditional deformation monitoring techniques using geodetic surveying, while the latter two aspects (i.e., *terrestrial laser scanning* and *point cloud processing*) presented terrestrial LiDAR techniques and deformation analysis using point clouds. Traditional deformation analysis methods based on discrete points are neither sufficient nor suitable for directly processing high-density point clouds. Existing methods on deformation monitoring via point clouds are still on the very initial stage. Point cloud datasets contain detailed geometric information of the monitored object. This enables researchers and engineers to build informative deformation modelling and computation. Therefore, the main objectives of this thesis are firstly to build a hybrid model for achieving rich and explicit deformation description, and secondly to automatically compute deformation from 3D point clouds utilising the hybrid model.

Chapter 3

Deformation Modelling

> *A model should be as simple as possible but no simpler.*
>
> *Albert Einstein (1879-1955)*

3.1 Introduction

In this chapter, we investigate the problems of deformation modelling, focusing on the fundamental modelling issues as well as the modelling approaches provided in this thesis. In particular, we analyse the modelling requirements for describing deformation and design *a novel hybrid model* to provide a rich deformation for a monitored region. Such hybrid deformation model is able to model a big region with diverse deformation characteristics inside.

This chapter is organised as follows: Section 3.2 discusses the detailed modelling requirements to represent a rich model for deformation monitoring, which includes deformation object ("what"), deformation type ("which"), and deformation level ("how"). Section 3.3 briefly overviews the traditional deformation models, which are typically global models, and point out the limitations of global modelling. In Section 3.4, we design a novel hybrid model that is able to support diverse deformation characteristics for detecting the deformation of a given object. Our hybrid model provides different levels of modelling, from "meta-deformation" for a small area (e.g., cells in our context), to "sub-deformation" for a partial area, and finally to "deformation-map" for the whole region. Section 3.5 summarises this chapter.

3.2 Medelling Requirements

This section presents the detailed requirements of deformation modelling. Based on our investigation in Section 2.2 on deformation scenarios, deformation can be generally defined as a change of an object caused by external forces. To provide a detailed deformation modelling, we need to analyse the three main components: (1) identifying "*what*" has deformation under a given monitoring scenario, i.e., *deformation object*; (2) detecting "*which*" kind of changes happened on the object, i.e., *deformation type*; and (3) investigating "*how*" to describe the volume of the changes, i.e., *deformation level.*

3.2.1 Deformation object - "What"

As already briefly mentioned in Section 2.2, deformation objects can be at different scales, e.g., the local level, or the regional level, or even the worldwide global level. Deformation objects belong to the "what" requirement in modelling deformation. We now discuss deformation objects in terms of their two main types, i.e., *artificial objects* and *natural objects.*

Artificial Objects – In deformation monitoring, artificial objects typically are human-made structures. Artificial objects can be structures in laboratory for the behaviour analysis of some basic elements used for construction (e.g., beams) under load test [GLFS04]. Additionally, a huge amount of real-life structures are monitored continuously or periodically. Some real-life structures are in a large scale. For example, a dam of Cancano lake (Valtellina, Italy) is an arc gravity structure with a height of 136 m and a width of 381 m at the top, which generates a basin of about 124 million m^3 of water [AFP$^+$06]. The security of these big structures is a serious problem, regarding to their significant impacts on the region where they are located. Therefore, detection and tracking real-life structures are required for the conservation of the structure itself and the surroundings. In addition, researchers also pay a lot of attention on other types of civil structures such as *bridges* (e.g., the Cernadela Bridge in Spain with five arches), *towers* (e.g., the Dresden television tower in Germany with the height 252 m and the diameter 9.4 m at the bottom), *lock gates* at hydropower station, *tunnels* (e.g., Zaventem, Belgium) and so on.

For the engineering structures discussed above, we summarises their characteristics as follows: (1) Materials of artificial structures are relatively uniform; (2) Artificial structures usually contain some regular shapes. For instance, construction elements are normally following a standard form, dams are frequently in arches shape, bridges also comprise arches, towers can be in a cone shape, tunnels are similar as a cylinder. Some basic geometrical characteristics can be applied to simply describe these objects and easily find their common features.

Natural objects - Compared to artificial structures, natural objects are more complicated. Natural objects like mountain valley and road-side slopes are typically arbitrary and unique, and they frequently locate in complex geological condition areas, especially in hilly and mountainous regions. Their stability is affected by many aspects such as *geology* (e.g., structural rock properties), *geomorphology* (e.g., slope gradients), *hydrology* (e.g., water content) and vegetative factors (e.g., vegetation cover).

We identify the common features of natural objects as follows: (1) natural structures might contain different materials in a test site; (2) surface of a natural object is frequently irregular and is difficult to summarise geometrical features; (3) conditions of natural objects are complex, affected by geological, hydrologic and vegetative factors. Table 3.1 provides a summary of deformation objects mentioned before.

Table 3.1: Summary of deformation objects

Object Types	Examples	Features
Artificial objects	Construction elements in laboratory Real-life engineering structures e.g.,dams, bridges, towers, hydro-stations	a. Relatively uniform material b. Regular surface
Natural objects	Mountain valley Road-slide slopes	a. Different materials b. Irregular surface c. Complex condition

3.2.2 Deformation type - "Which"

Deformation of an object is a complicated natural phenomenon that could be driven by environmental changes (e.g., temperature, erosion, subsidence, etc.) or external forces (e.g., rainfall and snowfall, earthquakes, vegetation change, etc.). To provide a meaningful deformation description, we need to identify a suitable *deformation type*, which is one of the three most important requirements in modelling deformation. Deformation type can be an atomic geometric change, e.g.,*displacement*, *rotation*, *pure shear*, *simple shear*, or a complicated deformation that is composed of several different atomic changes. In this section, we investigate the most frequently used deformation types.

We firstly study the atomic deformation types that are widely used in deformation monitoring, for example, *displacement*, *rotation*, *translation*, *strain*, *pure shear*, *simple shear*. According to the literatures and the deformation characteristics, we can divide these atomic deformation types into two categories, i.e., *rigid deformation* and *non-rigid deformation.*

Rigid Deformation - Rigid deformation is widely used for analysing the change of an object, which can be measured by using displacement. Such displacement has no change in terms of *shape* or *size* of the object, i.e., the object keeps identical and the points insides the object are relatively stable. Displacement is one of the most general concepts for describing deformation in different domains such as physics, structural engineering and geology. Typically, displacement is defined as the movement of an object due to some external forces, i.e., measuring the difference from the initial position to its final position. To analyse the displacement of a given object under monitoring, researchers and engineers frequently apply the Euclidian distance to measure the change, from specific dimensions (e.g, 1D, 2D, or 3D). For example, distances along the longitude/latitude direction (i.e., X and Y axis) is typically used for analysing *horizontal displacement*; whilst the distance along the altitude direction (i.e., Z axis) is used for analysing *vertical displacement.* The vertical displacement is widely used for analysing the deformation of *subsidence* for natural environment. In addition, some literatures and scenarios focus on analysing another type of displacement along non-fixed directions, i.e, *angle displacement.* In some deformation terminologies, *horizontal displacement* and *vertical displacement* are grouped together and entitled as *translation*; whilst *angle displacement* is called *rotation.* Figure 3.1 illustrates these rigid transformation types in terms of three aspects, i.e., *semantic description*, *graphic description* and *examples.* More detailed explanation of these examples are listed in the following paragraphs.

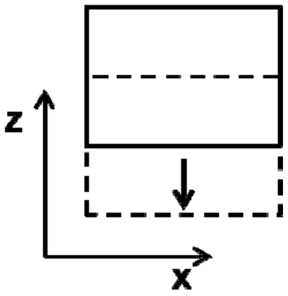
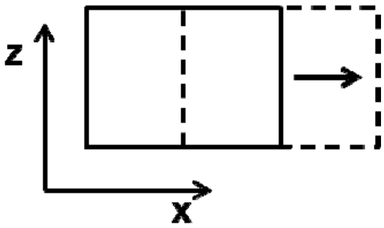
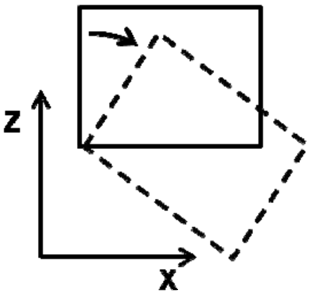
Semantic Description	Graphic Description	Examples
Vertical displacement (along z axis)		• Vertical deflection of construction elements under incremental loads • Displacement between structure-surface model at different epochs • Subsidence, erosion • Translational movement • Elevation difference
Horizontal displacement (along x or y axis)		• Horizontal displacement of a dam caused by water level change in a basin • Horizontal surface displacement between the reference and template
Angle displacement (Rotation)		• Rotation of rock movements in a landslide • Approximation of rotational motion of a landslide • Angle variation of partial area inside a slope

Figure 3.1: Rigid deformation types

There are a lot of application scenarios using these types of rigid deformation to represent the deformation of artificial structures and natural objects. *Vertical displacement* is a most general change description applied for construction elements under loading test, real-life engineering structures and unstable slopes. Some examples are lists in the following aspects:

- *Construction elements.* Many construction elements including a timber beam (5.0 m × 0.2 m × 0.1 m), L-shaped concrete beam (7.0 m × 0.5 m × 0.5 m) and timber stringers (a horizontal beam with a cylindrical shape) are tested with incremental loads [GLFS04]. Vertical displacement is frequently chosen as the major deformation feature in these case studies. Such displacement can be extracted from both terrestrial LiDAR datasets and photogrammetric targets and compared. More examples can be seen in [OKCH10].
- *Dams.* Dams are one kind of most common real-life structures. Displacement has been

frequently used to describe the deformation of dams, for example, Pacoima Dam (California, U.S.) [BHK98], Alibey Dam in Turkey [GKHMA06], the Cancano Lake (Valtellina, Italy)[AFP$^+$06], Mornos dam in Greece [GS08], Ataurk Dam (Turkey) [KAB10], etc.

- *Bridges.* Vertical displacement also plays an important role in deformation analysis for bridge safety monitoring. One-dimensional displacement is taken from several positions on the bridge and used to estimate the maximum displacement in the centre [MH06]. Moreover, deformation of the bridge displacement between two epochs is extracted in static load test and dynamic test. More case studies of bridge displacement are presented in [SBT93, Nak00, NPS06, Tan09].

- *Other structures.* For other kinds of structures, displacement is also applied for deformation representation. For example, displacement of cooling tower (via surface model) at two epochs [IVGT06], surface displacement of the lock gates of a hydropower station [SWKZ04], a tunnel surface displacement [NDWB$^+$10], and so on.

- *Unstable slopes.* Vertical displacement is also widely used in analysing deformation for slope movement. As a worldwide natural phenomenon, slope movement has several types, such as falls, topple, slide, spread, and flow [WS09b]. From geometrical change perspective, all these slope movements include vertical displacement. For instance, a displacement is identified in the central part of La Valette landslide located in French Alps [SDA05]. In Galierm landslide case study, slope movement rate was modelled by displacement vectors [PP09]. Elevation difference of a highway embankment (an earthwork slope in Northumberland, U.K.) is tested and this displacement is coincidence with known changes [MMB$^+$08]. More examples are given in [KÖ0, SPT$^+$03, KYS08, JOA$^+$10, CSR10, LDHH11].

In addition to vertical displacement, *horizontal displacement* is another frequently-used deformation indicator in various scenarios. The dam deformation can be caused by water level difference in the basin, and this is modelled by horizontal displacement on the middle of dam crest [AFP$^+$06]. Horizontal surface displacement is extracted by using Euclidian distance between the reference surface and the matching templates [DGK12].

Besides vertical and horizontal displacement, *rotation (angle displacement)* of monitored object is yet another kind of deformation model which is interest to researchers and engineers. For example, three rocks movements in the landslide located in the Central Pyrenees (Spain) were represented by using rotation [MC08]. A displacement vector map is generated to approximate the rotational motion of the landslide at Perarolo di Cadore (Italy) [TPGG08]. The stability of a slope is described by angle variation of partial area inside [WGM12].

Non-Rigid Deformation - In contrast to rigid-deformation that analyses the displacement without change on size or shape of the monitored object, non-rigid deformation can analyse changes in terms of shape and size (inside). In other words, rigid deformation is about *absolute* change according to a reference coordinate; whilst non-rigid deformation studies the *relative* change for the partial points on the object under monitoring. Non-rigid deformation has three main types, i.e., *pure shear*, *simple shear*, and *strain.* Pure shear is to measure the expand (enlarging) or contract (diminishing) of an object along one or more axis (X, Y, or Z), corresponding to the change of the size. Simple shear is to measure the local angle change; as shown in Figure 3.2, an object to simple shear does not change in length but undergoes a change in shape. Strain is to further describe the unit relative displacement of particles on the object, and it typically uses the gradient to quantify the tensor. Figure 3.2 illustrates these non-rigid deformation.

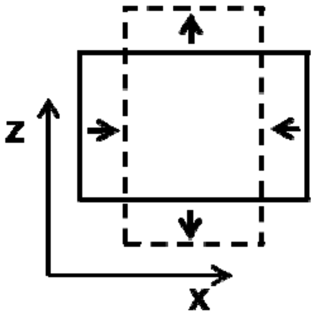
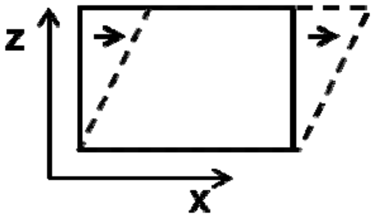
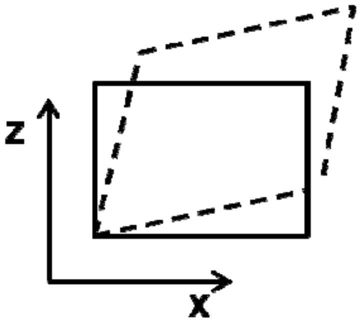
Semantic Description	Graphic Description	Examples
Pure shear		• Compression motion (volume decreasing) • Extension motion (volume increasing) • Non-rigid deformed object matching • Elongation of an object along x or y axis • Volumetric change in rock glacier
Simple shear		• High shear caused by significant change in flow direction and speed • Relative rotation trend of partial area in landslide
Strain (ratio)		• Normal strain (the change in length of lines inside an object) • Shear strain (the change in angle between intersecting lines inside an object) • Characterization of the kinematics of the landslide surface

Figure 3.2: Non-rigid deformation types

These non-rigid deformation types have been largely used in many application scenarios. *Pure shear* is a factor related to the size of the object and is usually categorised into *compression* and *extension*. These two motions show the volume decreasing or increasing of the object. If the pure shear factor is greater than 1, the object becomes bigger, which is in the state of extension. On the contrary, if the pure shear factor is less than 1, the object becomes smaller, which is in the state of compression. Volumetric changes are explicated presented in rock glacier [HHA$^+$06]. Pure shear parameter is included as one element in a 7-parameter 3D similarity transformation method; such 7 parameters include 3 translation (along X, Y, Z axis), 3 rotation (for the 3 axis), and 1 pure shear. This method has the ability of handling multi-scale datasets [GA05]. Similarly, pure shear is also employed in least squares matching algorithm to represent the elongation of an object along X or Y axis [DGK12].

Simple shear is an indicator of relative rotational movement of an object. For example, a high simple shear rate is detected from partial areas in the rock glacier located in the Muragl valley of the Upper Engadine area in the Swiss Alps, which caused by the significant changes in the flow direction and speed. Moreover, the Nigardsbreen Glacier and the La Clapiere Landslide is also demonstrated by using simple shear pattern [DGK12].

As a normalised value for describing deformation, *Strain* can be divided into *normal strain* and *shear strain*. Normal strain is to represent the change in length of lines inside an object; whilst, shear strain is to describe the change in angle between intersecting lines inside an object. Normalised shear pattern has been shown in the contour map for analysing the change at Lamosano village in the northeastern Italian Alps, which indicates the complex kinematics of the landslide happened there [TPGG08].

Complex deformation - In real-life deformation scenarios, different deformation types usually happen in parallel, and compose a more complicated deformation situation. Rigid (like displacement, rotation) and non-rigid deformation (like pure shear, simple shear and strain) are mixed together in a real-life case study. Therefore, we need to combine these atomic deformation types together to demonstrate the real change and movement of an object. Recently, researchers start to build mixed models for analysing real-life deformation phenomena. For example, the deformation analysis of the Perarolo di Cadore landslide (in Italy) has applied the strain model on the field, as well as the displacement model [TPG08].

3.2.3 Deformation description - "How"

After the summarisation of various deformation types, how to describe deformation is another important issue to assist in the understanding of the deformation phenomena. Generally speaking, there are two kinds of strategies for deformation description: *quantitative* and *qualitative*.

Quantitative description. Quantitative description is a numeric representation of one specific deformation including magnitude and direction. For each deformation, we can calculate a quantitative value through post-processing. Based on the physical meanings of these values, quantitative description can be categorised into *absolute value* and *relative value*. Absolute values, such as displacement and rotation, are the movement of a rigid object, which means that the initial positions inside the object keep the same. Relative values, like normal strain and shear strain, are the the change inside the object, which means that the size or the shape of the object could change in this procedure. The description of these deformation types has already been mentioned in Section 3.2.2. In this section, we provide a more detailed quantitative measurement of these deformation types.

Translation displacement — We assume that there is an object with the spatial changes driven by the action of external force. A point P is assumed to be on the object surface and is moved from the initial position $P^I = (x^I, y^I, z^I)$ at Epoch I to the final position $P^{II} = (x^{II}, y^{II}, z^{II})$ at Epoch II, shown by Figure 3.3. P^{II} can be expressed by vector as Equation 3.1 and by coordinate as Equation 3.2:

$$P^{II} = P^I + D \tag{3.1}$$

$$\begin{bmatrix} x^{II} \\ y^{II} \\ z^{II} \end{bmatrix} = \begin{bmatrix} x^{I} \\ y^{I} \\ z^{I} \end{bmatrix} + \begin{bmatrix} D_x \\ D_y \\ D_z \end{bmatrix} \tag{3.2}$$

where D is the translation displacement vector; D_x and D_y are the horizontal displacement along $X-$ and $Y-$ axis, respectively; D_z is the vertical displacement along $Z-$ axis.

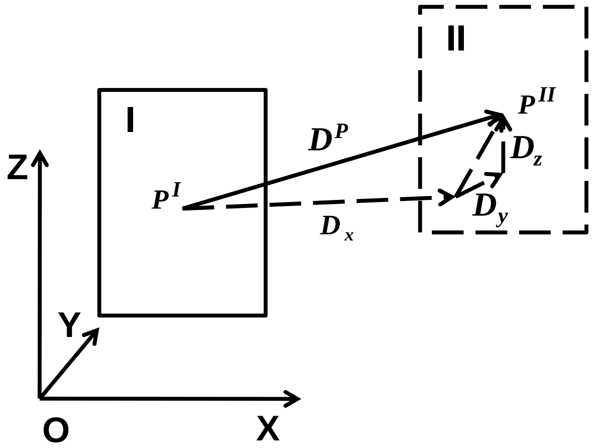


Figure 3.3: Vertical and horizontal displacement of a point from Epoch I to Epoch II

Rotation displacement — We assume that an area is rotated from the initial position at Epoch I to the final position at Epoch II, shown by Figure 3.4. This area can be a whole object or a sub-area of an object. Points on this area are $\mathbf{P}^I = \{P_1^I, \cdots, P_n^I\}$ at Epoch I and $\mathbf{P}^{II} = \{P_1^{II}, \cdots, P_n^{II}\}$ at Epoch II. $\mathbf{P}^{II}$ can be written as

$$\mathbf{P}^{II} = \mathbf{R} \times \mathbf{P}^I \tag{3.3}$$

$$[P_1^{II}, \cdots, P_n^{II}] = \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix} \times [P_1^I, \cdots, P_n^I] \tag{3.4}$$

where $\mathbf{R}$ is the rotation matrix of this rigid area and is regarded as rotation displacement; R_{ij} is the element of this rotation matrix.

Pure shear — We assume that an object is extended or compressed from Epoch I to Epoch II (see Figure 3.5). Pure shear is the change in size between the final object and the initial object,

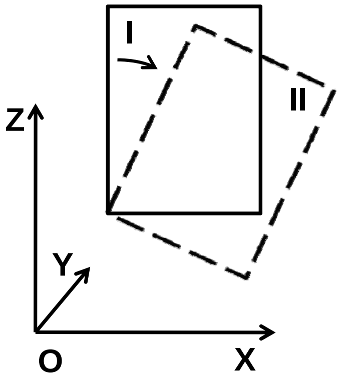


Figure 3.4: Rotation of an area from Epoch I to Epoch II

which be expressed as follows,

$$S = \frac{l + \Delta l}{l} \tag{3.5}$$

where S is the anisotropic scaling change between two epochs; l is the original distance at Epoch I; Δl is the changed distance at Epoch II, as shown in Figure 3.5.

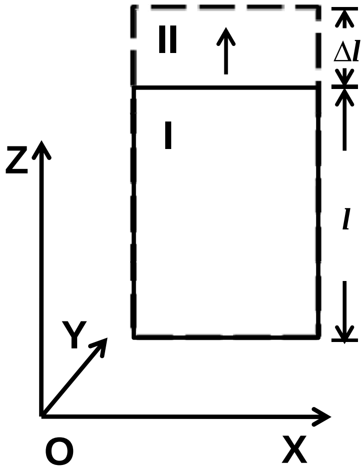


Figure 3.5: Anisotropic scaling change of an object from Epoch I to Epoch II

Simple shear — We assume that the shape of an object is changed under an external force. The force direction is shown by an arrow in Figure 3.6. Simple shear is the change in angle between two lines that are initially perpendicular and is expressed as:

$$\gamma = \theta \tag{3.6}$$

where θ is the angle at the corner of an initial rectangular element.

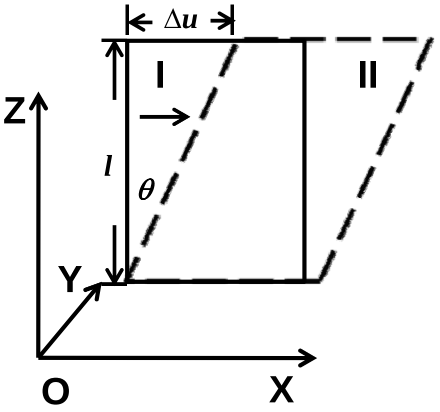


Figure 3.6: Simple shear of a line from Epoch I to Epoch II

Strain — Strain is the ratio of deformation (e.g., the change of the length) and the original length. Normal strain is about the length change, whilst shear strain is more related to the angle change. Normal strain and shear strain can be expressed by Equation 3.7 and Equation 3.8, respectively.

$$\varepsilon_n = \frac{\Delta l}{l} \tag{3.7}$$

$$\varepsilon_s = \frac{\Delta u}{l} = \tan\theta \tag{3.8}$$

where Δl and l are the same as the previous definitions in Equation 3.5; Δu is the movement of a changed corner under the force. Table 3.2 summarises the quantitative deformation description.

Deformation type		**Deformation parameter**	**Geometric meaning**
Rigid deformation	Translation displacement	$\mathbf{D} = \{D_x, D_y, D_z\}$	Displacement along along $X-$, $Y-$, and $Z-$ axis, respectively
	Rotation displacement	$\mathbf{R}$	Rotation of an object from the initial position to the final position
Non-rigid deformation	Pure shear	$\mathbf{S}$	The change in length between the final position and the initial position
	Simple shear	γ	The change in angle between two lines that are initially perpendicular
	Strain	$\varepsilon_{\mathbf{n}}, \varepsilon_{\mathbf{s}}$	The ratio of deformation and the original length

Table 3.2: Summary of deformation parameters

Qualitative description - The quantitative deformation description is largely used in the literature by researchers and domain experts in the environmental scientists. However, such quantitative deformation values are hard to be directly understood by non-expert people in most cases. Therefore, from the semantic point of view, we need related qualitative description to provide a more meaningful representation of the associated numeric deformation. Usually, these semantic description is defined by empirical knowledge.

For example, to monitor the deformation of a slope condition, there are two numeric measures, i.e., the factor of stability and the topographic wetness index. However, even with such two values, people are still hard to know the real underlying meanings. In order to help users to easily understand current safety situation of the slope, a three-level semantic description $\langle Low,\ Medium,\ High\rangle$ are classified for risk grading according to the defined threshold [MMB11]. Similarly, the geometric significance of relevant deformation results is also categorised into the same three categories, i.e., $\langle Low,\ Medium,\ High\rangle$, and finally such qualitative description is used to generate a significance map [TPGG08].

3.3 Global Model

In the previous section, we have already expressly discussed the deformation modelling requirements, which include the problem of "*what+which+how*". Based on this modelling requirement, the objective of deformation modelling is to concretely identify these components for a given deformation scenario that needs to be analysed, especially deformation type and description.

The deformation monitoring of an unstable object in essence is to estimate geometrical changes of this object during a time period. For a given deformation scenario, researchers and engineers set up an experimental platform that can capture observations (measurements, or datasets). Based on the measurements, the deformation (or changes) can be estimated in terms of comparing the observations sampled on this object between different epochs; and we can consider the differences of this object at different epochs as the deformation. Technically speaking, the main concern of deformation modelling is to formalise the input and the output of the deformation monitoring, where the input is the observation datasets, and the output is the modelling three components (i.e., the "*what* + *which* + *how*"). The detailed procedure of "how to get the output from the input" is the focus of deformation computation, discussing our cell-based computing framework that will be the main content of Chapter 4.

Low-sampling datasets collected by classical geodetic methods such as total station or GNSS technique are one kind of standard inputs for deformation analysis. This is largely used in traditional deformation analysis literatures. Figure 3.7 shows the sparse points on an object under monitoring. In this scenario, the coordinates of these control points $\{L_1, \cdots, L_n\}$ are the observations with low-sampling density (typically less than 20 observation points for most scenarios). For deformation analysis, measurements at two epochs are at least required such as $\mathbf{L}^I = \{L_1^I, \cdots, L_n^I\}$ at Epoch I and $\mathbf{L}^{II} = \{L_1^{II}, \cdots, L_n^{II}\}$ at Epoch II.

To estimate the deformation based on these observations at two epochs, researchers and engineers typically build a global model (see Equation 2.2 in Section 2.4.2) using these discrete point datasets. The modelling outputs are the deformation parameters that can be extract-

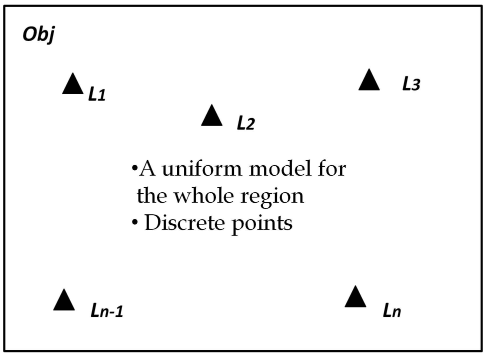


Figure 3.7: Low-sampling datasets of a monitored object

ed/estimated by relevant optimisation method. In this section, we present some general global models that frequently used in deformation estimation, and explain the geometric meanings of the parameters in these models.

Displacement vector - Displacement vectors **D** are associated with a set of common control points at two epochs and are formalised as

$$\mathbf{D} = \mathbf{L}^{\mathbf{II}} - \mathbf{L}^{\mathbf{I}} \tag{3.9}$$

Usually, these control points have two groups: stable points (i.e., L_{Stable}) and unstable points (i.e., $L_{Unstable}$). Stable points are used to compute the coordinate frame; whilst, unstable points are applied for extracting the difference (e.g., displacement). Therefore, the displacement vector is restricted to the stable common points and their related coordinate system.

Polynomial Approach - As a well-applied displacement function, polynomial approach provides the approximation of the displacement field. A polynomial function in 2D dimensions can be formalised as

$$\begin{pmatrix} dx \\ dy \end{pmatrix} = \begin{pmatrix} a_0 + a_1 x + a_2 y + a_3 xy + a_4 x^2 + \cdots \\ b_0 + b_1 x + b_2 y + b_3 xy + b_4 x^2 + \cdots \end{pmatrix} \tag{3.10}$$

where x and y are the coordinates of the discrete points; dx and dy are the displacement along the x and y axes.

S-transformation - Based on the displacement vectors of a deformation field, we assume that deformation of points between two epochs follows a relative rotation, translation and a scale adjustment. Therefore, a s-transformation (similarity or Helmert transformation) model is built [Che83, SS01]:

$$\mathbf{D} = H \times T^S + V \tag{3.11}$$

where **D** is displacement vector in deformation field; V is residual vector; T^S is S-transformation matrix expressed in Equation 3.12; H is a design matrix given by Equation 3.13

$$H = \begin{bmatrix} t_x & t_y & r_z & s \end{bmatrix}' \tag{3.12}$$

where t_x and t_y are the translation along $X-$ and $Y-$ axis, respectively; r_z is the rotation around $Z-$ axis; s is the scale of the network.

$$H = \begin{bmatrix} 1 & 0 & 1 & 0 & \cdots & 1 & 0 \\ 0 & 1 & 0 & 1 & \cdots & 0 & 1 \\ y_1^c & -x_1^c & y_2^c & -x_2^c & \cdots & y_n^c & -x_n^c \\ x_1^c & y_1^c & x_2^c & y_2^c & \cdots & x_n^c & y_n^c \end{bmatrix}' \tag{3.13}$$

where x_i^c and y_i^c are the coordinates of point $L_i = \{x_i, y_i\}$ which are reduced to the centre of gravity of geodetic network formed by all the control points, i.e.,

$$x_i^c = x_i - \frac{\sum_{i=1}^{n} x_i}{n} \tag{3.14}$$

$$y_i^c = y_i - \frac{\sum_{i=1}^{n} y_i}{n} \tag{3.15}$$

where x_i and y_i are the coordinates of point l_i; n is the number of common points in the geodetic network. This S-transformation model is suitable for two-dimensional geodetic network. The last row of H can be omitted, when there is no scale change.

3.4 Hybrid Model

In the previous section, we discussed the global models, which are largely used in traditional deformation monitoring. For a given object under monitoring, the global model typically applies only one type of deformation, e.g., displacement. However, for a large area (e.g., a big mountain slope) being monitored, we may need to apply different models for specific partial areas; therefore, this impulses us to build a richer and hybrid model, where different models can be applied to selected regions in the monitoring object.

Traditional geodetic data collection has limited data points, which are suitable for applying the global model (as the observation points are quite sparse like only 20 points for a large region) but not for the hybrid model. Recently, high density point cloud datasets (e.g., a point cloud has one point per 5mm) offer us the possibility to utilise *the geometrical features* of the monitored object. The geometrical features include the basic shapes of monitored objects, e.g., plane, cube, cylinder and so on, which can not be used in the classical analysis methods with low sampling density data (discrete points). Based on these geometrical features, a global model of the monitored object is applied to fit the point clouds on the surface of this monitored object at different epochs. For example, this method is performed in the deformation analysis of a bored tunnel by means of terrestrial LiDAR [GLP06]. In this work, A cylinder - the main geometrical features of tunnel - is used to parameterise and fit the actual tunnel point clouds; the existence of the deformation occurred between two epochs of tunnel measurements is verified. In addition, the Cartesian coordinate system is adjusted into a cylindrical coordinate system to reduce the observation space and build more efficient estimation.

Point clouds (see the dense points in Figure 3.8) provide the capability to detect detailed surface information of the monitored object, especially for the partial areas of this object. In Figure 3.8, D_1, D_2 and D_3 are the three important partial areas in this object; and there three partial areas have distinctive deformation characteristics that need to be independently identified. Therefore, we need a richer and hybrid model for such real-life scenarios. Now, the problem of deformation analysis can be formulated as follows: *given two sets of point clouds sampling on a monitored object Obj at the two different epochs (i.e.* $\mathbf{P}^I = \{P_1^I, P_2^I, \cdots, P_{n_I}^I\}$ *and* $\mathbf{P}^{II} = \{P_1^{II}, P_2^{II}, \cdots, P_{n_{II}}^{II}\}$ *both in* $\Omega \subseteq \mathbb{R}^3$*, where each Point* $\mathrm{P}_i^I = [\mathrm{x}_i^I, \mathrm{y}_i^I, \mathrm{z}_i^I]^\top$ *and* $\mathrm{P}_i^{II} = [\mathrm{x}_i^{II}, \mathrm{y}_i^{II}, \mathrm{z}_i^{II}]^\top$*), we want to compute informative deformation description of Obj.*

Now, we want to further identify the limitation of global modelling on the rich point clouds dataset, and claim that the global model based on discrete points (see Equation 2.2) is not an appropriate solution for our proposed problem. Two main reasons can be summarised as *computational ability* and *distinctive deformations of the object.*

- From the point of view of *the computational ability*, the global model based on discrete points can not satisfy the requirements of modelling high density point clouds. With millions of point cloud observations $\mathbf{P}$, the number of observations N_P is too much larger than the number of parameters N_X ($N_P \gg N_X$), if applying a global model (e.g., rigid-deformation like displacement). Therefore, it is ineffective to solve this overdetermined system. A global deformation model can not be estimated for the whole object using the point cloud datasets. Therefore, instead of using a global model for all of the big region under monitored, we propose cell-based method that only a small cell applies a single model. In Chapter 4, we will further discuss why we need the cells to split the problem.
- For traditional geodetic observations, the global model works well for an object with similar/uniform deformation in the whole object $M(X)$, but not *an object with distinctive deformations* in different partial areas. Because the deformations on an object in nature is heterogeneous, which means the change direction, change orientation and deformed volume

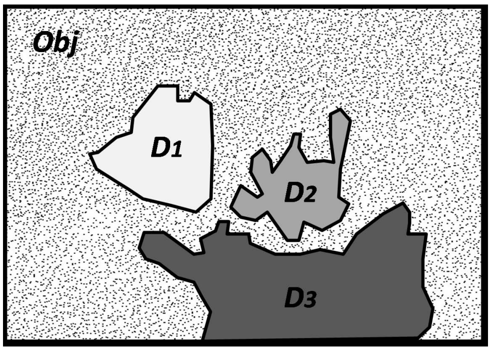


Figure 3.8: Point cloud datasets of a unstable object

are not necessary to be consistent in the whole deformed object. For example, Figure 3.8 has three specific areas (D_1, D_2 and D_3) with distinctive deformation characteristics, then a complete global model cannot provide a correct model for all partial areas.

Therefore, with the high density point clouds, we develop a hybrid model for analysing deformation:

$$\mathbf{P}^I - \mathbf{P}^{II} = F(M_1(X_1), M_2(X_2), \cdots, M_u(X_u)) + V \tag{3.16}$$

where $\mathbf{P}$ is the point clouds at two epochs ($\mathbf{P}^I$ and $\mathbf{P}^{II}$), and V is the modelling residual that also used in global modelling in Equation 2.2; $F(M_1(X_1), M_2(X_2), \cdots, M_u(X_u))$ is the hybrid model, and this hybrid model includes u sub-models for comprehensively modelling the deformations of this object; $M_i(X_i)$ is deformation models of partial areas on this object. This hybrid model can offer more precise information on the partial areas and also an informative description of the whole object.

In order to provide better explanations and understandings of this hybrid model, we subsequently define three main concepts in this hybrid modelling approach. These three concepts are: *meta-deformation*, *sub-deformation*, and *deformation map.*

3.4.1 Meta-deformation

The hybrid model (Equation 3.16) is designed to give a comprehensive description including different distinctive sub-models for partial areas of the monitored object. In some cases, the priori information of the interested/partial areas are given to analyse the deformation. But, in many real scenarios, we cannot get such a priori knowledge of the partial areas in advance. To automatically identify such partial areas, we split this object into many 3D cells and each cell has two corresponding point cloud datasets inside – at Epoch I and Epoch II respectively, e.g., C^I and C^{II} in Figure 3.9. *Meta-deformation* is proposed to describe the deformation of each pair of corresponding cells at two epochs.

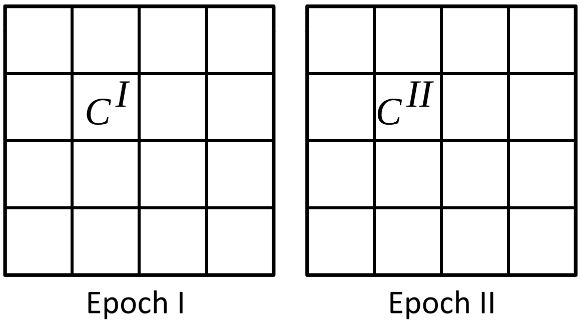


Figure 3.9: Two corresponding point cloud datasets at Epoch I and Epoch II respectively (to simply the draw, this is shown in 2D)

Definition 3.1 (Meta-deformation)**.** Let C^I and C^{II} be two correspondent split 3D cells at Epoch I and Epoch II, $C^I \in \mathbf{P}^I$ and $C^{II} \in \mathbf{P}^{II}$. The meta deformation D^C is the geometrical change between two correspondent parts C^I and C^{II}:

$$D^C = C^{II} - C^I = D^C(C^I, C^{II}) \tag{3.17}$$

D^C can apply different parameters to describe the geometrical changes, such as vertical displacement, angle variation, rotation, translation and pure deformation [Che83].

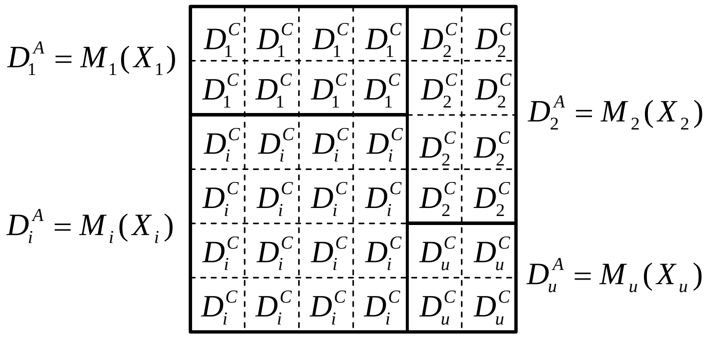
- *Vertical displacement* $D^C(V) : V = Z(C^{II}) - Z(C^I)$ are the classical description of deformation and are widely used in geodetic engineering.
- *Angle variation* $D^C(\delta) : V = \delta(C^{II}) - \delta(C^I)$ is to identify the rotation of the object and can be described as the angle between the two normals of initial and deformed surfaces.
- *Translation and rotation* $D^C(R, t) : C^{II} = [R|t]C^I$ is the rigid transformation of one cell.
- *Pure deformation* is the stress-induced change in the shape or scale of the body.

Those geometrical changes (including vertical displacement and angle variation) are applied in this study for meta-deformation estimation. This is because the cell size is reasonably small, and we can achieve a uniform geometrical change in a small cell.

3.4.2 Sub-deformation

Based on the estimation of meta-deformation for each cell by a uniform model, the adjacent cells with similar deformations are merged together as a partial area of the monitored object. We can then achieve the second level of deformation, i.e., *sub-deformation.*

Definition 3.2 (Sub-deformation)**.** A sub-deformation is dedicated to the i^{th} partial area of the object and describes the deformation property of this partial area, i.e., $D_i^A = M_i(X_i)$. Sub-deformation is typically computed by merging the adjacent cells $(C_i, \cdots, C_j)$ based on their similar meta-deformations D_i^C.



$D_1^A = M_1(X_1)$
$D_2^A = M_2(X_2)$
$D_i^A = M_i(X_i)$
$D_u^A = M_u(X_u)$
D_1^C D_1^C D_1^C D_1^C D_2^C D_2^C
D_1^C D_1^C D_1^C D_1^C D_2^C D_2^C
D_i^C D_i^C D_i^C D_i^C D_2^C D_2^C
D_i^C D_i^C D_i^C D_i^C D_2^C D_2^C
D_i^C D_i^C D_i^C D_i^C D_u^C D_u^C
D_i^C D_i^C D_i^C D_i^C D_u^C D_u^C

Figure 3.10: Deformation map of one monitored object (with u partial areas of sub-deformation)

Taking Figure 3.10 for example, cells are merged with their neighbourhood cells according to their meta-deformation. For $M_i(X_i)$, we can compute integrated deformation values based on meta-deformations of the cells $(C_i, \cdots, C_j)$ inside this partial area. In this figure, we compute u partial areas' sub-deformation, i.e., from D_1^A to D_u^A.

3.4.3 Deformation map

Based on the *meta-deformation* for small cells and the *sub-deformation* for partial areas, we are able to generate a complete deformation description to represent the whole monitored object, i.e., *deformation map.*

Definition 3.3 (Deformation map)**.** $D = \{D_1^A, D_2^A, \cdots, D_\mu^A\}$ is the description of distinctive deformations for all of the partial areas in the complete monitored object, i.e., $D_i^A = M_i(X_i)$. Each $M_i(X_i)$ is a sub-deformation model of the partial area that is composed of a group of cells $(C_i, \cdots, C_j)$. $F(M_1(X_1), M_2(X_2), \cdots, M_u(X_u))$ is the hybrid model with sub-deformation models for all of the partial areas on the monitored object.

Regarding building the deformation map of the monitored objects, there are two kinds of deformation map:

1. All of the partial areas in the monitored object have the same deformation type but with different deformation parameters defined in Equation 3.18.

$$\mathbf{P}^I - \mathbf{P}^{II} = F(M(X_1), M(X_2), \cdots, M(X_u)) + V \tag{3.18}$$

   where

   - $\mathbf{P}^I$ and $\mathbf{P}^{II}$ are two point cloud observations of the monitored object at Epoch I and Epoch II, respectively;
   - $M(X_i)$ is a sub-deformation model of a partial area inside this monitored object;
   - $F(M(X_1), M(X_2), \cdots, M(X_u))$ is the hybrid model including the same deformation type $M(X)$ with distinctive parameters X_i at different partial areas;
   - V is the residuals between the hybrid model $F()$ and the deformation of the monitored object.

2. All the partial areas in the object have different types of deformations (e.g., subsidence, slope angle, translation and rotation) defined in Equation 3.19.

$$\mathbf{P}^I - \mathbf{P}^{II} = F(M_1(X_1), M_2(X_2), \cdots, M_u(X_u)) + V \tag{3.19}$$

   where

   - $\mathbf{P}^I$, $\mathbf{P}^{II}$, V are the same symbols as in Equation 3.18;
   - $M_i(X_i)$ are sub-deformation models of partial areas inside this monitored object;
   - $F(M_1(X_1), M_2(X_2), \cdots, M_u(X_u))$ is the hybrid model including different deformation types $M_i(X)$ with distinctive parameters X_i at different partial areas.

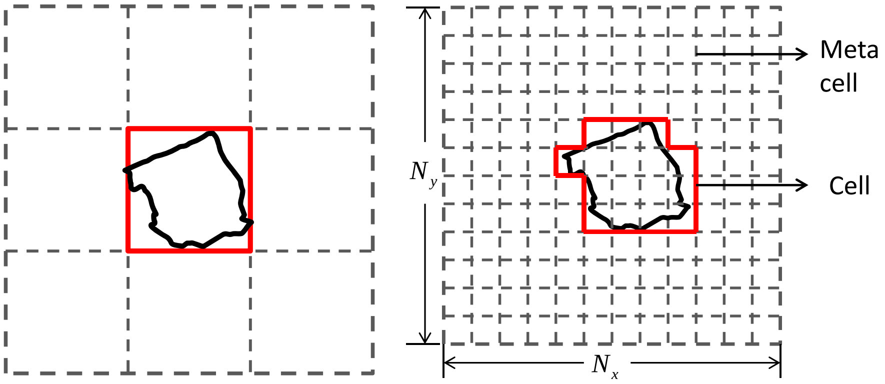
Table 3.3 summaries all of the important symbols in this chapter.

Symbol	Description
$\mathbf{P}^{I}$, $\mathbf{P}^{II}$	Two point cloud datasets of the monitored object sampled at Epoch I and Epoch II
P_i: $[x_i, y_i, z_i]$	One point including three coordinate components inside a point clouds data set
C^{I}, C^{II}	One cell with two corresponding point cloud datasets inside – respectively at Epoch I and Epoch II
$D^{C}_{i,j,k}$	Meta deformation of the $\langle i, j, k \rangle$cell
D^{A}_{i}	Sub-deformation of the ith partial area
D	Deformation map of the monitored object
$M(X_i)$	A global model for all partial areas
$M_i(X_i)$	A uniform model for the *i-th* partial area
F	Hybrid model of the monitored object

Table 3.3: Notations of symbols

3.4.4 Discussion

Previously, we proposed the hybrid model including *meta-deformation*, *sub-deformation* and *deformation map*. The meta-deformation is a uniform deformation for a single "rectangle"/"cube" cell by regularly split. The simple strategy can be applied to any shape kind of object. There is a limitation about using such regular cells for meta-deformation and deformation tracking. In real-life, object has arbitrary shape and deformation analysis should consider these irregular shapes. Therefore, we additionally introduce the "meta-cell" concept to increase the continuous attribute of deformed parts inside the monitored object and extend the hybrid model to support data-driven cells with arbitrary boundary (see Figure 3.11). This meta-cell is similar to "pixel" in image analysis, and this meta-cell will be detailedly designed in building arbitrary data-driven cells. It will be used in movement tracking as one important extension in Section 6.4.



(a) Regular cell (b) Arbitrary cell

Figure 3.11: Data-driven cell

The red cell in Figure 3.11(a) depicts the first assumption that one cell contains the deformed object. This red cell cannot accurately describe the boundary of the irregular deformed parts. There exists large empty space at the corners of this red cell. Compared with Figure 3.11(a), the red cell in Figure 3.11(b) can provide the deformed object boundary more accurately. This cell is not a regular shape (a single rectangle or cube), which is composed of multiple small cells. This kind of small cell is defined as *meta-cell.* Now, one cell composes of several meta-cells and can be any arbitrary boundary. Besides supporting arbitrary boundary, this irregular cell can also keep continuous deformation of this deformed part, i.e., movement tracking in terms of small meta-cells as the unit not a big cell. Additionally, the arbitrary boundary of the deformed part can help us to better understand its geometrical features.

3.5 Summary

This chapter presented a detailed explanation about deformation modelling. Specifically, we provided an explicit and comprehensive overview of deformation modelling requirements, including three main parts, i.e., *what has deformation*, *which kind of deformation type*, and *how to describe deformation.* For deformation objects, we presented two main kinds of objects such as artificial structure and natural objects, and studied their common geometrical features. For deformation types, we illustrated geometric changes of an object in terms of atomic deformation type (e.g., rigid and non-rigid deformation) and complex deformation. Regarding deformation parameters, we demonstrated general deformation representation methods, in terms of both *quantitative* numeric values and *qualitative* based semantic description.

After investigating a comprehensive modelling description, we analysed the global model based on low-sampling datasets (i.e., discrete/sparse points) collected by classical geodetic methods (e.g., GNSS) and observed that the global model can not satisfy the requirements driven by high density point cloud datasets. Therefore, we developed a hybrid model including *meta-deformation*, *sub-deformation*, and *deformation map* to describe deformation from cell to the whole object. Meta-deformation is the geometrical change of small cells between two epochs; sub-deformation is the deformation description of partial areas inside the object; deformation map is the composed deformation description for the whole region. Additionally, we extend this hybrid model to support the arbitrary boundary cell with continuous deformation and geometric features.

Chapter 4

Cell-based Deformation Computation

> *A journey of a thousand miles*
> *begins with a single step.*
>
> *Laozi (∼571 BC-471 BC)*

4.1 Introduction

In Chapter 3, we discussed the deformation modelling, with a specific focus on the "input" and the "output" of deformation monitoring. In this chapter, we present a supportive computing solution to transfer the input (i.e., high density point cloud datasets) to the output (i.e., the informative deformation map). In particular, we design a novel *cell-based* deformation computing framework that is able to extract hybrid deformation in a large monitored region. This is a novel approach for deformation monitoring in Earth environment using LiDAR and can generate an informative deformation description automatically. Distinctive deformation characteristics for different partial areas are extracted, without given *a priori* knowledge.

This chapter is organised as follows: Section 4.2 provides a generic problem statement of deformation computation, and introduces our cell-based approach that has three main steps, i.e., "split", "detect", and "merge". Section 4.3 concretely presents the "split" step, which divides the whole region into uniform cells using the R-tree indexing technique. Section 4.4 focuses on the detailed deformation detection algorithms for individual cells, where the deformation can be extracted via a single uniform model. Section 4.5 discusses the final "merge" step that is able to group the cells into partial areas according to the deformation status, for generating the final deformation map. Finally, Section 4.6 gives a summary of this cell-based deformation computation approach.

4.2 Deformation Computation

In this section, we firstly present the problem statement of deformation computation, secondly discuss the motivation of designing a cell-based approach, and finally explains our cell-based deformation computation framework.

4.2.1 Problem statement

Deformation computation is the process of designing relevant algorithms to automatically extract the deformation description for a given monitored region with two epochs of points clouds from LiDAR. During the deformation computing process, relevant deformation description can be represented as the models defined in Chapter 3, particularly the final deformation map. Therefore, this "deformation computation" problem has the following input and output:

- *Input*: the two epochs of scanned point clouds by LiDAR and each epoch contains a large set of 3-dimensional data points, i.e., $\mathbf{P}^I = \{P_1^I, P_2^I, \cdots, P_{n_I}^I\}$, $\mathbf{P}^{II} = \{P_1^{II}, P_2^{II}, \cdots, P_{n_{II}}^{II}\}$, where P_i is a single point $[x_i,\, y_i,\, z_i]$.
- *Output*: the complete "deformation map" for this area, i.e., $\mathbf{D} = \{D_1^A, \cdots, D_m^A\}$, where D_i^A is the i^{th} partial area A_i with a uniform deformation description. Therefore, this output "deformation map" ($\mathbf{D}$) supports hybrid and distinctive deformation description in the complete area.

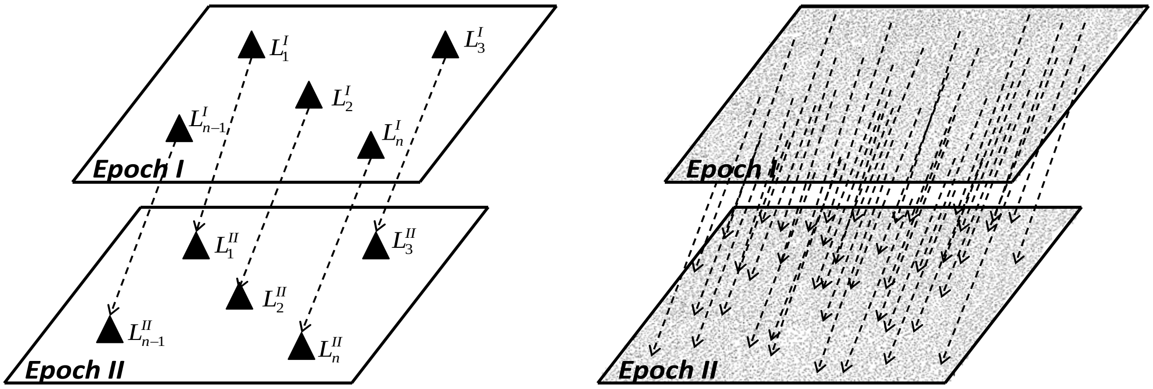
4.2.2 Why cells for deformation computation?

We now discuss the motivation of designing a cell-based approach for deformation computation, as well as the benefits from using cells.

Potentiality of Point Clouds - Recently, the development of the terrestrial LiDAR technology provides ability to efficiently capture a huge amount of point clouds on the monitored object, as shown in Figure 4.1(b). This is in contrast to traditional geodetic surveying techniques that only capture a small set of discrete points as shown in Figure 4.1(a). The new point cloud observations of the monitored object become highly-dense 3D points at Epoch I ($\mathbf{P}^I = \{P_1^I, P_2^I, \cdots, P_{n_I}^I\}$) and Epoch II ($\mathbf{P}^{II} = \{P_1^{II}, P_2^{II}, \cdots, P_{n_{II}}^{II}\}$). These point clouds can capture a very high coverage on all of the partial areas of this object. Point clouds with high redundancy in spacial dimensions show the great potentiality for richer deformation detection. Point clouds provide the possibility to expand the description of deformation of this region from *the displacement of selected discrete points* to *the displacement of the whole surface.*

Challenges of Detection via Point Clouds - Given two point clouds of the same region collected at two epochs ($\mathbf{P}^I$ and $\mathbf{P}^{II}$) in Figure 4.1(b), the traditional method for extracting the displacement of this surface is to find each pair of corresponding points at two epochs (i.e., P_i^I and P_i^{II}) and then compare these two points directly. However, this method is impractical for such huge mount point cloud data because of the following aspects:

- *Point-to-point correspondence searching* - For traditional displacement-based approaches, it is hard to find the corresponding pairs of points for two huge point cloud data sets, i.e., point $P_i^I = [x^I, y^I, z^I]$ in Epoch I corresponds to point $P_i^{II} = [x^{II}, y^{II}, z^{II}]$ in Epoch II. Even for a stable object without any deformation that was scanned twice by a terrestrial



(a) Discrete points

(b) Point cloud

Figure 4.1: Displacement detection. Sub-figure 4.1(a) is detection with geodetic network; sub-figure 4.1(b) is detection with point clouds.

laser scanner located at the same position, point clouds captured at two epochs cannot be completely uniform; therefore, it is impossible to find all point correspondence from two different epochs. Actually, such one-to-one correspondence between P_i^I and P_i^{II} does not exist for all data points. In addition, it is also hard to validate such correspondence because the data points are very close to each other as the high density in both $\mathbf{P}^I$ and $\mathbf{P}^{II}$. Nevertheless, the surface registration techniques does not necessarily require searching point-to-point correspondence.

- *Time consuming* - Point-to-point correspondence searching is a time-consuming procedure for such a huge dataset. By using a non-index searching algorithm (i.e., exhaustive searching), the time complexity of searching a corresponding point P_i^{II} from the dataset $\mathbf{P}^{II}$ at Epoch II is $O(n_{II})$, where n_{II} is the size of $\mathbf{P}^{II}$. For finding all of the correspondence between $\mathbf{P}^I$ and $\mathbf{P}^{II}$, we need to have a complexity of $O(n_I \times n_{II})$, where n_I and n_{II} are the point number of $\mathbf{P}^I$ and $\mathbf{P}^{II}$, respectively. By using an indexing technique (e.g., a r-tree, or a k-d tree structure), we can achieve better performance, i.e., $O(\log n_I \times \log n_{II})$ for finding all point-to-point pairs; however, this is still very time consuming. But fir

- *Less support of using specific point characteristics* - In real-life data sets, there exist a couple of meaningful information like intensity, colour, etc. Such information can be used to improve or guarantee the accuracy of correspondence between two points at two epochs. Positions of these specific points are quite useful and important, especially in tracking the single point from the datasets in future epochs. However, as the data point density is so large in point cloud, tracking these specific point characteristics for finding point correspondence is still quite inflexible and expensive.

Benefits of Cell-based Approach - We introduce the “cell” concept into our approach to meet the potentialities and the challenges brought by point clouds. Cell is a well-divided geometric shape in space, e.g., a rectangle in 2D space and a cube in 3D space. We divide the space that comprises the monitored object into cells, which can have uniform or varying size. By analysing the cells rather than the points directly, the cell-based approach can provide effective solutions to deal with the challenges mentioned previously:

- *Cell-to-cell correspondence* - Different from the direct point-to-point correspondence searching, this approach focuses on finding the cell pair in two epochs, i.e., cell-to-cell correspondence searching. Due to the high density of points, correspondence searching for all points is impracticable; on the contrary, there is non-overlap between cells; therefore, it is easier to find cell-to-cell correspondence. There are three kinds of cell-to-cell correspondence searching: (1) *Direct correspondence* – this is for some deformation scenarios with micro deformation, where the deformation is small and the points/object keep inside one cell at different epochs. In Chapter 5, the experiment will be focusing on the direct cell correspondence. (2) *Non-direct correspondence* – this can work for large deformation scenarios, which is suitable for the cases when the physical deformed object materials move across cell boundaries (e.g., from one cell to another). Section 6.3 will present the non-direction cell correspondence searching method. (3) *Overlapping cell correspondence* – Furthermore, cell correspondence can have overlap, e.g., $cell_1$ at Epoch 1 is deformed to the boundary part of both $cell_1$ and $cell_2$ with overlap (or the boundary of any $cell_i$ and $cell_{i+1}$) at Epoch 2. In such case, richer cell correspondence needs to be designed. In such case, data-driven cell is developed to support deformed parts boundary detection and its movement tracking. The data-driven cell is composed of multiple meta-cells that can form an irregular cell with arbitrary shape. The details of such data-driven cell will be discussed in Section 6.5.

- *Efficiency of detection* - The space of the monitored object is divided into many cells. Obviously, the number of these cells is significantly less than the number of points. N_c^I and N_c^{II} are the number of the cells in this space at Epoch I and Epoch II, respectively; and n_I and n_{II} are the number of the points in the same space at Epoch I and Epoch II, respectively. Due to the high density of point clouds, of course $N_c^I << n_I$ and $N_c^{II} << n_{II}$; in other words, the cell number in both epochs is significantly less than the point number. Therefore, the time complexity of cell-to-cell correspondence searching is $O(N_c^I \times N_c^{II})$ without index (or $O(\log N_c^I \times \log N_c^{II})$ with index like k-d tree) is significantly less than the time complexity of point-to-point correspondence searching, i.e., $O(n_I \times n_{II})$ without index (or $O(\log n_I \times \log n_{II})$ with index).

- *Cell characteristics for supporting correspondence searching* - The characteristics of cells can be calculated using the points inside. The group of points in two cells can formulate certain characteristics, e.g., similar normal, distinct curvature, etc. To identify cell characteristics, in Section 4.4, we will focus on modelling individual cells via a uniform model like a surface regression to find common displacement, a ICP (Iterative Closet Point) to find rigid transformation. These computed cell characteristics can be used to support the cell-correspondence searching. In Chapter 6, we will discuss applying the ICP-based rigid transformation to search cell correspondence in a rockfall deformation scenario.

4.2.3 Computing framework

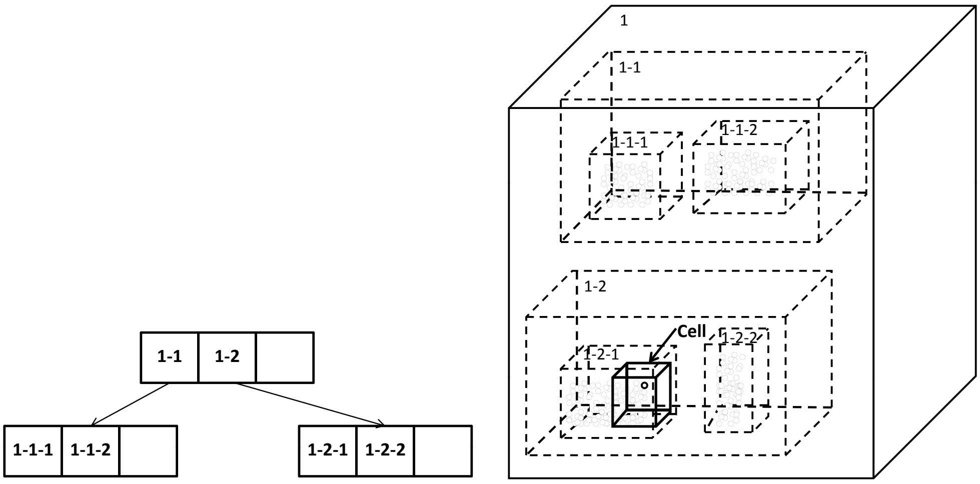
So far, we have already identified the importance of using "cells" to compute a comprehensive deformation map for describing the deformation of a large monitored region. Hereby, we design the complementary technical framework of such "cell-based deformation monitoring". This framework is designed for computing deformation description with regards to the hybrid model of the monitored object. The input of our approach is the point cloud datasets of the monitored object captured by terrestrial LIDAR at Epoch I and Epoch II. The output of this approach is the computed deformation map comprising distinctive sub-deformations for all of the partial areas of the monitored object. The framework of the proposed cell-baesd deformation computation is sketched in Figure 1.1. Between the input and the output, there are three main steps, i.e., *split*, *detect*, and *merge*.

- **Split** - This step is to divide the space of the monitored object into 3D uniform cells and capture their corresponding point clouds in Epoch I and Epoch II, respectively. We apply the R-tree spatial indexing technique to enable fast access of the point clouds in individual cells. In addition, we can transfer the three dimensional index to one dimensional index. Section 4.3 will provide the detailed algorithms for the split part.
- **Detect** - This step is to estimate the meta-deformation for each cell in the monitored object by comparing the corresponding point clouds in the cell at two epochs. Such meta-deformation can be vertical displacement, angle variation, and rigid transformation. Section 4.4 will provide the detailed estimation algorithms for the detect part.
- **Merge** - This step is to group the adjacent cells with consistent deformation behaviours (i.e., the meta-deformations between neighbouring cells are similar) together as a partial area, and additionally compute the sub-deformation for each partial area. Finally, the sub-deformation of partial areas compose the complete deformation map. Section 4.5 will provide the detailed merging algorithms.

4.3 Step I - Split

In the real-life experiment using point cloud data from LiDAR to monitor deformation, the number of 3D points captured by LiDAR is typically huge. For example, in our Flamatt dataset about a landslide region of size 200 $m \times 25$ m, there are about 20 millions 3D points at Epoch I and about 10 million 3D points at Epoch II. It is non-trivial to handle such a big amount of point cloud data in an very efficient way. Deformation analysis should be able to quickly identify interesting parts of the object, i.e., the subset of the complete point clouds. In our cell based approach, we design the "*split* " step to divide the space of the whole object into 3D uniform cells. With cells, detection algorithms are able to deal with a small part of the points. To efficiently access the data points inside a cell, we apply relevant *acceleration strategy.* This is called the indexing technique from the computer science field.

In this step, we apply the R-tree indexing technique [MNT05] to efficiently manage the point cloud data. R-tree is the tree data structure that is widely used for indexing multi-dimensional spatial data such as geographical coordinates, rectangles or polygons. R-tree can be considered as a hierarchical data structure based on B$^+$-tree in multi-dimensional spaces. Figure 4.2 shows an example of R-tree structure. The point clouds in the whole object are built into two subspaces $R1$-1 and $R1$-2, based on the data density. Subsequently, the point clouds in these subspaces can be split into more detailed levels of subspaces subsequently, i.e.: $R1$-1 is further divided into $R1$-1-1 and $R1$-1-2, $R1$-2 is further divided into $R1$-2-1 and $R1$-2-2. As shown in Figure 4.2(b), to access the cell, we can search starting from the root of the tree in Figure 4.2(a). Compared with non-index methods that access the point clouds directly via coordinates of these data points, it is much faster to access one given cell by using this R-tree structure. In addition, there are largely available open-sources that can be reused or easily adapted for building the R-tree structure on our point cloud datasets. As we do not need to update the point cloud data, the only focus here is on efficiently accessing the point clouds in the cells for computing deformation. R-tree has good performance in such cell-based spatial data query [MNT05].



(a) (b)

Figure 4.2: R-tree structure for spatial access. (a) is a R-tree structure; (b) is the visualization of this R-tree by 3D cubes.

Split Object Space into Cells - We apply the R-tree spatial access technique to manage the point cloud data. After loading a mass point cloud dataset $\mathbf{P}$, a R-tree structure of $\mathbf{P}$ is built (like 4.2). According to all of the coordinates of $\mathbf{P}$ in the R-tree, the maximum and minimum along x-, y- and z- axes (i.e., $min_x, min_y, min_z, max_x, max_y, max_z$) can be quickly extracted and a bounding box that includes all of the point cloud is created. The size of the cells are user-defined (*length, width, height*); and then the number of the cells along x-, y- and z- axes (N_x, N_y, N_z) can be calculated (see Equation 4.1).

$$\begin{pmatrix} N_x \\ N_y \\ N_z \end{pmatrix} \leftarrow \begin{pmatrix} (max_x - min_x)/length \\ (max_y - min_y)/width \\ (max_z - min_z)/height \end{pmatrix} \tag{4.1}$$

According to $\{min_x, min_y, min_z, max_x, max_y, max_z\}$ and $\{N_x, N_y, N_z\}$, we can now compute the bound of all cells: the cell $cell_{ijk}$'s bound Box_{ijk} is defined as follows,

$$Box_{ijk} \leftarrow \begin{pmatrix} min_x + i * length, min_x + (i+1) * length \\ min_y + j * width, min_y + (j+1) * width \\ min_z + k * height, min_z + (k+1) * height \end{pmatrix} \tag{4.2}$$

With such bound, the point clouds in Box_{ijk} can be efficiently accessed by using R-tree index from the complete data set of $\mathbf{P}$. The cells can be numbered by the 3D index along X, Y and Z axes as $Cell\ {}^{(3)}_{ID}(id) = \langle i, j, k \rangle$, i.e. the cell identity (id). Such three dimensional (3D) identity can be refined into 1D identity $Cell\ {}^{(1)}_{ID}(id)$ by Equation 4.3.

$$Cell^{(1)}_{ID}(id) = i + j * (N_x + 1) + k * (N_y + 1) * (N_z + 1) \tag{4.3}$$

The detailed procedure of *3D cell splitting* is shown in Algorithm 4.1: firstly, we load all of the 3D data points in the point cloud dataset; secondly, we build the R-Tree index for the data points that are previously loaded, and compute the bounding boxes (min, max values for each dimension); afterward, based on the min, max values as well as the cell size given, we compute the number of cells (N_x, N_y, N_z); finally, for each cell C_{id}, we can efficiently output the split data points based on the indexing structures of the R-Tree. For the sake of searching time by R-tree, we output the point clouds for each cell, and these points for each cell can be directly used in the later detection step in Section 4.4. In such case, we do not need to re-access the data every time when run the detection algorithms for computing meta-deformation.

Algorithm 4.1: *Split*

```
Input: a file of original point cloud data set (filename), cell size (length, width, height)
Output: 3D split cells $\{C_{idMin}, \cdots, C_{idMax}\}$
begin
    /* load all the points into P */
    $\mathbf{P} = \{P_1, \cdots, P_N\} \leftarrow$ loadFile(filename);
    /* build index based on R-tree */
    pointRtree $\leftarrow$ buildRTreeindex(P);
    /* find the bound of point cloud */
    boundP $\leftarrow \{min_x, min_y, min_z, max_x, max_y, max_z\}$;
    /* calculate cell numbers */
    $(N_x, N_y, N_z) \leftarrow$ getCellNumber(boundP, cellSize); // by Equation 4.1
    /* group point w.r.t the bound of each cell by R-tree */
    forall the $([i, j, k] \in [0 : N_x, 0 : N_y, 0 : N_z])$ do
        $Box_{ijk} \leftarrow$ getCellBound(); //by Equation 4.2
        $C_{ijk} \leftarrow$ getPointInBox(pointRtree, $Box_{ijk}$);
        /* give the identity for each cell */
        switch the_dimension_of_index do
            case 3D identity
                id $\leftarrow$ get3DIdentiy$(i, j, k)$;
            case 1D identity
                id $\leftarrow$ get1DIdentiy$(i, j, k)$ by Equation 4.3;
    return 3D split cells $\{C_{idMin}, \cdots, C_{idMax}\}$
```

4.4 Step II - Detect

In the previous "split" step, we have already divided the point clouds sampled on the whole object into 3D cells, for both data sampling at Epoch I and Epoch II. Figure 4.3 sketches the two point cloud datasets at two epochs. Black points are captured from a unstable slope at Epoch I before deformation happened; while, red points are collected from the same slope at Epoch II after the deformation. To simplify the explanation, we divide these areas into only four cells, from *Cell* 1 to *Cell* 4, by using the previous "split" algorithm.

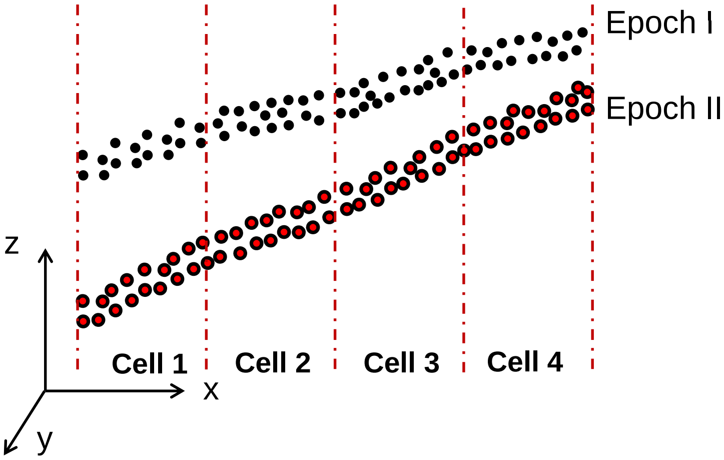
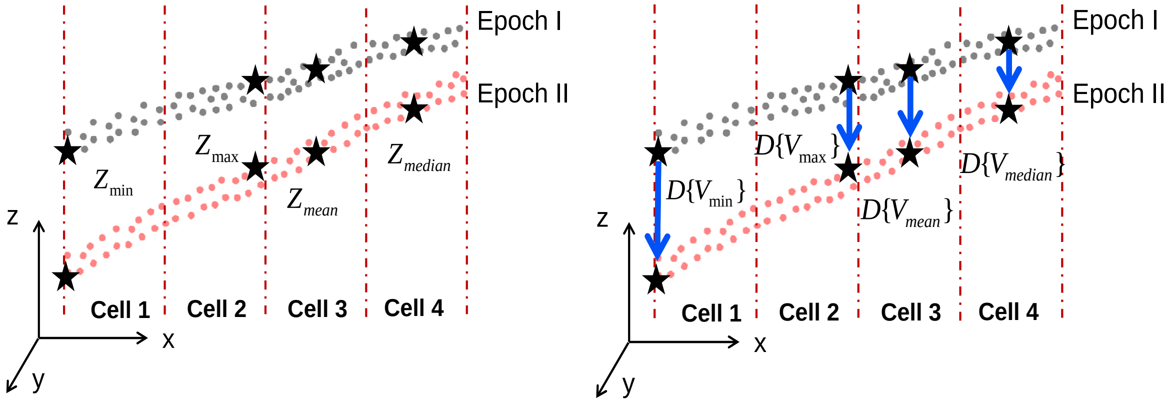


Figure 4.3: Point clouds in cells after splitting

Based on the 3D points in each cell, the objective of the "detect" step is to compute the meta-deformation for each cell. Such meta-deformation indicates the deformation magnitude for each cell. In this section, we present the detailed estimation methods of computing both *atomic deformation* (such as displacement and angle variation) and *complex deformations* (such as rigid transformation). Some detailed formula description of these deformation types have already been introduced in Section 3.2. In this section, we focus on technical process to extract these values from 3D points in the cell.

Vertical displacement – In traditional discrete point-to-point deformation analysis, we usually compute vertical displacement by comparing the Z-coordinate values of the points at two epochs. Therefore, for each cell, we need to calculate a statistical parameter to represent its altitude information (i.e., the Z-axis), for the subsequent estimation of vertical displacement. There are several ways to approximate this parameter, using features such as *min*, *max*, *mean*, *median* and *centre* of each cell's Z value.

The first four methods use standard statistical properties (i.e., *min*, *max*, *mean*, *median*). We apply these baseline methods to approximate the vertical information of each cell. In Figure 4.4, vertical approximation of each cell is shown by black stars, which are chosen by the estimation methods to represent the cells. In *Cell* 1 of Figure 4.4(a), two star points, $Z_{min}(C^{I})$ and $Z_{min}(C^{II})$, are selected by the *minimum* method, which picks up the minimum values along Z-direction amongst all of the data points in the two point sets at Epoch I and Epoch II, respectively. Similarly for other methods like *max*, *mean*, *median*, they pick up relevant data point corresponding to the *max*, *mean*, *median* values of the Z coordinate of all data points in the cell. As shown in Figure 4.4(a), the star points in *Cell* 2, *Cell* 3, *Cell* 4 are calculated by the *max*, *mean*, *median* methods, respectively. Compared to other methods, the *min* method is robuster with good performance, as it can avoid the noisy data (outliers) above the slope, e.g., vegetation on the ground such as bush or grass.



(a) Baseline methods

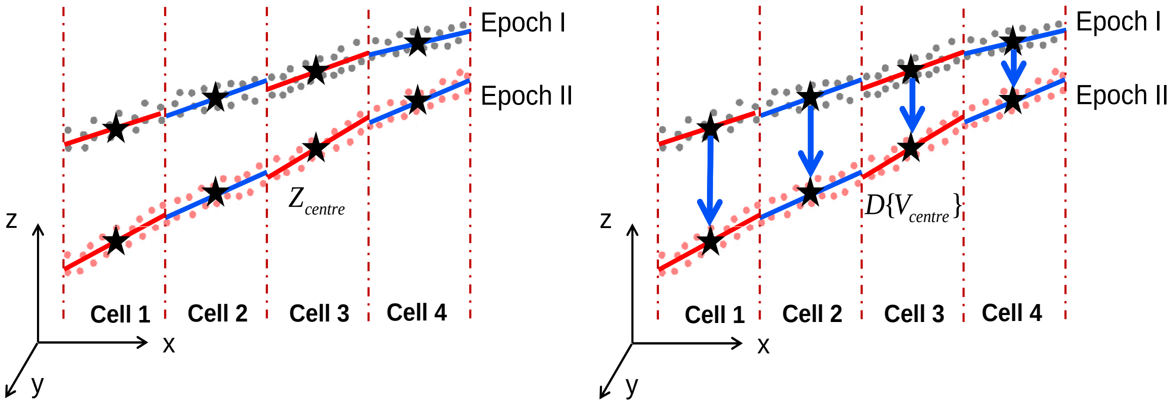
(b) Estimation displacement

Figure 4.4: Vertical information representation for cells. Sub-figure 4.4(a) is attitude approximation by baseline methods; sub-figure 4.4(b) is the estimation of displacement for each cell.

After determining the estimation of cell altitude (by the *min*, *max*, *mean*, *median* methods), we can easily identify the displacement between two epochs for each cells. Taking the *min* method for instance, we compare the minimum approximation value at Epoch II (i.e. $Z_{min}(C^{II})$) and the minimum value at Epoch I in the same cell (i.e. $Z_{min}(C^{I})$). The vertical displacement $D^{C}_{id}\{V_{min}\}$ can be calculated by Equation 4.4. As shown in Figure 4.4(b), the cell displacement can be estimated using the "altitude" (calculated by the four methods in Figure 4.4(a)) difference between the corresponding cells at Epoch 1 and Epoch 2.

$$D^{C}_{id}\{V_{min}\} = Z_{min}(C^{II}_{id}) - Z_{min}(C^{I}_{id}) \tag{4.4}$$

In addition to standard methods like *min*, *max*, *mean*, *median*, we also design another altitude estimation using the "centre" point estimated from all point clouds in the cell. To compute such centre, we need to do surface regression first; therefore, we design the *planeFit* function in Algorithm 4.2, which aims to find an optimal fit plane that can minimise the sum of squared errors of all points in the cell. The solid lines in Figure 4.5(a) are planes fitted to the points at two epochs, by using the "plane fit" function. On these planes, the Z-value at the centre of cells (i.e., $Z_{centre}(C)$) are considered as the vertical information to represent these cells; therefore, this estimation method is entitled "centre". When the cell size is reasonably small, we can build a better fitting plane of the points in cells. Algorithm 4.2 illustrates the procedure of fitting a plane to the points inside a cell. Output of this algorithm are centre approximated coordinates of the cell $Centre\{C_i\}$ and the normal of the best fit plane. We can apply polynomial hyperplane rather than 1-order plane fitting, and calculate the deformation with the centre on the plane or regression coefficients. In Chapter 6, we will discuss higher-order polynomial regression.



(a) Centre with plane fit

(b) Estimate displacement

Figure 4.5: Vertical displacement detection for cells using *centre*. Sub-figure 4.5(a) show the fitting plane and the centre point for each cell at two epochs; sub-figure 4.5(b) is the displacement estimated by the *centre* method.

Algorithm 4.2: *planeFit*

```
Input: One cell $C_{id}$
Output: Statistical magnitude at the centre of the cell $Centre\{C_{id}\}$; Normal of the best-fit plane.
begin
    /* get all points inside this cell by R-tree */
    P ← lengthofPointsInCell($C_{id}$);
    /* get the length of points in this cell */
    $n$ ← lengthofPointsInCell(P);
    if $n < 3$ then
        Error(Less than three points for plane fitting!);
    /* Calculate the centroid of points in this cell */
    $Centroid$ ← mean(P);
    /* Translate points in this cell to the centroid frame; */
    $\mathbf{P}_{Centroid} \leftarrow \mathbf{P} - Centroid$;
    /* Singular value decomposition of $P_{Centroid}$ */
    $[u, s, v]$ ← svd($P_{Centroid}, 0$) ;
    /* Find the smallest singular value in s */
    $[S, i]$ ← min($diag(s)$) ;
    /* Get the normal to the best-fitting plane (the corresponding singular vector) */
    $Normal \leftarrow v(i)$ ;
    /* Return the approximation of the cell and the normal of the best fitting plane */
    return $\langle Centre\{C_i\}, Normal\rangle$
```

Similar to the standard methods (*min*, *max*, *mean*, *median*), the *centre* method calculates the vertical displacement using the distance between the approximation of the *centre* from the fitting plane at Epoch 1 and the *centre* from the fitting plane at Epoch 2, as shown in Figure 4.5(b). The total least squares method via SVD can not work well on the point clouds with too many outliers. Therefore, we additionally apply a robuster plane fitting method *RANSACplaneFit*. The RANSAC (Random Sample Consensus) algorithm is an iterative method to fit a model [FB81], which is reliable even for the point clouds with a high proportion of outliers. The

Algorithm 4.3: *RANSACplaneFit*

```
   Input: One cell $C_{id}$
   Output: The best fitting plane for the points inside this cell
1  begin
2      /* get all points inside this cell by R-tree */
3      $\mathbf{P} \leftarrow$ getPointsInCell($C_{id}$);
4      /* get the length of points in this cell */
5      $n \leftarrow$ lengthofPointsInCell($\mathbf{P}$);
6      /* estimate the maximum iteration number $k$ */
7      $k \leftarrow$ round$\frac{\log(1-\alpha)}{\log(1-(1-w)^n)}$; // $\alpha$ is the probability of success; $w$ is the outlier percent number; $n$ is the inlier sample number;
8      $bestError_{plane} \leftarrow \infty$; // initialise the error of this plane
9      while $i < k$ do
10         /* randomly select $n$ points from $\mathbf{P}$ */
11         inliersP $\leftarrow$ randomSelectPoints($\mathbf{P}$);
12         /* compute a plane to fit inliersP */
13         $Plane_{current} \leftarrow$ planeFit(inliersP);
14         $consensusSet \leftarrow$ inliersP;
15         forall the point $Q \in$ ($\mathbf{P}$ − inliersP) do
16             /* compute the distance from the point $Q$ to the current plane */
17             $dis_Q \leftarrow$ dis2plane($Q$);
18             /* $t$ is the threshold to determine the point belonging to the fitting plane */
19             if $dis_Q < t$ then
20                 $consensusSet \leftarrow consensusSet \cup Q$;
21         /* $d$ is the minimum point number to decide the current plane is good enough */
22         if $|consensusSet| > d$ then
23             if $Error_{plane} < bestError_{plane}$ then
24                 BestPlane $\leftarrow Plane_{current}$;
25                 $bestError_{plane} \leftarrow Error_{plane}$;
26     /* Return the best fitting plane */
27     return $BestPlane$
```

objective of RANSAC method is to find the best plane in a dataset of point cloud. The detailed procedure is shown by Algorithm 4.3. Firstly, it randomly selects n points and calculates a plane to fit these n points (line 10 - 14). Secondly, it judges whether the rest of the points belong to this plane, according to a given threshold (line 15 - 20). The two steps are iteratively performed; and in each iteration, the obtained plane is compared with the last best one. If the current plane is better than the last best one, the current plane will replace it as the new best plane (line 21 - 25). Besides plane fitting, the RANSAC method can also detect other basic geometric shapes from point clouds, like spheres, cylinders and cones [SWK07].

With the plane-fitting algorithm (either the SVD-based method or the RANSAC one), the detailed algorithm of computing meta-deformation using vertical displacement is summarised in Algorithm 4.4: firstly, points inside the cell at Epoch I and Epoch II are retrieved by the associate index (based on the R-tree structure and the output of the “split” step in Section 4.3); secondly, according to the chosen method (*min*, *max*, *mean*, *median*, or *centre*), vertical displacement is calculated by dedicated estimation method; finally, the output is vertical displacement for each cell between two epochs.

Algorithm 4.4: ***getVerticalDisplacement***

```
Input: Two corresponding cells $C_{id}^{I}$ and $C_{id}^{II}$ at two epochs,
       A chosen approximation method $md \in \{min, max, mean, median, centre\}$
Output: Vertical displacement for the corresponding cells $\{C_{id}^{I}, C_{id}^{II}, D_{id}^{C}(V)\}$
begin
    /* get the points of correspondent cells */;
    $P_{id}^{I} \leftarrow$ PointsInCells($C_{id}^{I}$);
    $P_{id}^{II} \leftarrow$ PointsInCells($C_{id}^{II}$);
    /* calculate the statistical parameter to approximate this pair of cells */;
    switch $md$ do
        case $min$
            $D_{i}^{C}\{V_{min}\} \leftarrow$ min($P_{id}^{II}$) - min($P_{id}^{I}$) ;
        case $max$
            $D_{i}^{C}\{V_{max}\} \leftarrow$ max($P_{id}^{II}$) - max($P_{id}^{I}$) ;
        case $mean$
            $D_{i}^{C}\{V_{mean}\} \leftarrow$ mean($P_{id}^{II}$) - mean($P_{id}^{I}$) ;
        case $median$
            $D_{i}^{C}\{V_{median}\} \leftarrow$ median($P_{id}^{II}$) - median($P_{id}^{I}$) ;
        case $centre$
            /* Find a best fitting plane for the points in each cell*/;
            $Centre^{I}\{C_{id}\} \leftarrow$ planeFit($P_{id}^{I}$);
            $Centre^{II}\{C_{id}\} \leftarrow$ planeFit($P_{id}^{II}$);
            $D_{i}^{C}\{V_{centre}\} \leftarrow Centre^{II}\{C_{id}\}$ - $Centre^{I}\{C_{id}\}$ ;
    return Vertical displacement $\{C_{id}^{I}, C_{id}^{II}, D_{id}^{C}(V)\}$
```

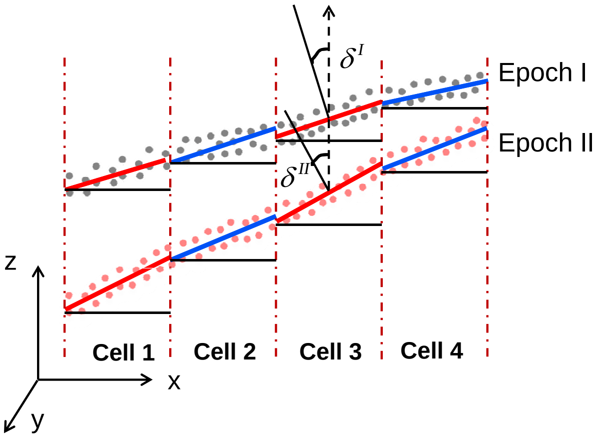


Figure 4.6: Angle representation for cells

Angle variation – To detect the angle variation, a plane is fitted based on the point cloud at each epoch in a cell (i.e., using the *planFit* function in Algorithm 4.2), like the vertical displacement using the centre of the cell. Different from the centre method using a centre value of Z-axis to approximate the statues of each cell, the angle variation method computes the angles between the fitted plane and the horizontal surface. Based on the *normal* returned by Algorithm 4.2 for the cell at two epochs, we can calculate the difference of such angles, and apply this as the meta deformation for this cell. As shown in Figure 4.6, the 8 solid lines are the best fitting planes for 3D points (for 4 cells at 2 epochs). δ^{I} and δ^{II} are the angle between the fitting plane and the horizontal surface at Epoch I and Epoch II. Calculating angle difference between δ^{I} and δ^{II} is the objective of this angle variation detection.

Equation 4.5 shows two fitting plane for 3D points in the cell C_{id} at Epoch I and Epoch II. For simplicity, we can use the vector of four coefficients to indicate such surface, i.e., $\langle a^{I}_{id}, b^{I}_{id}, c^{I}_{id}, d^{I}_{id}\rangle$ and $\langle a^{II}_{id}, b^{II}_{id}, c^{II}_{id}, d^{II}_{id}\rangle$ for the two epochs, respectively.

$$\begin{aligned} C^{I}_{id} &: a^{I}_{id}x + b^{I}_{id}y + c^{I}_{id}z + d^{I}_{id} = 0 \\ C^{II}_{id} &: a^{II}_{id}x + b^{II}_{id}y + c^{II}_{id}z + d^{II}_{id} = 0 \end{aligned} \tag{4.5}$$

Based on the two planes best fitted to the point clouds in one cell, we calculate the normals of these two planes at Epoch I (i.e., $normal^{I} = (n^{I}_{x}, n^{I}_{y}, n^{I}_{z})$) and Epoch II (i.e., $normal^{II} = (n^{II}_{x}, n^{II}_{y}, n^{II}_{z})$). Meta-deformation $D^{C}_{id}\{\delta\}$ is designed as the angle $\widehat{\delta}$ between the normals of two planes at two epochs and calculated by Equation 4.6.

$$D^{C}_{id}(\delta): \ \widehat{\delta} = \delta(C^{II}) - \delta(C^{I}) = \arccos(normal^{I} \cdot normal^{II}) \tag{4.6}$$

The detailed procedure of computing angle variation is presented in Algorithm 4.5. After receiving the points in this cell at two epochs from the "split" outputs, "planeFit" function is used to approximate the surfaces at two epochs and calculate the normal of the surface. Finally, the angle difference between the two normals is extracted as the meta-deformation.

Algorithm 4.5: *getAngleDifference*

```
Input: Two corresponding cells $C^{I}_{i}$ and $C^{II}_{i}$ at Epoch I and Epoch II, respectively
Output: Angle variation for correspondent split cells $\{C^{I}_{id}, C^{II}_{id}, D^{C}_{id}(\delta)\}$
begin
    /* get the points of correspondent cells */;
    $\mathbf{P}^{I}_{id}$ ← pointsInCells($C^{I}_{id}$);
    $\mathbf{P}^{II}_{id}$ ← pointsInCells($C^{II}_{id}$);
    /* get the normals of the best-fit planes for two epochs */;
    $normal^{I}$ ← planeFit($\mathbf{P}^{I}_{id}$);
    $normal^{II}$ ← planeFit($\mathbf{P}^{II}_{id}$);
    /* calculate the angle between the two normals for two epochs */;
    $D^{C}_{id}(\delta) \leftarrow \arccos \frac{normal^{I} \cdot normal^{II}}{|normal^{I}||normal^{II}|}$;
    return Angle difference for correspondent split cells $\{C^{I}_{id}, C^{II}_{id}, D^{C}_{id}(\delta)\}$
```

Rigid Transformation - Vertical displacement and angle variation are the two most representative for detecting the meta-deformation. They stand for detecting "translation" and "rotation", the two widely-used atomic deformation types. In this paragraph, we discuss a complicated deformation type, i.e., the rigid transformation that consists of both translation and rotation during the deformation procedure. We apply the well-known ICP (Iterative Closest Point) method to extract the rigid transformation parameters for each cell. ICP is initially developed for shape registration in a 3-D space such as free-from curves and surfaces [BM92]. These objects with the common parts, even containing some noises and small differences, still can be matched together in a short time. The ICP method can utilise the 3D geometry of the object, without priori assumption of surface model (e.g., a plane in displacement detection). Furthermore, this method can extract the complex deformation such as translational and rotational displacement at the same time, which is suitable for our problem. Therefore, we combine ICP method with our cell-based approach and name it as the "Cell-based ICP" method. We apply the cell-based ICP method to match point sets in each pair of corresponding cells and extract the suitable rigid transformation parameters for this cell.

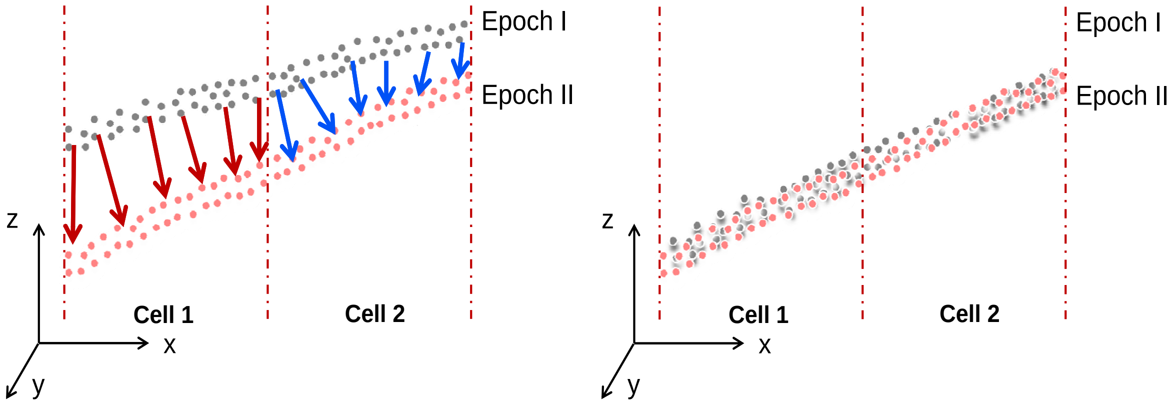
The main procedure of cell-based ICP method is summarised in Algorithm 4.6. The input of this algorithm is a pair of correspondent cells C_i^I and C_i^{II} at two epochs; and the output is rigid transformation $D_i^C\{Rt\}$ for this cell. This algorithm include two main stages: the first stage (line 2-8) is "data preparation", which is used to get two point sets $\mathbf{P}^I$ and $\mathbf{P}^{II}$ and preset the initial transformation parameters and error threshold; and the second stage is "iteration" (line 9-17), which is the crucial part of this algorithm. After getting the closest points between two point sets $\mathbf{P}_k^I$ and $\mathbf{P}^{II}$, rigid transformation parameters (R, t) is computed and is used to transform $\mathbf{P}_k^I$ at Epoch I. The "iteration" process will be terminated until the updated mean square error between two point sets is less than a given error threshold τ, or the iteration time reaches the up bound. Figure 4.7 provides a simple explanation of this procedure.

To give an illustrative example, the projection between the closest points of two point sets is sketched in Figure 4.7(a); after ICP registration, point sets at Epoch I are transformed to close point sets at Epoch II, as shown in Figure 4.7(b).

In summary, the process of meta-deformation estimation is shown in Algorithm 4.7 with two main parts: firstly, the point clouds of each pair of corresponding cells are loaded; secondly,

Algorithm 4.6: ***getRigidTransformation***

```
   Input: Two corresponding cells $C_i^I$ and $C_i^{II}$ at Epoch I and Epoch II, respectively
   Output: Rigid transformation for corresponding cell $\{C_{id}^I, C_{id}^{II}, D_i^C\{Rt\}\}$
1  begin
2      /* get the points of correspondent cells */;
3      $\mathbf{P}^I \leftarrow$ pointsInCells($C_{id}^I$);
4      $\mathbf{P}^{II} \leftarrow$ pointsInCells($C_{id}^{II}$);
5      /* give the initial transformation matrix $[R_{initial}, t_{initial}]$ */;
6      $[R, t] \leftarrow$ initialise$[R_{initial}, t_{initial}]$;
7      /* set an error threshold $\tau$ */;
8      $\tau \leftarrow$ errorThreshold;
9      while $d_k - d_{k+1} < \tau$ do
10         /* calculate correspondence between two point sets */;
11         $\mathbf{P}_k^{I_{closest}} \leftarrow$ getClosest $(\mathbf{P}_k^I, \mathbf{P}^{II})$;
12         /* compute the transformation matrix $(R, t)$ based on the correspondence */;
13         $(R, t) \leftarrow$ extractRt $(\mathbf{P}^I, \mathbf{P}_k^{I_{closest}})$;
14         /* transform the points at Epoch I */;
15         $P_{k+1}^I \leftarrow R * (P_k^I) + t$;
16         /* update the mean square error between two point sets */;
17         $d_k \leftarrow \frac{1}{n_p} \sum_{i=1}^{n_p} \|\mathbf{P}_k^{I_{closest}} - \mathbf{P}_{k+1}^I\|$;
18     return Angle difference for correspondent split cells $\{C_{id}^I, C_{id}^{II}, D_i^C\{Rt\}\}$
```



(a) Before registration (b) After registration

Figure 4.7: Rigid transformation detection. Sub-figure 4.7(a) is detection by ICP method; sub-figure 4.7(b) is the point clouds after registration.

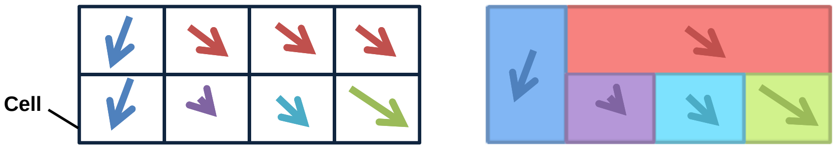
meta-deformation of each pair is computed by using the selected estimation method, including vertical displacement, angle variation, and rigid transformation.

Algorithm 4.7: ***Detect***

```
Input: 3D split cells $\{C^I_{idMin}, \cdots, C^I_{idMax}\}$ at Epoch I and $\{C^{II}_{idMin}, \cdots, C^{II}_{idMax}\}$ at Epoch II,
       the type of meta deformation $metaDetectMethod$
Output: Deformation model for correspondent split cells $\begin{pmatrix} \{C^I_{idMin}, C^{II}_{idMin}, D^C_{idMin}\} \\ \vdots \\ \{C^I_{idMax}, C^{II}_{idMax}, D^C_{idMax}\} \end{pmatrix}$
begin
    /* get the length of correspondent cells */;
    $n \leftarrow$ lengthCells($C^I_{idMin}, \cdots, C^I_{idMax}$);
    forall the ($i \in [1:n]$) do
        /* load two cells separately into $C^I_i$ and $C^{II}_i$ */;
        $C^I_i \leftarrow$ loadCell($C^I_i$);
        $C^{II}_i \leftarrow$ loadCell($C^{II}_i$);
        /* according to the chosen meta deformation such as vertical displacement and angle
        variation, get the difference of this value between $C^I_i$ and $C^{II}_i$ */;
        switch metaDetectMethod do
            case Vertical displacement
                $D^C_i\{V\} \leftarrow$ getVerticalDisplacement($Z(C^I_i), Z(C^{II}_i)$) ;
            case Angle variation
                $D^C_i\{\delta\} \leftarrow$ getAngleDifference($\delta(C^I_i), \delta(C^{II}_i)$) ;
            case Rigid transformation
                $D^C_i\{Rt\} \leftarrow$ getRigidTransformation($Rt(C^I_i), Rt(C^{II}_i)$) ;
    return Meta-deformation model $\begin{pmatrix} \{C^I_{idMin}, C^{II}_{idMin}, D^C_{idMin}\} \\ \vdots \\ \{C^I_{idMax}, C^{II}_{idMax}, D^C_{idMax}\} \end{pmatrix}$
```

4.5 Step III - Merge

In the previous "detect" step, the meta-deformations of cells are computed by comparing the point clouds in the corresponding cells at Epoch I and Epoch II. Now, we need to check whether neighbouring cells have consistent deformation behaviour, and merge the neighbouring cells together if the deformation is similar. For example, in Figure 4.8(a), arrows with difference colour and length are various meta-deformation of cells after "detect". This kind of description can present the deformation for each cell separately. To have a more abstract view on the deformation for this region, the objective of "merge" is to compute *deformation map* for the whole monitored object with sub-deformations for different partial areas. From Figure 4.8(b), we can see that neighbouring cells with similar meta-deformation are grouped together to form a partial area with one type of sub-deformation. This deformation map can present the consistent behaviour inside each partial area. The deformation map is easier to be understand and useful from a global point of view.



(a) Cells with meta-deformation (b) Partial areas with sub-deformation

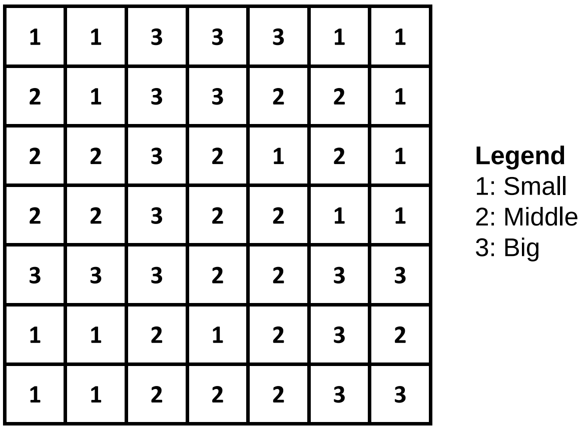
Figure 4.8: Deformation map generation. Sub-figure 4.8(a) is cells with meta-deformation after "detect"; sub-figure 4.8(b) is deformation map consisting partial areas with sub-deformation.

In a deformation map, one partial area of this object should have consistent and even uniform deformation. Based on the characteristics of meta-deformation, clustering methods based on one meta-deformation parameter (e.g., vertical displacement and angle variation) are used to compute the deformation map from the meta-deformation, in terms of the following sub-steps:

1. *Define deformation degree* - We divide the meta-deformation values into a set of non-overlapping degrees. For instance, with the displacement between -1 m to 1 m to be detected, we segment it by intervals of 0.1 m, i.e., $[-1 : 0.1 : 1]$ and then achieve 20 degrees. In addition to such quantitative labels, we can also define qualitative labels, e.g., $\{Small, Middle, Stable, Big, Huge\}$ to the deformation.

2. *Assign deformation degree* - For each cell, we assign predefined deformation degree ("label") according to the raw meta-deformation value computed in the detect step (Section 4.4). Figure 4.9 shows an illustrative example: each cell is assigned a deformation label, from 1 to 3, corresponding to the "small", "middle", and "big" deformation.

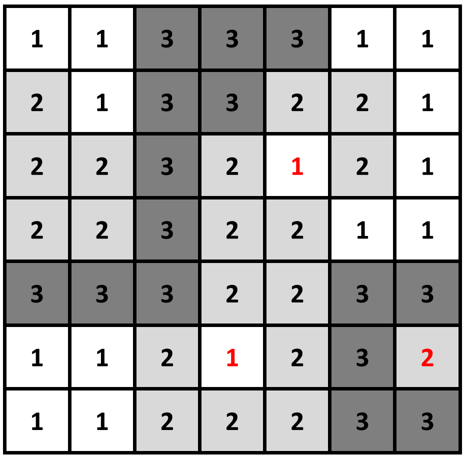
3. *Merge neighbouring ells with the same degree* - All of the meta-deformation of the adjacent cells are compared. If the values are the same, the cells can be grouped together. This procedure is repeated until all cells have been processed and possibly merged with their adjacent cells. After this step, the cells in Figure 4.9 are somehow merged together and generate some partial areas, as shown in Figure 4.10.



1	1	3	3	3	1	1
2	1	3	3	2	2	1
2	2	3	2	1	2	1
2	2	3	2	2	1	1
3	3	3	2	2	3	3
1	1	2	1	2	3	2
1	1	2	2	2	3	3

Legend
1: Small
2: Middle
3: Big

Figure 4.9: Initial deformation map (with assigned labels)



1	1	3	3	3	1	1
2	1	3	3	2	2	1
2	2	3	2	1	2	1
2	2	3	2	2	1	1
3	3	3	2	2	3	3
1	1	2	1	2	3	2
1	1	2	2	2	3	3

Figure 4.10: Intermediate deformation map (after merging)

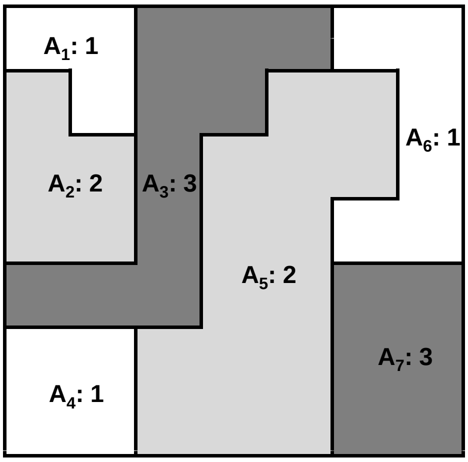


Figure 4.11: Final deformation map (after removing outlier)

4. *Generate the deformation map* - A cell may show particular deformation characteristics significantly different from all of its neighbouring cells. For example, cells with red labels in Figure 4.10 are not the same as their neighbours. The meta-deformation of these cells is considered as the "outlier"; therefore, these cells' meta-deformation will be refined and replaced with their neighbouring cells' status. After this step, we can get the final deformation map shown in Figure 4.11.

Algorithm 4.8 summarises the "merge" step, based on the previously mentioned four sub-steps. The main challenging part in this algorithm is designing a recursive function (see Algorithm 4.9) to expand the area with adjacent cells that have the same deformation degree. In this the *expandArea* function, a candidate area (D^A) is checked with a cell C_i: if C_i has the same deformation degree with D^A, the algorithm recursively checks its 6 neighbouring cells (i.e., *top*, *down*, $left$, *right*, $front$, *back*) and add these cells if the deformation is the same as this area, until there is no more neighbourhoods can be added in.

Algorithm 4.8: ***Merge***

```
Input: Meta deformation model for correspondent split cells $\begin{pmatrix} \{C^{I}_{idMin}, C^{II}_{idMin}, D^{C}_{idMin}\} \\ \vdots \\ \{C^{I}_{idMax}, C^{II}_{idMax}, D^{C}_{idMax}\} \end{pmatrix}$
Output: Deformation map $\mathbf{D} = \{D^A_1, \cdots, D^A_n\}, D^A_i = G(D^C_i, \cdots, D^C_j)$
begin
    $\mathbf{D} \leftarrow \varnothing$ ; // initial an empty deformation map
    /* get the number of cells with meta deformations */;
    $n \leftarrow$ lengthDeformModels $\begin{pmatrix} \{C^{I}_{idMin}, C^{II}_{idMin}, D^{C}_{idMin}\} \\ \vdots \\ \{C^{I}_{idMax}, C^{II}_{idMax}, D^{C}_{idMax}\} \end{pmatrix}$;
    /* assign the label of deformation degree */;
    forall the $(i \in [1:n])$ do
        $degree_i \leftarrow$ degreeLable $(D^C_i)$;
    /* merge the cells into partial areas */;
    forall the $(i \in [1:n])$ do
        if $C_i$ is unprocessed then
            mark $C_i$ as processed;
            $D^A_k \leftarrow \{C_i\}$; // a new $D^A_k$ is created;
            $degree \leftarrow degree_i$; // get the deformation degree;
            $D^A_k \leftarrow$ expandArea $(D^A_k, C_i, degree)$; // expand with adjacent cells
            $\mathbf{D} \leftarrow \mathbf{D} \cup D^A_k$; // a new partial area added to the map;
    return Deformation map $\mathcal{D} = \{D^A_1, \cdots, D^A_n\},\ D^A_i = G(D^C_i, \cdots, D^C_j)$;
```

Algorithm 4.9: ***expandArea***

```
Input: Rough partial areas with given degrees ($D^A$, $C_i$, $degree$)
Output: Expanded partial areas $D^A$
begin
    /* the recursive function of expanding area: to merge neighboring cells that belong to the
    same deformation degree. */
    if $degree! = currDegree(C_i)$ then
        return $D^A$;
    foreach point $C_j$ in $adjectCells$ do
        if $C_j$ is unprocessed then
            mark $C_j$ as processed;
            $neighbor \leftarrow$ {6 adjacent cells}; // top, down, left, right, front, back
            foreach adjacent $C_k$ in $neighbor$ do
                if $degree == currDegree(C_k)$ then
                    $D^A = D^A \cup$ expandArea $(D^A, C_k, degree)$
return Expanded partial areas $D^A$;
```

It is worth noting that the merge step is not mandatary for all applications. We can keep all cells' meta-deformation as the final output, or merge neighbouring cells together according to the similarity of meta-deformation. By using the merge step, the meta-deformation is smoothed with its neighbouring cells, and a more abstract-level deformation description will be created, and some detailed information will be lost. Therefore, this is a tradeoff: "more abstract with the lost of details" vs. "higher accuracy with too much information". Whether we need to do merging depends on the application requirement: focusing on the high-level view or focusing on the details. In addition, we have to admit that there are two main technical challenges in merge: (1) The threshold (or called "smoothing parameter") to determine whether the two neighbouring cells have similar deformation (to group or not). This needs to be tuned according to the application knowledge; in such case, the clustering technique can be applied to determine the configuration of the smoothing parameter. (2) Dealing with outliers – We simply consider the cell with significantly different meta-deformation with its all neighbouring cells as the outlier. In most cases, this is true. However, some applications are more interested in finding abrupt changes; in such case, the outliers might be important message to the application. The merge step is more sensitive and might be not suitable for this scenario. In this thesis, the work is focusing on the "smooth deformation" cases; therefore, the merge is utilised to generate an abstraction of the deformation, e.g., some main parts with semantic descriptions.

4.6 Summary

This chapter presented a novel "cell-based" deformation computation. Based on the hybrid model (including *meta-deformation*, *sub-deformation* and *deformation map*) discussed in Chapter 3, we developed this complementary computation framework to automatically generate interesting deformation parameters and finally formalised the deformation map for the whole object. There are three crucial steps in this framework, i.e., "split", "detect", and "merge".

Through the procedure of "split", we can build uniform cells to divide the point clouds at two epochs, and by using relevant indexing techniques like R-Tree we can quickly access the data points in each cell. Afterwards, meta-deformation parameters such as atomic and complex deformation are detected for the cell by comparing the 3D points in the corresponding cells at the two epochs (by the "detect" step). We particularly focused on atomic meta-deformation like *displacement* and *angle variation*. Furthermore, we designed cell-based ICP to detect complex *rigid-deformation* that includes both rotation and translation. Finally, by the "merge" step, we can group adjacent cells with similar/same meta-deformation together as a partial area. All of the partial areas with their distinctive sub-deformation constitute a deformation map of the whole monitored object. Without given any prior knowledge, our cell-based deformation computation is able to automatically and incrementally generate an informative deformation description of this object from the initial 3D point clouds.

Chapter 5

Evaluation of Cell-based Approach

An ounce of practice is worth
more than tons of preaching.

Mohandas Gandhi (1869-1948)

5.1 Introduction

In this chapter, we present the evaluation of our "cell-based deformation computation" approach. We test the approach using various types of datasets, including both synthetic point clouds and point cloud datasets from real-life natural deformation cases.

Section 5.2 provides our synthetic scenarios that imitate the subsidence of a surface by programming in Matlab, analyses this dataset using our cell-based approach, and compares the results with the ground truth as well as with traditional methods. Section 5.3 presents a real-life case study on the landslides in Flamatt. Section 5.4 analyses the deformation of another real-life case happened in Valais. Section 5.5 provides the comparison between our approach and the available results from BBHN[1]; and additionally compares our cell-based approach with a commercial software (i.e., 3DReshaper). Finally, Section 5.6 summarises and concludes this chapter.

5.2 Evaluation by Synthetic Data

We first evaluate our cell-based approach based on a dataset that generated by a computing program implemented in Matlab. This dataset is designed to imitate a subsidence of a surface, a typical deformation scenario. Three traditional deformation analysis methods (i.e., *displacement*, *angle variation* and *rigid transformation*) are performed on this dataset to detect relevant deformation characteristics. We apply cell-based method to analyse this imitated dataset, apply the three traditional methods as the meta deformation model on cells, and computes the defor-

---

[1] BBHN is a Swiss private surveying company who provided us with the landslide data of Flamatt.

mation map. This is compared to both the results from traditional non-cell based methods and the ground truth from simulation.

5.2.1 Data from synthetic scenario

By coding in Matlab, we build synthetic point cloud datasets in terms of imitating a surface subsidence from Epoch I to Epoch II for deformation monitoring. Figure 5.1 shows the imitated subsidence: at Epoch I, the monitored surface is one single plane that is drawn in the solid line (the angle between the plane and the horizontal surface is $\alpha = 22.9114°$); at Epoch II, the surface slides into two surfaces after subsidence, which are drawn in the dashed lines. Such subsidence could be caused by the movement of surface materials (e.g., debris, mud, rock, etc.). For the two new planes in Epoch II, the top-right surface is steeper ($\alpha_1 = 30°$ to the horizontal surface) and the bottom-left surface is flatter ($\alpha_2 = 15°$ to the horizontal surface).

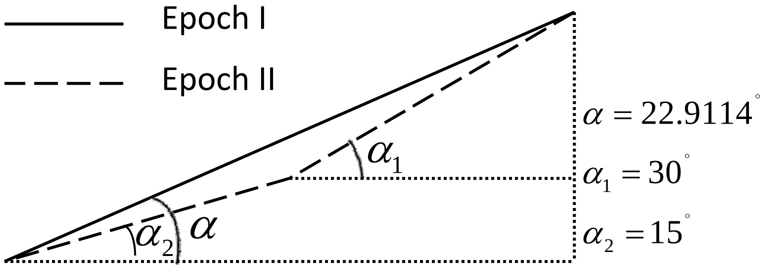
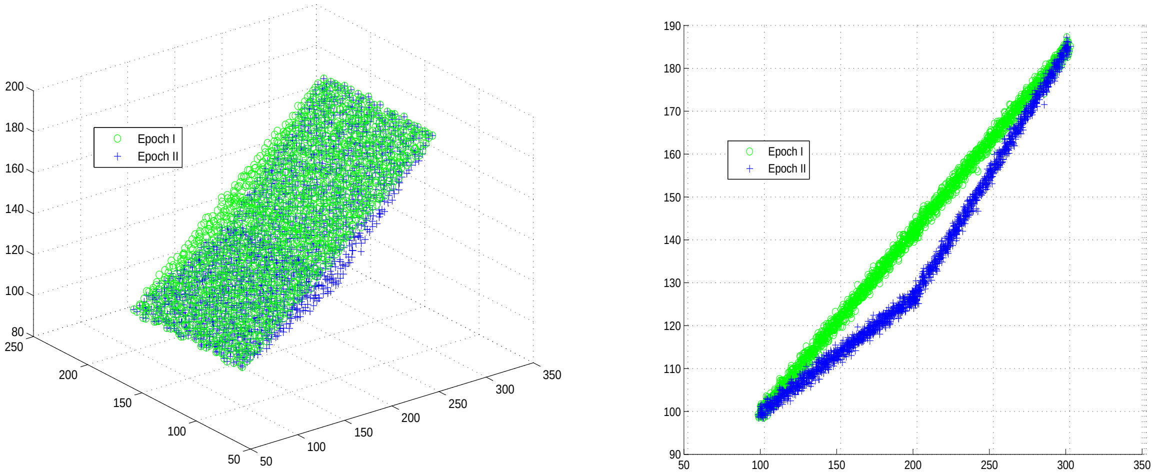


Figure 5.1: Subsidence a synthetic surface

We imitate and sample two sets of point cloud data, i.e., $\mathbf{P}^I$ and $\mathbf{P}^{II}$, corresponding to the two epochs, before and after the subsidence. To simulate the detailed information captured using LiDAR to monitor the surface, we choose reasonably high density to sample 3D points (as about one point per $5\,mm \times 5\,mm$ along X and Y axes) and generate 8.0802×10^4 points sampled on this surface for Epoch I and Epoch II, respectively. In order to imitate real-life data, we consider the random errors of terrestrial LiDAR, and add Gaussian noise to the synthetic data. Five groups of noises with zero mean and different standard deviation $\sigma = (1\,mm, 3\,mm, 5\,mm, 7\,mm, 10\,mm)$ are generated and added to the synthetic point clouds. Figure 5.2 shows two point cloud datasets of the imitated surface at two epochs containing the Gaussian noises with $\sigma = 3\,mm$. In the experiments, we additionally analyse the effects of using different noise levels on the performance of estimating deformation parameters.

This subsidence situation can be modelled using three different deformation methods, i.e., *displacement*, *angle variation*, and *rigid transformation*. Displacement of the surface is the change of one point on the surface from Epoch I to Epoch II along a pre-defined direction, e.g., X, Y, or Z axis. Especially the vertical displacement along Z axis is frequently used by civil engineers and geologists [MMB$^+$08, TOD$^+$08]. Angle variation is the change of the surface normals between two epochs; therefore it can simply illustrate the "direction" deformation of the surface. Rigid transformation is rotation and translation of the monitored surface from Epoch I to Epoch II and can explicitly formalise the movement of this surface. For monitoring the deformation of this synthetic dataset, we apply the three models as the meta-deformation in our



(a) Point clouds in 3D

(b) 2D projection

Figure 5.2: Two point cloud datasets of the simulated subsidence at Epoch I and Epoch II (sampling density: 5 mm * 5 mm, standard deviation $\sigma = 3\ mm$)

cell-based approach. We also compare the “cell” approach with traditional “non-cell” approach.

1. *Detection of displacement* - We estimate displacement of each cell using several detection methods such as *min*, *max*, *mean* and *median*, and utilise two metrics like *false positive ratio* and *false negative ratio* to analyse the error of estimation results. In addition, we perform sensitivity analysis to evaluate the influence brought by different *splitting cell sizes* and varying *noises of point clouds*.

2. *Detection of angle variation* - We detect angle variation of this synthetic scenario using the “plane fitting” regression method and compare our results with the ground truth. Similar to the sensitivity analysis in displacement, we eventually analyse the effects caused by different cell sizes and noises.

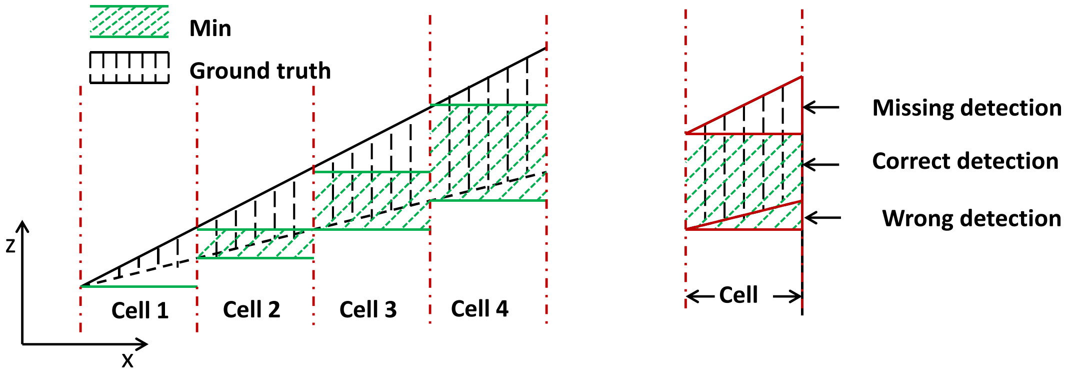
3. *Detection of rigid transformation* - We present cell-based ICP method to extract *rotation* and *translation* of each individual cell. Afterwards, we describe the comparison of estimated results by traditional ICP method with the results by cell-based ICP approach.

5.2.2 Detection of displacement

Displacement is a generic and important parameter for detecting deformation. In traditional methods, unstable points are extracted from the geodetic control network [Cas88, HR01]. Displacement of these discrete unstable points describe the changes of this region. To give a concrete example that uses a small set of monitoring points for deformation detection, Figure 4.1(a) shows n discrete points on the monitored object; then we get two n observations at Epoch I ($\mathbf{L}^I = \{L_1^I, L_2^I, \cdots, L_n^I\}$) and Epoch II ($\mathbf{L}^{II} = \{L_1^{II}, L_2^{II}, \cdots, L_n^{II}\}$), respectively. The change between two corresponding points (i.e., L_i^I and L_i^{II}), which can be calculated in terms of the distance ($dist(L_i^I, L_i^{II})$), approximates the displacement of this region from Epoch I to Epoch II. This is drawn as the dashed lines in Figure 4.1(a). Thanks to high-density point cloud data capture by the terrestrial LiDAR technology, we are able to develop a cell-based approach to estimate displacement of the whole monitored object from point clouds ($m1 >> n$ points $\mathbf{P}^I = \{P_1^I, P_2^I, \cdots, P_{m1}^I\}$ at Epoch I, and $m2 >> n$ points $\mathbf{P}^I = \{P_1^I, P_2^I, \cdots, P_{m2}^I\}$ at Epoch II), instead of some sparse points ($\mathbf{L}$) on this object.

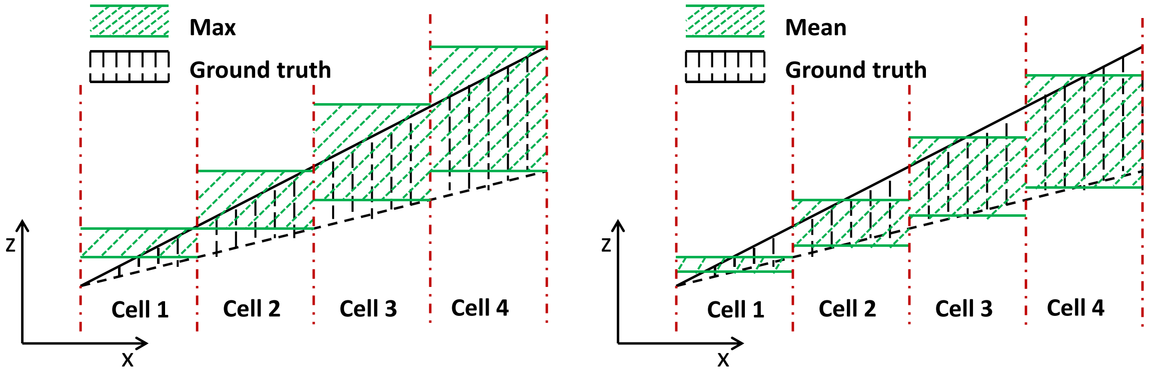
Estimation Methods of Displacement - We detect the displacement using the point clouds in the corresponding cells. For each pair of corresponding cells, they contain point clouds at Epoch I and II, respectively. We select one representative point in this cell for each epoch. Afterwards, we compare this pair of points and calculate the distance/difference to approximate the displacement of this cell.

Several estimation methods such as *min*, *max*, *mean* and *median* are applied to represent the points inside a cell. Figure 5.3 indicates the displacement of our imitated subsidence using the representative points generated by these methods. The areas with green-dashed-slash (/) texture are the displacement detected by our methods; while the areas with black-vertical (|) texture are the ground truth of displacement. We can observe that our results are not exactly the same as ground truth from Figure 5.3(a), 5.3(c), and 5.3(d). Therefore, we need to quantify the errors of deformation detection. Figure 5.3(b) shows errors of one cell estimated by the *min*



(a) Displacement detected by min

(b) Error for one cell by "min"



(c) Displacement detected by max

(d) Displacement detected by mean

Figure 5.3: Displacement of the synthetic subsidence between Epoch I and Epoch II via different detection methods

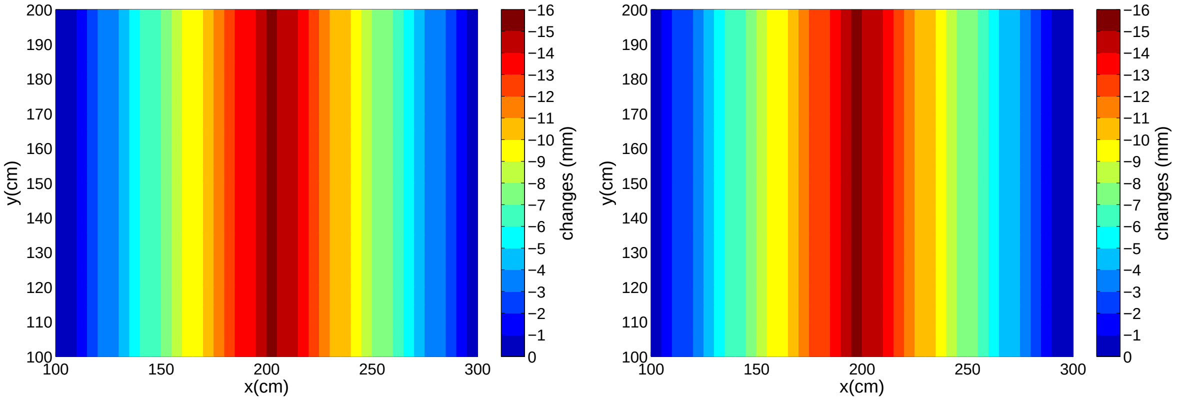
method. The detection error comprises two parts, i.e., *missing detection* and *wrong detection*. Missing detection is the real displacement but not extracted by the estimation methods; wrong detection is the displacement extracted by our methods, but in fact, such displacement does not exist. Therefore, we use two metrics (*falsePositiveRatio* and *falseNegativeRatio*) to quantify the errors of these estimation methods as:

$$falsePositiveRatio = \frac{\sum_i (Detection_i \bigcap \neg GroundTruth_i)}{\sum_i Detection_i} \tag{5.1}$$

$$falseNegativeRatio = \frac{\sum_i (\neg Detection_i \bigcap GroundTruth_i)}{\sum_i GroundTruth_i} \tag{5.2}$$

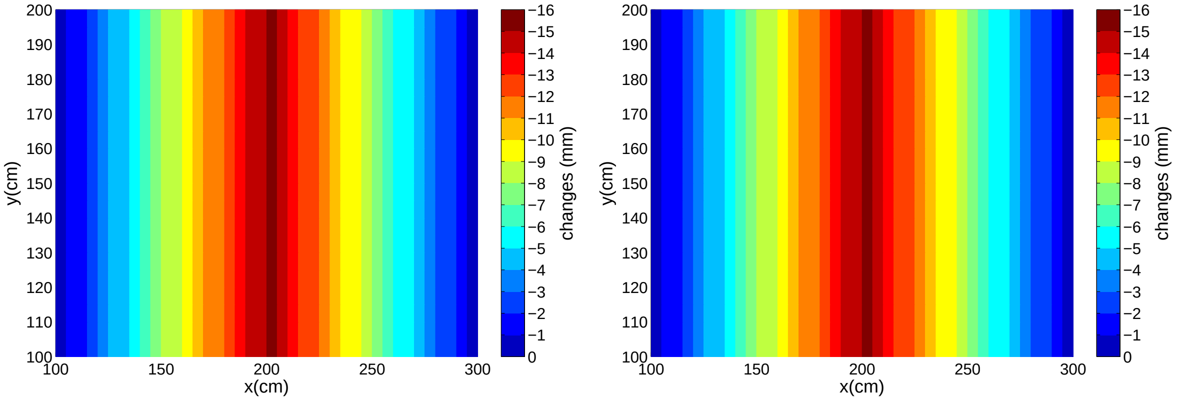
where *falsePositiveRatio* is the ratio of wrong detection (i.e., $\sum_i (Detection_i \bigcap \neg GroundTruth_i)$) to the whole detected displacement; and *falseNegativeRatio* is the ratio of missing detection to the ground truth of displacement. *falsePositiveRatio* describes how much detection is wrong and *falseNegativeRatio* indicates how much ground truth has not been detected.

We apply these methods to detect the displacement of the imitated subsidence; and analyse the results in terms of two perspectives, i.e, *deformation map* and *distribution of the displacement*. In addition, we evaluate the detection results compared to the ground truth.



(a) Min

(b) Max



(c) Mean

(d) Median

Figure 5.4: Displacement of the imitated subsidence between Epoch I and Epoch II via different detection methods (cell size: 5 *cm*)

Deformation Map - For the cell size of 5 *cm*, Figure 5.4 shows displacement of this subsidence (i.e., the deformation map) detected by the four estimation methods, i.e., *min*, *max*, *mean* and *median*, respectively. From the deformation maps of Figures 5.4(a), 5.4(b), 5.4(c) and 5.4(d), we observe that the results by different estimations methods such as *min*, *max*, *mean* and *median* are similar in general. At the top of the slope (i.e., the right side of the deformation map), the displacement is the minimum; from the top of the slope to the down side (i.e., the right to the left in the figure), the displacement first increases and reaches the maximum at the middle of the slope/map, then decreases until reach the bottom of the slope (i.e., the left side of the deformation map). Such results are consistent with the imitated displacement that we set up in the simulation program. The range of whole displacement is from -16 *mm* to 0 *mm*, which is same as the ground truth. To further check the detailed difference between these four deformation maps in Figure 5.4, we analyse the distribution of displacement.

Distribution of Displacement - Figure 5.5 shows the distribution of the displacement detected by several estimation methods such as *min, max, mean* and *median.* The distributions are calculated using histograms at the same intervals. From Figure 5.5, we observe the range of the displacement (from - 16 *mm* to 0 *mm*) is coincident with the graphic description (see Figure 5.4). Moreover, Figure 5.5(c) and Figure 5.5(d) are the same. This is because *mean* and *median* has the same value for our synthetic data without adding noise. The detailed definitions of *mean* and *median* are as follows,

$$\begin{aligned} &Z_{mean}(C) = mean(z_{p_m}, \cdots, z_{p_n}) \\ &D_{id}^{C} = Z_{mean}(C^{II}) - Z_{mean}(C^{I}) \qquad (5.3) \end{aligned}$$

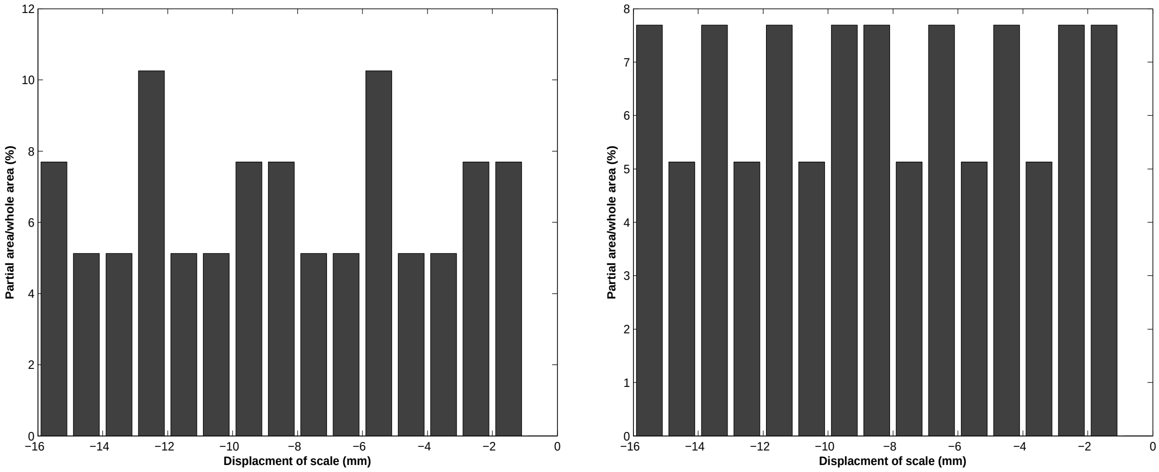
$$\begin{aligned} &Z_{mean}(C) = median(z_{p_m}, \cdots, z_{p_n}) \\ &D_{id}^{C} = Z_{median}(C^{II}) - Z_{median}(C^{I}) \qquad (5.4) \end{aligned}$$

where $\{P_m, \cdots, P_n\}$ are the points inside this cell.

Figure 5.5(a) and Figure 5.5(b) are more different compared with the other two distributions, because the *min* method chooses the minimum value to represent this cell, while the *max* method selects the maximum value to approximate this cell. Therefore, the results detected by the *min* method are smaller than the results detected by the *max* method for each interval.

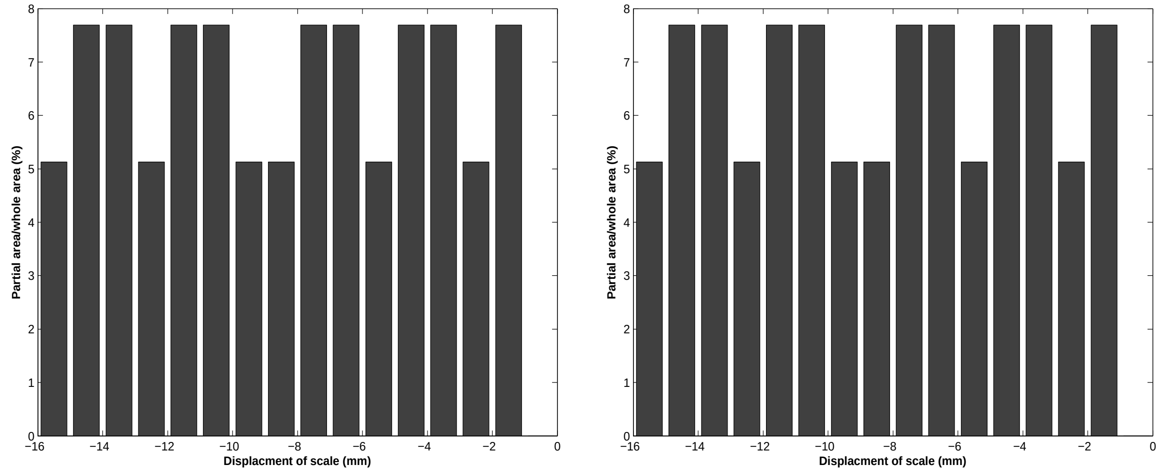
By now, we have analysed the displacement detection results using both (1) from the graphic viewpoint by deformation map and (2) from the perspective of distribution using histograms of displacement. Furthermore, we check the error by the two metrics (*false positive ratio* and *false negative ratio*) we defined previously. Table 5.1 shows the detected results by *min*, *max* and *mean* methods; we ignore *median* here as it has the same results with *mean.* We can observe that these three methods are quite similar in terms of the detection of displacement. As this slope has slid into two parts, we observe that the *mean* method has better performance compared to other methods.

Sensitivity Analysis - To check the robustness of this cell-based deformation analysis using displacement, we perform sensitivity analysis and investigates the influence of cell sizes and



(a) Min

(b) Max



(c) Mean

(d) Median

Figure 5.5: Distribution of displacement detected by min, max, mean and median with the same cell size as 5 *cm*

Methods	*False positive ratio* (%)	*False negative ratio* (%)
Min	12.42	8.3
Max	12.42	8.29
Mean	6.22	1.77

Table 5.1: Error analysis of deformation estimation methods

noises on the final displacement detection results: (1) error analysis of using varying cell sizes, and (2) error analysis of adding different noises.

Error Analysis of Different Cell Sizes. Cell size is an important parameter that influences the detection results. Smaller cell size increases the time of computation, but the detection accuracy is higher; and vice versa for using bigger cell size. Therefore, it is an important issue to find the most suitable cell size that can optimally balance between computing efficiency and detection accuracy for our cell-based approach.

We extract the displacement with the same estimation method (*min*), but using different cell sizes (i.e., 5 *cm*, 10 *cm*, 15 *cm*, 20 *cm*). The point cloud of this slope contains 80 802 points at each epoch and has Gaussian noises with standard deviation as 3 *mm*. Obviously, when the cell size increased from 5 *cm* to 20 *cm*, the number of cells for this dataset decreased from 861 to 66; and the average point number in one cell increased from 94 to 1243 (see Figure 5.2). Additionally, Table 5.2 shows the *false positive ratio* and *false negative ratio* achieved from different cell sizes. From Table 5.2, both *false positive ratio* and *false negative ratio* increase with the decline of cell size. As the estimation method is based on the points inside a cell, smaller cell can approximate the details of a surface more explicitly.

Cell sizes (*cm*)	*Cell number*	*Average point number in one cell*	*False positive ratio* (%)	*False negative ratio* (%)
5	861	94	12.29	7.91
10	231	351	25.95	18.55
15	98	824	39.68	36.78
20	66	1243	53.25	43.91

Table 5.2: Error analysis of deformation detection with different cell sizes

Error Analysis of Varying Noises. We also analyse the sensitivity of adding different levels of noises during the generation of the synthetic data. We test five noise levels: all have zero means, but different standard deviation (i.e., 1 *mm*, 3 *mm*, 5 *mm*, 7 *mm*, 10 *mm*). The same point cloud dataset of this slope containing 80 802 points at each epoch was split into cells with the size as 5 *cm*. For each noise level, the cell number is 861 and each cell has about 94 points inside. From Table 5.3, we observe the approach is quite robust for all noisy data in general, even for point clouds containing noises with standard deviation as 10 *mm*. Of course, the two error metrics rise with the increase of the noise level.

Noises (*mm*)	*False positive ratio* (%)	*False negative ratio* (%)
1	12.32	7.94
3	12.39	8.01
5	12.41	8.03
7	12.45	8.08
10	12.54	8.17

Table 5.3: Error analysis of deformation detection from datasets with different noises

5.2.3 Detection of angle variation

In addition to displacement, angle variation is another typical way to describe the change of a slope caused by subsidence or landslides. Angle variation represents the change of the normal of a surface. This can be modelled as meta-deformation for cells in deformation detection via 3D point clouds.

Estimation Method of Angle Variation - In Figure 5.6, a slope (solid line) at Epoch I changes to another slope (dashed line) at Epoch II. Taking $Cell3$ as an example, δ_1 and δ_2 are the two angles of this slope inside the cell at Epoch I and II, respectively. Difference between δ_1 and δ_2 describes the direction change of the plane in cell three and is entitled angle variation. To get angle variation in all cells, we fit a plane to the points in each cell, respectively (see bold lines in Figure 5.6), and then analyse the difference of these two planes via their normals.

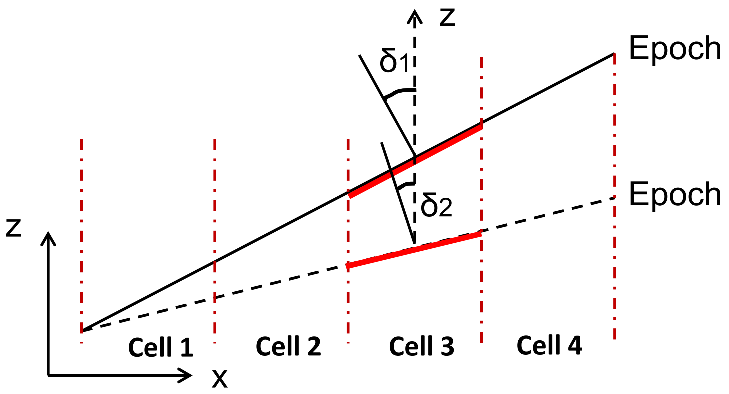


Figure 5.6: Angle variation of the slope between two Epochs

Split and Detection – In order to compute the deformation parameters, we split the space containing the monitored surfaces into 1.4636×10^4 uniform cells according to user-defined cell size in terms of $length, width, height = \langle 5\ cm, 5\ cm, 5\ cm\rangle$. For each cell including two point cloud datasets at two epochs, we calculate the meta-deformation of each cell $D^C_{id}(\delta) : \widehat{\delta}$. More details of this formula/notation can be referred to Chapter 4 and Equation 4.6.

To evaluate the estimated meta-deformation, we compare $\widehat{\delta}$ to the ground-truth value δ by $\Delta\delta = \widehat{\delta} - \delta$ and take $\overline{\Delta\delta}$ (the average of $\Delta\delta$ for all cells) as the evaluation criteria. For the synthetic dataset, the average of the errors $\overline{\Delta\delta}$ between $\widehat{\delta}$ and δ is $3.69° \times 10^{-4}$ (i.e., 1.33 seconds of arc) and the standard deviation is $6.69° \times 10^{-4}$ (i.e., 2.41 seconds of arc) calculated by Equation 5.5. The selected estimated meta-deformation are shown in Table 5.4. In this table, the results of $Cell^{(1)}_{ID} = \{822\ \&\ 12909\}$ are quite bad, because these cells contain so few points to estimate meta-deformation, e.g., cell 822 has 4 points at Epoch I and 10 points at epoch II.

$$\overline{\Delta\delta} = \frac{\sum_{i=1}^{N} |\widehat{\delta}_i - \delta_i|}{N}$$

$$\sigma^2 = \frac{\sum_{i=1}^{N} (\widehat{\delta}_i - \delta_i)^2}{N-1} \tag{5.5}$$

where N is the number of the cells containing the corresponding point cloud to fit the plane at Epoch I and Epoch II.

To avoid such completely wrong estimations, these cells contain so few points need to be rejected. We derived the rejection bound from the histogram of the number of the points in the cells, shown by Figure 5.7. Nearly 10% of cells with few points were rejected to estimate the meta-deformation for our imitated data.

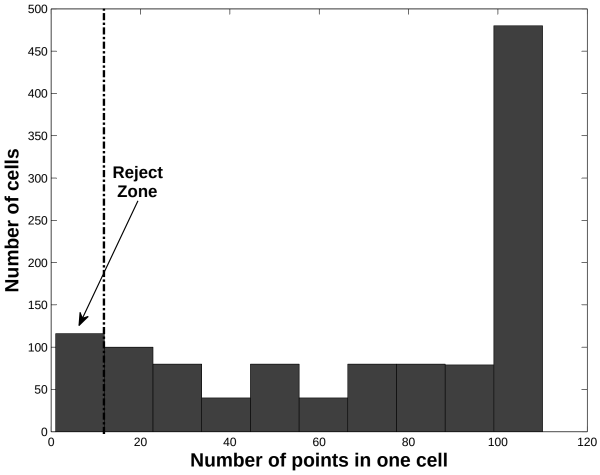


Figure 5.7: Histogram of the number of the points in each cell at Epoch I (sampling density: 5 mm * 5 mm)

Analysis of the Noise Level and the Cell Size – Figure 5.8 shows the influence of important parameters: (1) the noise levels (σ) when generating the synthetic data, and (2) the cell size (w) when building the cells for meta-deformation estimation. When the standard deviation of the noises σ in the point cloud dataset increases, the error of the estimation also increases. For detecting meta-deformation, the estimation result becomes better with the cell size increases; the reason is that big cells can have more data points to fit the plane and have better quality of fitting for this synthetic dataset.

	Epoch I				*Epoch II*				$\widehat{\delta}$	δ	$\Delta\delta$
$Cell^1_{ID}$	N^I_p	n^I_x	n^I_y	n^I_z	N^{II}_p	n^{II}_x	n^{II}_y	n^{II}_z	($\circ$)	($\circ$)	($\circ$)
1	100	-0.3893	0.0003	0.9211	100	-0.2588	0.0003	0.9659	7.9091	7.9114	-0.0023
2	40	-0.3903	-0.0003	0.9207	100	-0.2587	-0.0001	0.9659	7.9754	7.9114	0.0640
41	90	-0.3893	-0.0002	0.9211	90	-0.2588	-0.0003	0.9659	7.9116	7.9114	-0.0002
822	4	0.0000	-1.0000	0.0000	10	0.0000	-1.0000	0.0000	0.0000	7.9114	reject
1643	100	-0.3895	0.0001	0.9210	30	0.2579	0.0001	-0.9662	7.9740	7.9114	0.0626
1644	80	0.3892	0.0002	-0.9212	100	-0.2588	-0.0001	0.9659	7.9055	7.9114	-0.0059
11226	20	-0.3914	0.0006	0.9202	70	0.5006	-0.0001	-0.8657	7.0025	7.0886	-0.0861
11267	20	-0.3910	0.0002	0.9204	70	0.4998	-0.0002	-0.8661	6.9712	7.0886	-0.1174
12909	5	0.0000	-1.0000	0.0000	10	0.0000	-1.0000	0.0000	0.0000	7.0886	reject
14594	100	-0.3893	-0.0002	0.9211	20	-0.5000	-0.0003	0.8660	7.0885	7.0886	-0.0001
14595	100	-0.3907	-0.0006	0.9205	20	-0.5008	-0.0007	0.8656	7.0528	7.0886	-0.0358

Table 5.4: Selected results of the estimated parameters, grey parts are rejected estimation examples. The average of the errors $\overline{\Delta\delta}$ between $\widehat{\delta}$ and δ is $1.328''$.

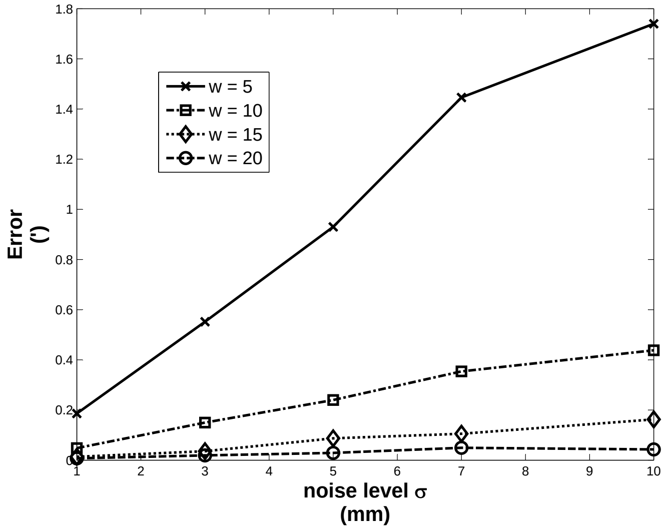


Figure 5.8: Estimation error with regard to noise level (σ) and cell size (w)

5.2.4 Detection of rigid transformation

In previous subsections, we applied "displacement" and "angle variation" to compute meta-deformation, representing "translation" and "rotation", respectively. Both belong to rigid transformation, which is a transformation from a Euclidean space to itself that preserves distances between every pair of points (i.e., there is only absolute change for the whole object/cell no relative change inside the object/cell). Rigid transformations include rotations, translations, reflections, or their combination. Like detecting rotation and translation of the whole object by analysing discrete points sampled sparsely on this object [Che83, EKB10], the combined rigid transformation is also largely used in deformation detection [TGZG07, MC08]. The high density point clouds can be used to identify the combined rigid transformation of the object. Now, we study cell-based detection of rigid transformation, not only for the whole monitored object, but also for partial areas inside. In this section, we present the cell-based ICP (Iterative Closest Point) method to detect rigid transformation, and then compare our results with the results directly computed using classical ICP method on the whole synthetic object without using cells.

Estimation Method of Rigid Transformation - Similar to cell-based displacement and angle variation detection, we apply the Iterative Closest Point (ICP) method as the meta-deformation in our cell-based deformation approach to extract rigid transformation of the monitored object. To better understand the cell-based ICP method, we firstly introduce the classical ICP method.

Classical ICP. ICP method is a well-known algorithm developed for solving registration of 3D shapes, especially in the field of Computer Vision [BM92, Zha94]. Given two point clouds $\mathbf{P}^I$ and $\mathbf{P}^{II}$, classical ICP method attempts to minimise the mean squared distance between the points in $\mathbf{P}^{II}$ and the transformed points from $\mathbf{P}^I$. This can be formulated as an optimization problem. The objective function is written as follows:

$$f(R,T) = \frac{1}{N_P}\sum_{i=1}^{N_P}(P_i^{II} - R(P_i^I) - T)^2 \tag{5.6}$$

where R and T are two fundemental components of rigid transformation matrix, corresponding to rotation and translation, respectively; R and T are performed on $\mathbf{P}^I$. N_P is the number of the corresponding points in these two datasets.

Beginning with the initial transformation matrix, this algorithm seeks to search each corresponding closest point P_i^{II} in $\mathbf{P}^{II}$ for P_j^I, expressed by

$$d(P_j^I, P_i^{II}) = \sqrt{(x_i^{II} - x_i^I)^2 + (y_i^{II} - y_i^I)^2 + (z_i^{II} - z_i^I)^2} \tag{5.7}$$

$$d(P_j^I, \mathbf{P}^{II}) = \min_{i \in \{1, \cdots, n\}} d(P_j^{II}, P_i^I) \tag{5.8}$$

where $d(P_j^I, P_i^{II})$ is the distance between two points P_j^I and P_i^{II}; $d(P_j^I, \mathbf{P}^{II})$ is the minimum distance between the point P_j^I and all points in the dataset $\mathbf{P}^{II}$.

The ICP algorithm has four steps: (1) build the matching based on Equation 5.7 for finding the closest point in P_j^{II} for each point in P_i^I; (2) based on the matching, compute the rigid transformation parameters, i.e., R and T; (3) apply the transformation to compute the mean distance between two point sets; (4) if the mean distance is below a given threshold (δ_{ICP}), then the algorithm stops and we reach the final R and T; otherwise, iteratively rebuild the matching and compute a new transformation. This procedure is performed iteratively until reach the convergence threshold or the maximum iteration number.

Cell-based ICP. From technical point of view, cell-based ICP is quite similar to ICP, by using the iterative procedure to build the matching and the transformation parameters. Assuming that two point clouds are sampled on the monitored object at two epochs I and II, i.e., $\mathbf{P}^I = \{P_1^I, \cdots, P_m^I\}$ and $\mathbf{P}^{II} = \{P_1^{II}, \cdots, P_n^{II}\}$ both in $\Omega \subseteq \mathbb{R}^3$, where each Point $P_i^I = [x_i^I, y_i^I, z_i^I]^\top$ and $P_i^{II} = [x_i^{II}, y_i^{II}, z_i^{II}]^\top$. We divide the data into some cells, and focus on ICP rigid transformation independently for each cell. As shown in Figure 5.9, we apply the cell-based approach on the same imitated subsidence scenario in Section 5.2.1. This experiment is to test the possibility of application of cell-based approach on subsidence detection. Firstly, the imitated surface is split into two cells. Secondly, we use ICP method to detect the transformation parameters of cell 1 and cell 2 independently. To compare the performance between non-cell ICP and cell-based ICP, we additionally apply ICP method for the detection of transformation of the whole object.

In order to compare the classical ICP method and cell-based ICP approach, the results via these two methods are evaluated by two metrics *mean squared distance*, expressed as Equation 5.9 and *mean distance*, expressed as Equation 5.10.

$$dis_{square} = \sqrt{\frac{1}{N_P} \sum_{i=1}^{N_P} (P_i^{II} - R(P_i^I) - T)^2} \tag{5.9}$$

$$dis_{mean} = \frac{1}{N_P} \sum_{i=1}^{N_P} |P_i^{II} - R(P_i^I) - T| \tag{5.10}$$

where dis_{square} is the mean squared distance between P_i^{II} and transformed P_i^I. When the alignment of P_i^{II} and transformed P_i^I is better, dis_{square} is smaller.

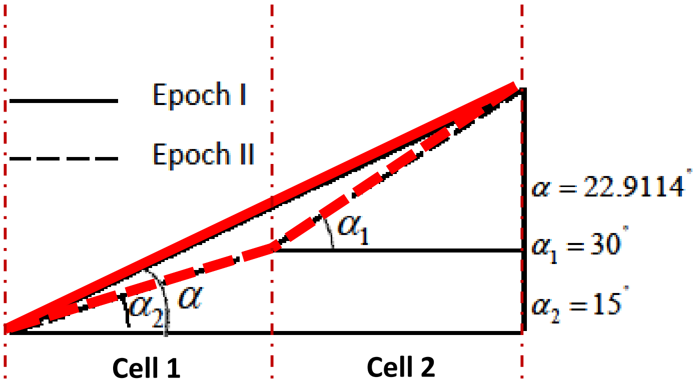


Figure 5.9: Cell-based ICP approach

Estimation Results - Figure 5.10 shows the performance comparing the classical ICP and our cell-based ICP on the simulated subsidence, based on the metric of mean squared distance. The solid line is the mean square distance of traditional ICP at each iteration; while the dotted line and dashed line are the mean square distance of cell-based ICP for *cell*1 and *cell*2, respectively. For the whole object, traditional ICP method converges at 5.5 *mm*; while cell-based ICP method converges around 2 *mm* for both cells. Therefore, we can claim that the cell-based ICP has better results in terms of estimating rigid transformation. In addition, Figure 5.11 shows the second metric, i.e., the mean distance; and we also observe that cell-based ICP has much less errors compared to the non-cell one, when the point cloud is split into suitable cells. We have to admit a coincidence about the deformation slope in Figure 5.9, which is likely aligned with the cell boundary when using two cells. Nevertheless, if without such coincidence, cell-based ICP is still able to provide better performance compared to the non-cell one. In general, the more cells the experiment uses, the better results it can achieve.

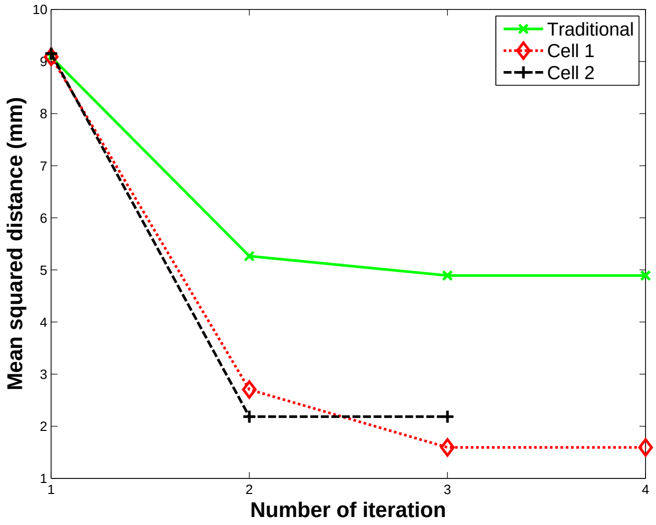


Figure 5.10: Mean squared distance via traditional and cell-based ICP

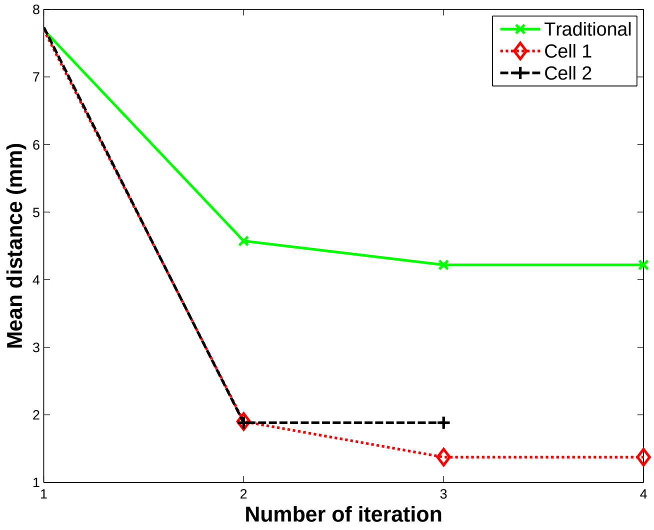


Figure 5.11: Mean distance via traditional and cell-based ICP

5.3 Case Study I - Landslides in Flamatt

The synthetic data only simulate the deformation scenarios. To further evaluate our cell-based approach, we apply this approach on point cloud datasets from real-life scenarios. We need to compare our results with the results from traditional methods/softwares or provided by the third party. We obtain two real-life datasets with typical deformation. One is about a landslide on a slope along a Swiss major railway line, and the other is erosion on a concrete slope in Switzerland as well. More explicit descriptions of these two cases are offered in Section 5.3.1 and Section 5.4.1, respectively.

5.3.1 Data description

We test our approach for analysing a landslide in Switzerland, which has affected a major railway line. The size of this unstable slope is approximately 200 m × 25 m (see Figure 5.12). The landslide slope is the region inside red lines in the figure. Two dense point cloud datasets are collected by TLS separately at August 14, 2007 and October 5, 2007, with 20,280,793 and 10,741,995 points, respectively. The first measurement was done before the landslide, whilst the second measurement was made just after the landslide happened.

5.3.2 Results by cell-based approach

The main results of our approach are presented in terms of the following two main aspects: (1) the deformation map computed by cell-based deformation analysis, and (2) the sensitivity analysis of cell size when testing our cell-based approach.

Deformation Map – The deformation map of this landslide (computed by using the vertical displacement as the meta-deformation) is sketched in Figure 5.13. This map is produced by the “min” method to extract the meta-deformation model (i.e., vertical displacement) of each cell. This deformation map illustrates: (a) most subsidence values of this region are centred around

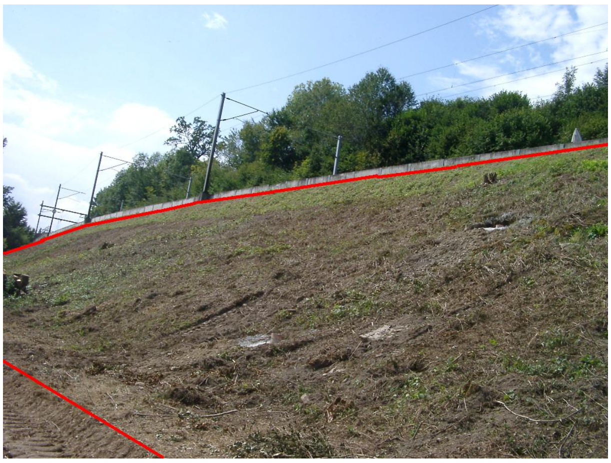


Figure 5.12: Monitored slope at Flamatt

0 m; (b) the lower edge of this region has slightly increased: about 0.4 m; (c) two spots of this region have larger increment: more than 1 m. While (a) and (b) have a predominantly natural cause, (c) is due to earthworks realised to allow construction vehicles to access the site.

The corresponding deformation distribution is shown in Figure 5.14, which is the histogram of vertical displacement for all cells. We observe that (1) the histogram is centred around zero; (2) the sinking (i.e., negative values) is between 0.5 and 0 – many cells have small sinking (close to zero) but a few go to 0.5; and (3) the floating (i.e., positive values) is almost uniformly distributed between 0 and 1m. In summary, the most different deformation are in subsidence, which is quite consistent with this landslide scenario.

Sensitivity Analysis of Cell Size – A suitable cell size is the guarantee to obtain the deformation description of the whole region efficiently and accurately. If the cell size is too small, it will cost more computation time; if the cell size is too big, it will lose the accuracy of the deformation. Therefore, how to optimize the cell size is a crucial step in 3D cell splitting. According

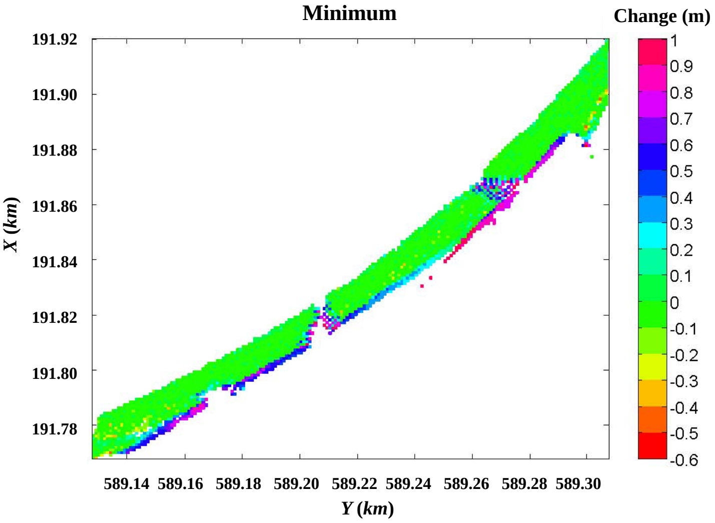


Figure 5.13: Deformation map of the whole slope

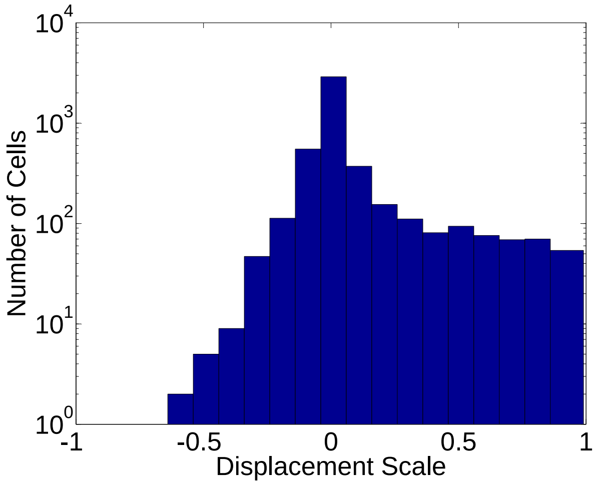


Figure 5.14: Deformation distribution (histogram) by “min”

to the value of displacement, 1 m, 0.5 m, 0.3 m and 0.2 m are chosen as the hierarchy of cell size. Figure 5.15 shows the detecting stability of different cell size. In a generic view, the results are quite similar amongst different cell sizes; however, small cell size gives more stable ability to detect displacement in details. For example, the cell size of 0.2 m provides more results on small changes in [-0.2 m $\sim$ -0.1 m] and [-0.1 m $\sim$ 0 m], while quite less in [0 m $\sim$ 0.1 m] and [0.1 m $\sim$ 0.2 m] when using cell sizes of 1 m or 0.5 m.

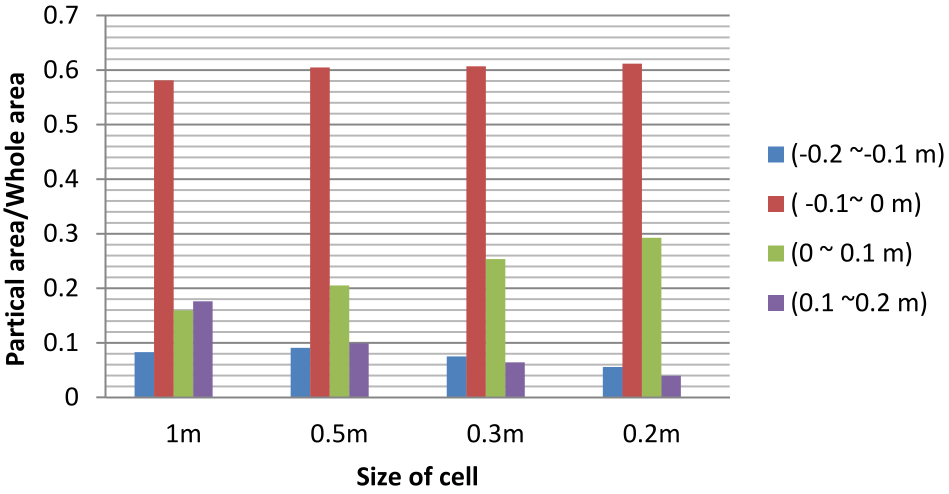
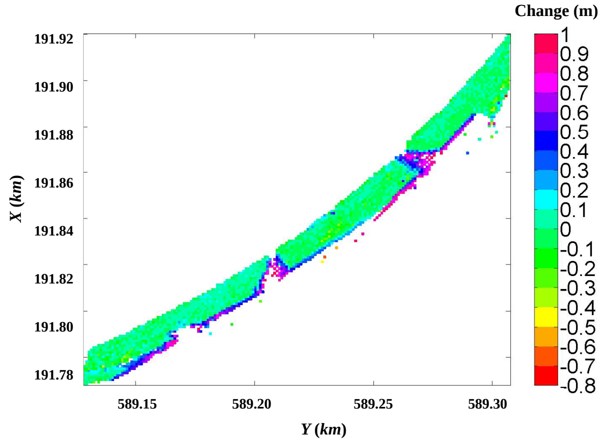
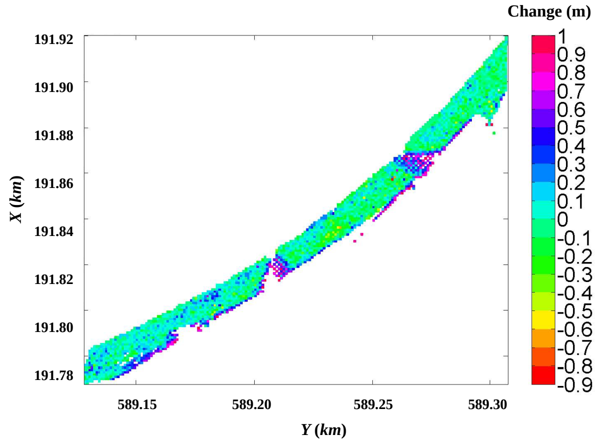


Figure 5.15: Deformation distribution with regard to different cell size

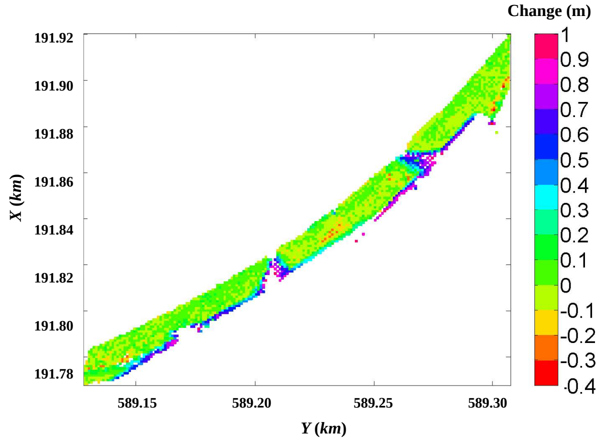
Deformation Model Estimation – Similar to the displacement detection using various methods in analysing the synthetic data in Section 5.2.2, we can also apply other methods to estimate meta-deformation of the point cloud in each cell between two epochs, besides the “min” method. Figure 5.16 shows the vertical displacement of this unstable slope detected by the “median”, “max” and “mean” methods. Compared to the deformation map by the “min” method in Figure 5.13, we observe noises from the deformation maps by these three methods in Figure5.16(a), 5.16(b) and 5.16(c). In particular, the “max” and “mean” methods (Figure 5.16(b) and 5.16(c))



(a) median

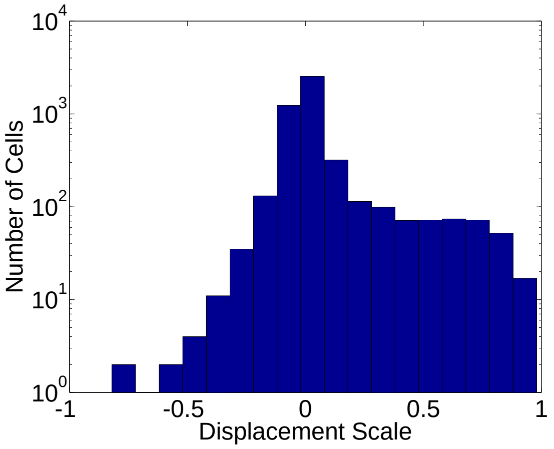


(b) max

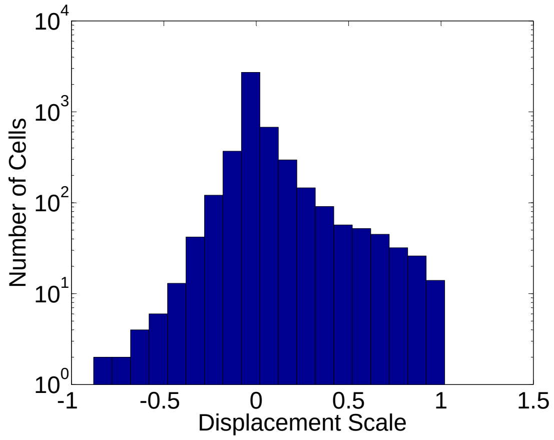


(c) mean

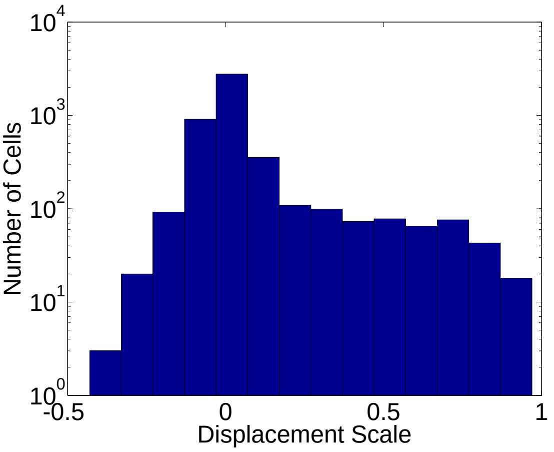
Figure 5.16: Displacement extracted by different deformation estimation



(a) median



(b) max



(c) mean

Figure 5.17: Deformation distribution (histogram) of vertical displacement result

show more noises that are scattered on this slope.

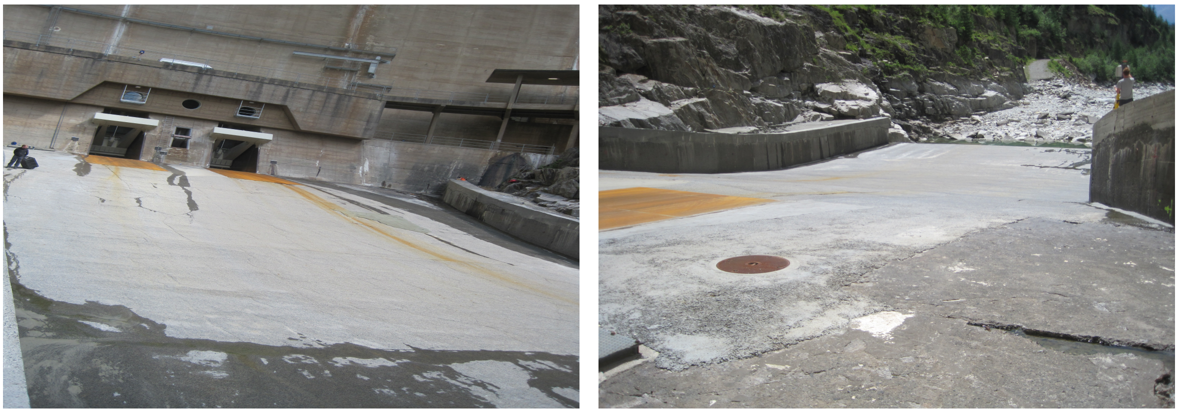
Similar to presenting the detailed displacement of the "min" method in Figure 5.14, the displacement distributions by these methods are given in Figure 5.17. This figure shows that the vertical displacement achieved by the "median", "max" and "mean" methods have similar histograms with the "min" method in general, where most vertical displacement are centred at 0 *m*. However, the detailed distribution in Figure 5.17(a) looks more reasonable compared to the distribution in Figure 5.14, which is consistent with the deformation map plots: the "median" method has closer results to the "min" method.

5.4 Case Study II - Erosion on a Concrete Slope

In the previous section, we applied our cell-based approach on real-life case study I, a slope landslides along a railway line in Switzerland. We detected the displacement of this slope is from -0.6 *m* up to more than 1 *m*. In this section, we further test our approach in another real-life deformation case in Switzerland, i.e., erosion on a concrete slope with slight changes.

5.4.1 Data description

The point cloud datasets were captured by Haute Ecole d'Ingenierie et de Gestion du Canton de Vaud (HEIG-Vd)[2] on a concrete slope. This concrete slope connects a big dam with a natural riverbed, as shown in Figure 5.18. The monitored slop is at the bottom of a dam (see Figure 5.18(a)); and there is a riverbed at the end of this slope, as shown in Figure 5.18(b).



(a) At the bottom of the dam

(b) At the start of the riverbed

Figure 5.18: A concrete slope

The main deformation on this slope is erosion caused by the water flow. The size of this slope is approximately 40 *m* × 18 *m*. This concrete slope has been monitored since 2009. At the beginning, the data acquisition was quite frequent, one measurement during two months; while the acquisition frequency was later decreased to once per year; because it was found that the erosion of this slope is quite small. This reason might be (1) the material of the slope is concrete, and (2) the external force to change this slope is small like water flow. In total, we

---

[2] http://www.heig-vd.ch/

acquire four dense datasets captured by terrestrial LiDAR at four different epochs. Table 5.5 provides the details of these four point cloud datasets.

Epoch	0	1	2	3
Date of Collection	May 8, 2009	July 2, 2009	June 8, 2010	July 1, 2011
Number of Points	9 791 034	4 456 905	4 247 106	7 617 623
Number of Stations	4	3	2	2

Table 5.5: Details of the point cloud datasets at four epochs

5.4.2 Results by cell-based approach

Cell size is a significant parameter of the detection in cell-based approach. In the investigation of the cell-based approach, the cell size is set to $0.02m$, $0.04m$, $0.06m$, $0.08m$ and $0.1m$, respectively. Table 5.6 indicates the cell number with different cell sizes after split. We observe that the number of the cells is quite different according to the cell size, from 1,305 (with cell size as 0.1 m) to 31,802 (with cell size as 0.02 m). Obviously, with the increase of cell size, there will be less cells after split.

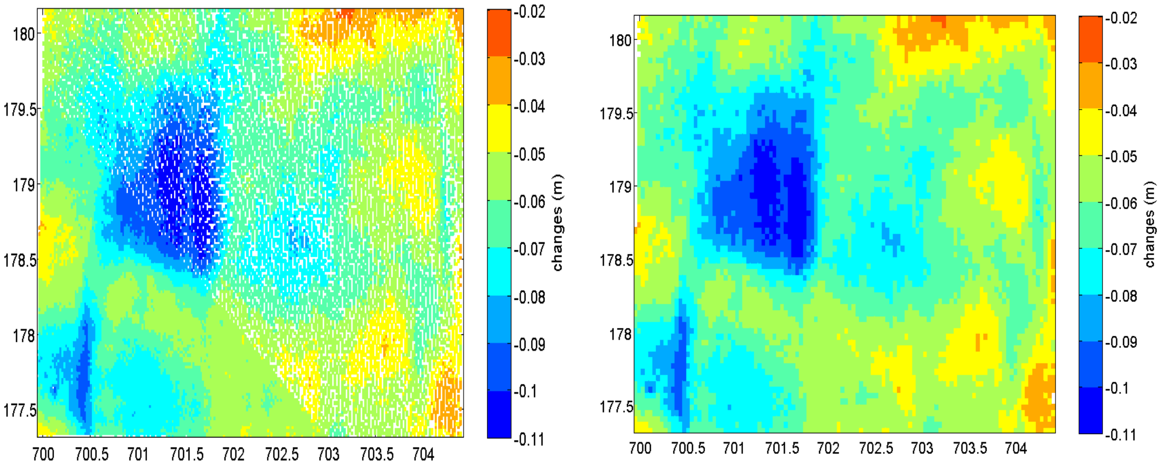
Cell size	0.02 m	0.04 m	0.06 m	0.08 m	0.1 m
Number of cells	31802	8034	3596	2016	1305

Table 5.6: Number of cells in cell-based approach with different cell sizes

Firstly, we compute deformation map by applying the cell-based approach on the point clouds from this concrete slope where the erosion existed. We test the five different cell sizes mentioned in Table 5.6. Figure 5.19 shows the five deformation map that we computed using the "min" method of Z axis as the meta-deformation of displacement. In general, the five sub-figures of deformation map all show similar deformation information. With the increase of cell size (from 0.02 m to 0.1 m), we observe the deformation map becomes smoother.

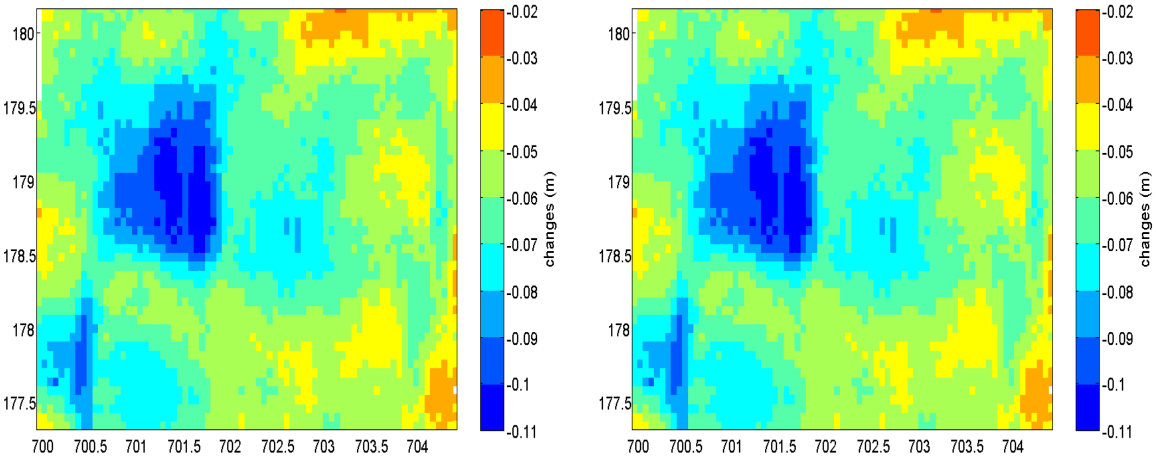
Additionally, we also compute the displacement distributions for these five deformation maps. Figure 5.20 shows the histograms of displacement between Epoch 0 and Epoch 2, corresponding to the five choice of cell size (0.02 m, 0.04 m, 0.06 m, 0.08 m and 0.1 m). The partial area with displacement from -0.06 m to -0.05 m is about 59% in this whole region. On the other hand, this region contains 5% area with some big displacement as -0.1 m. These subfigures show the high coincidence on the distribution of the displacement of this region.

We also tested different meta-deformation methods for detecting the displacement, such as "min", "max", "median", and "mean". From Figure 5.21, there is no significant difference between different methods. Displacement achieved by different detection methods show the analogous distribution; and the reason might be the surface is quite flat and regular in general.



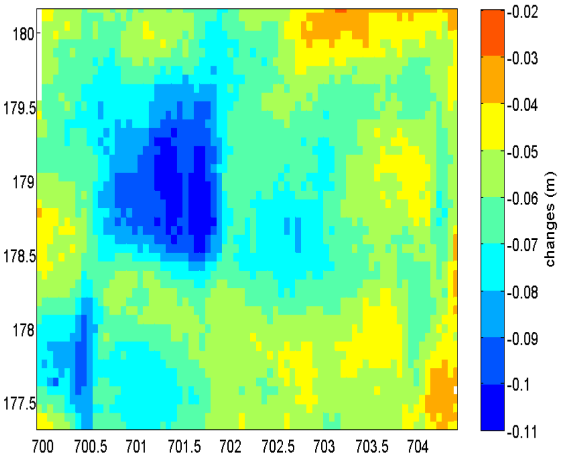
(a) 0.02 m

(b) 0.04 m



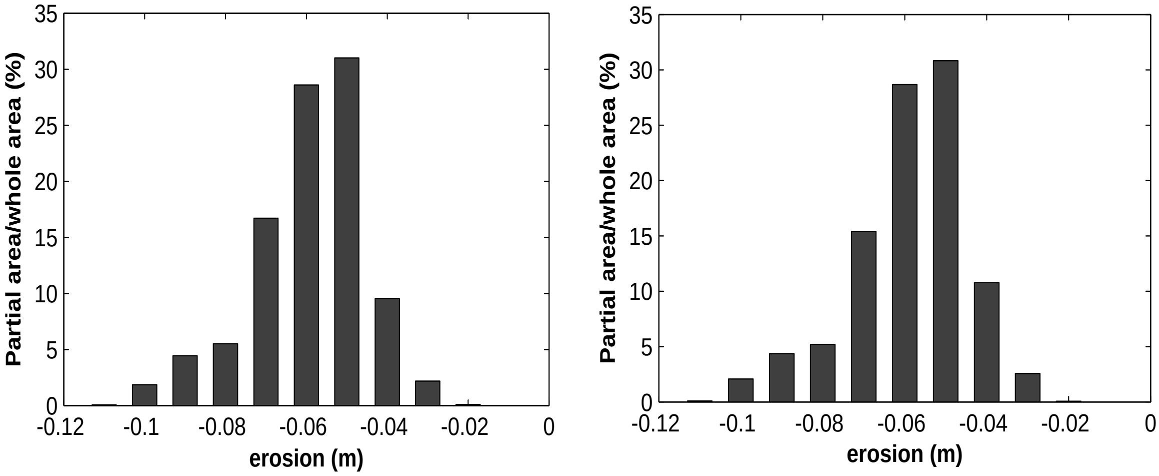
(c) 0.06 m

(d) 0.08 m



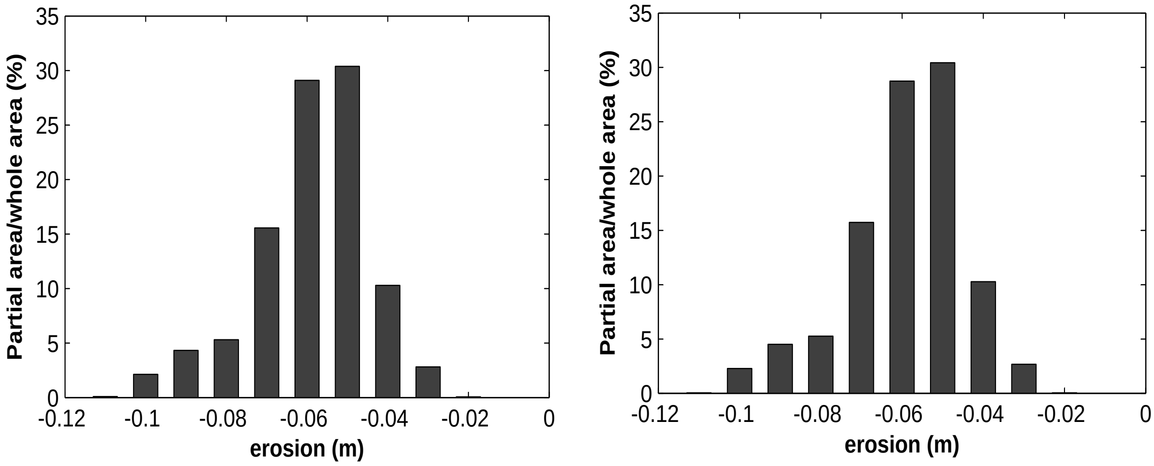
(e) 0.1 m

Figure 5.19: Deformation map of the point cloud between Epoch 0 and Epoch 2 (cell sizes are 0.02 m, 0.04 m, 0.06 m, 0.08 m and 0.1 m).



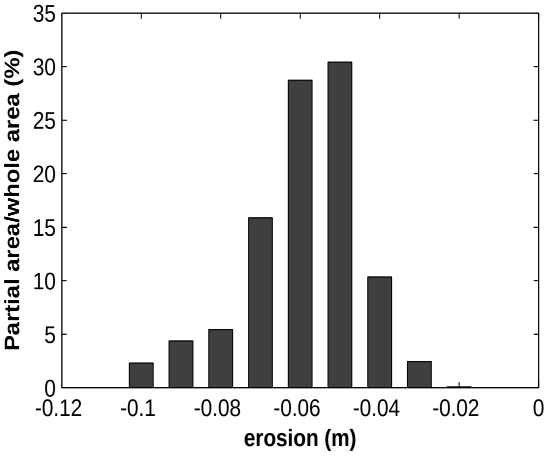
(a) 0.02 m

(b) 0.04 m



(c) 0.06 m

(d) 0.08 m



(e) 0.1 m

Figure 5.20: Distribution of displacement between Epoch 0 and Epoch 2 (cell sizes are 0.02 m, 0.04 m, 0.06 m, 0.08 m and 0.1 m).

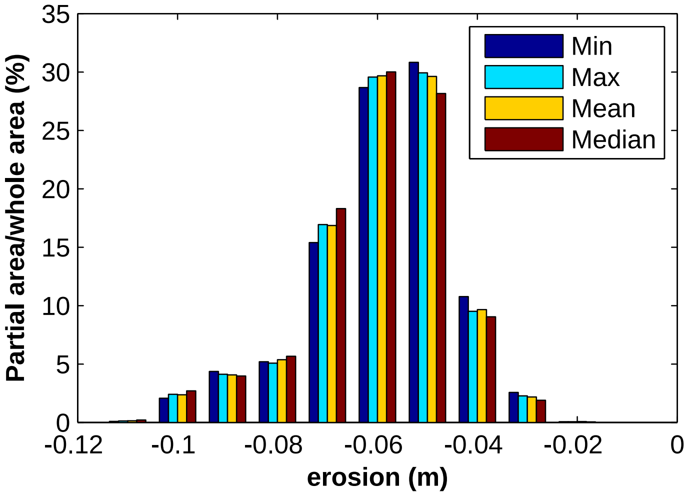


Figure 5.21: Distribution of displacement with different detection methods

5.5 Comparison

To further evaluate the deformation results by the cell-based approach, we compare our cell-based outputs with the results generated by 3rd parties, e.g., a commercial software (e.g., 3DReshaper), or a Swiss surveying company (e.g., BBHN SA). In this section, we provide the detailed comparison: we firstly introduce the 3DReshaper software and investigate its inspection function for displacement detection (Section 5.5.1); afterwards, we present the comparison based on the test of the same datasets used in previous sections, including the synthetic data with the ground truth (Section 5.5.2), and real-life datasets (Section 5.5.3 and Section 5.5.4) with other results.

5.5.1 Introduction of 3DReshaper

Before providing the comparison results, we introduce a commercial software – 3DReshaper, which is a well-known software for point cloud processing such as surface reconstruction, and calculation of geometric shapes. Afterwards, we provide the sensitivity analysis of data inspection to tune the important parameter in 3DReshaper (i.e., the average length of the triangle).

3DReshaper is a 3D scanner software dedicated to point cloud processing, 3D meshing, CAD surface reconstruction and three dimensional comparisons [3]. The working principle of inspection using 3DReshaper is to project the data points onto the base mesh. The main steps of using 3DReshaper to compare two point cloud datasets include: (1) generate a mesh based on the point cloud of the first epoch as a base geometry, (2) project the point cloud at the second epoch onto this base mesh and identify the difference.

Inspection Procedure – The inspection function is the core part of 3DReshaper, which is performed on two related objects, and one of the two objects is considered as *a reference geometry*. According to the types of the object, we have: (1) two surfaces or contours (e.g., a CAD model with the corresponding “as-built” model); (2) a surface and a point cloud. This function can not perform direct comparison on the two point clouds captured at different epochs. Therefore, we

---

[3] $http://www.3dreshaper.com/en1/En\_software.htm$

create a 3D mesh based on one point cloud (chosen as a reference geometry) and then compare this mesh with the other point cloud. In order to guarantee the direct comparison of the two objects, the reference geometry and the object must be registered in the same coordinate system as a preliminary step. This step includes rough registration and fine registration.

The comparison between the reference geometry and the other object is performed by projecting the object on the reference geometry. In the case of "erosion on a dam", the reference geometry is an arbitrary triangle mesh built on a point cloud at a chosen epoch; the other object is a point cloud at the other epoch. Figure 5.22 is a subset of point cloud data on the dam at Epoch I; and Figure 5.23 is the triangle mesh generated based on the points in Figure 5.22. Afterwards, the point clouds at later epochs (Epoch II, III an IV) are compared with Epoch I to analyse deformation.

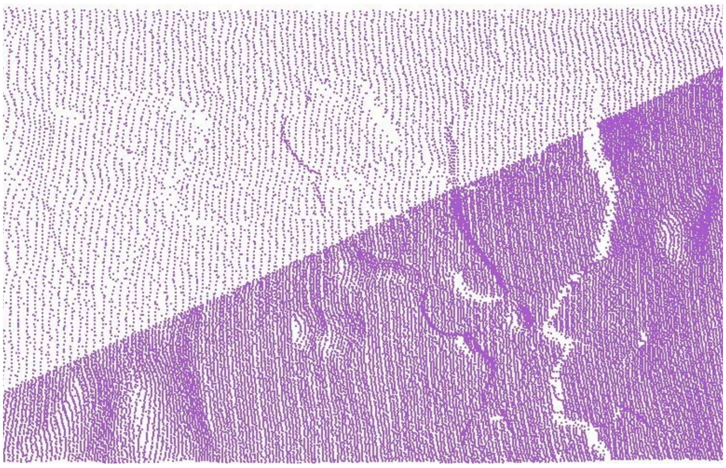


Figure 5.22: A point cloud at a chosen epoch (Epoch I)

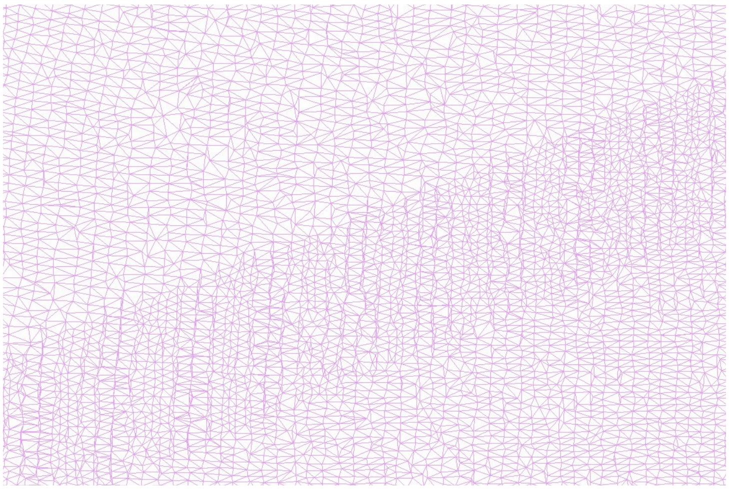
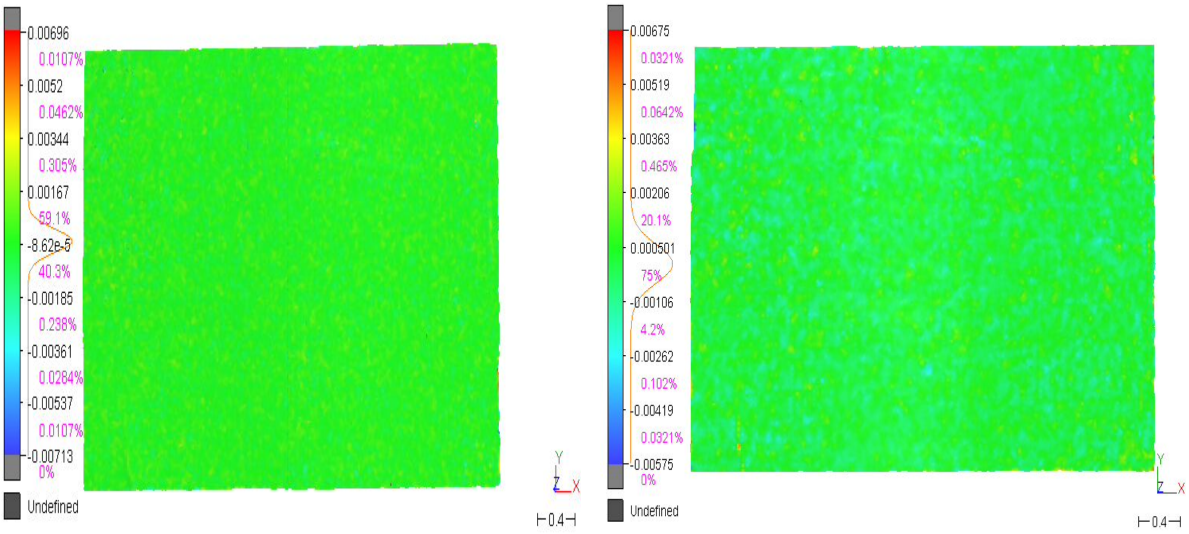


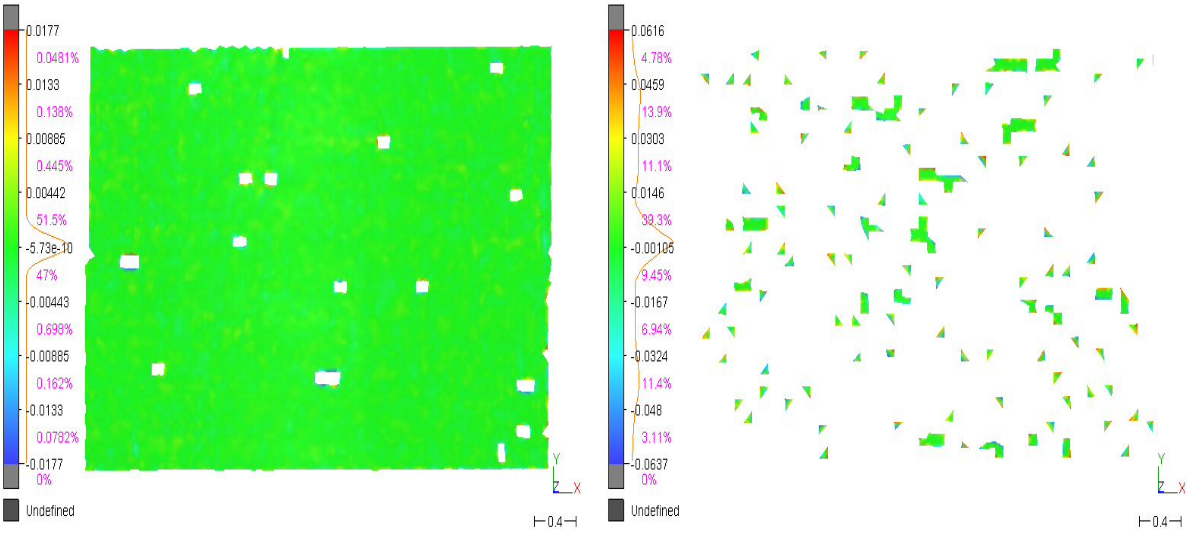
Figure 5.23: A mesh at a chosen epoch (Epoch I)

Investigation of inspection by 3DReshaper - To evaluate the quality of 3DResahper, we compare a point cloud with the mesh created on the same dataset, by testing different mash triangle length. The comparison result between this mesh and its original point cloud is theoretically zero but is experimentally in Gaussian distribution. When this mesh is closer to this point cloud, the inspected displacement is more accurate. In this case, the essential parameter to create 3D mesh is the average length of the triangles constructing the 3D mesh. Therefore, we focus on testing the same area and creating 3D mesh with different parameter setting. In the experiments, the average length of the triangles ($L_{triangle}$) is set to 0.02 m, 0.04 m, 0.06 m, 0.08 m and 0.1 m, respectively. Figure 5.24 is the inspection results corresponding to the four triangle length, i.e., 0.02 m, 0.04 m, 0.06 m, 0.08 m and 0.1 m respectively. The analysis of these four sub-figures is as below:



(a) 0.02 m

(b) 0.04 m



(c) 0.06 m

(d) 0.08 m

Figure 5.24: Inspection of the point cloud and its mesh at epoch 0 (the average length of the triangle $L_{triangle}$ are 0.02 m, 0.04 m, 0.06 m and 0.08 m, respectively).

- *Capability of Detection.* From the intuitive view of the figures, the detection capability becomes weaker and even unreliable with the increase of $L_{triangle}$. From Figure 5.24(a) to Figure 5.24(d), more errors appear with regard to increasing $L_{triangle}$. This worse situation is obviously described in Figure 5.24(c) and 5.24(d). When $L_{triangle}$ is 0.06 m, we can observe some empty spots in the comparison results, which are caused by the created mesh with the holes. The sizes of these holes are diverse from 0.1 m to 0.2 m. Furthermore, when this triangle length raises to 0.08 m, a mesh with substandard quality is created and leads to unreliable displacement shown by Figure 5.24(d). Finally, when $L_{triangle}$ is set to 0.1 m, 3DReshaper is unable to create the mesh.

- *Distribution of Displacement.* In the bar at the left side of each sub-figure in Figure 5.24, we observe the distribution of the displacement inspected by 3DReshaper. These distributions demonstrate the experimental hypothesis, i.e., Gaussian distribution. All distributions are almost symmetrical around 0 m, which is the theoretical displacement value. We have some detailed observations: (1) The maximum of the displacement grows dramatically with the rise of $L_{triangle}$ from 0.007 m to 0.0177 m; and even very unreliable like 0.06 m, and the minimum indicates the same growth to the minus direction. (2) The displacement with smaller length is more centralised than the displacement with larger length. For example, 95.9% and 96.3% displacement are between -0.00125 m and 0.00125 m, when the average triangle length are 0.02 m and 0.04 m, respectively; but, only 37.6% displacement are between -0.005 m and 0.005 m, when the length is 0.08 m. (3) The errors of the displacement ascend widely with the increase of $L_{triangle}$.

In addition to analysing Figure 5.24 produced by 3DReshaper, we explore the numeric output from 3DReshaper. We compute additional statistical properties including minimum, maximum, average of the displacement and standard deviation are summarised in Table 5.7. Firstly, compared with the minimum and maximum value from the figures, we can see that the displacement shown by Figure 5.24 are the subset of the numeric output. In fact, the trend of the minimum and maximum indicate the same change situation as shown in Figure 5.24, which is a dramatic increase as the rise of $L_{triangle}$. Secondly, the average value of the displacement are quite close to 0 and the standard deviation of the displacement is very small, e.g., around 0.001 m when the average length of the triangles are 0.02 m and 0.04 m; 0.003 m when $L_{triangle}$ is 0.06 m.

Average length of triangles ($L_{triangle}$)	0.02 m	0.04 m	0.06 m	0.08 m	0.1 m
Minimum	-0.019	-0.021	-0.053	-0.1	Error
Maximum	0.021	0.021	0.052	0.1	Error
Average	1.28134E-05	-4.31836E-08	-1.22745E-05	0.003291	Error
Standard Deviation	0.001138615	0.001404663	0.003017065	0.056921	Error

Table 5.7: Minimum and maximum of the inspection result by 3DReshaper

Investigation of detection by Cell-based approach - We also investigate the results achieved by our cell-based approach. In order to keep the consistency with the detection of 3DReshaper, we use the same dataset to evaluate our cell-based approach. In the previous investigation of 3DReshaper, the average length of triangles ($L_{triangle}$) can affect the detection of displacement. Similarly, cell size is a significant parameter of the detection in cell-based approach. To make a fair comparison, the same set of parameter setting (from 0.02 m to 0.1 m) is chosen for the investigation of the cell-based approach.

As described in Chapter 4, the cell-based approach has three main steps, i.e., split, detect, and merge. We first split the space into uniform cells, then the point cloud in each cell at Epoch I is compared with the point cloud in its corresponding cell at Epoch II. Like the inspection in 3DReshaper, we can do the same inspection that the point cloud in each cell is compared with itself. We get all zero differences – all displacement of the cells between the points in the same cells are 0, which is in contrast to the non-zero values in Table 5.7 from the inspection of 3DReshaper. This is because the cell-based approach is directly performed on the point cloud; but in 3DReshaper, the software builds an intermediate step (i.e., the mash created from the point cloud), which could bring additional noises/errors in analysing deformation. Therefore, we can call our cell-based approach is a kind of "point-to-point" comparison, while 3DReshaper is a "surface-based comparison" method.

5.5.2 Synthetic data - comparison with ground-truth

In Section 5.2.2, we have already presented the deformation map generated by the cell-based approach. For comparison, we first apply 3DReshaper on synthetic data to produce the displacement map. Therefore, we compare the results from both methods, surface-based approach (e.g., 3DReshaper) and the cell-based approach. Additionally, we compare the estimated results with ground truth using several statistical values.

Figure 5.25 is the displacement inspected by 3DReshaper. From the perspective of graphic view, Figure 5.25 shows a similar changing trend as Figure 5.4(c). From the left side of this figure, the displacement began to decrease from 0.07 mm to -14.2 mm until the middle of the slope; afterwards, the displacement continuously increased, until reaching 0.07 mm at the top of this slope.

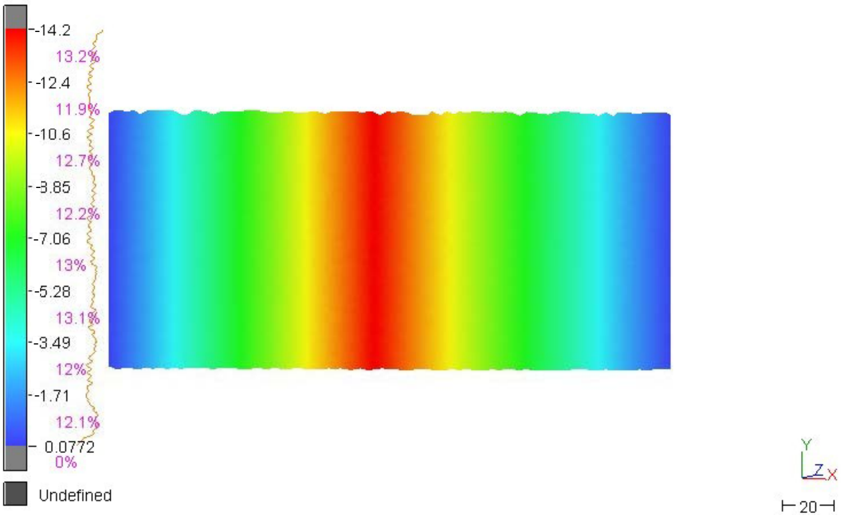


Figure 5.25: Displacement inspected by surface-based approach (3DReshaper)

In addition, we know the ground truth of the synthetic data, which can be used to evaluate the results from both 3DReshaper (surface-based) and our cell-based method. Table 5.8 shows some basic statistic values (e.g., *maximum*, *minimum*, *mean* and *standard deviation*) detected from both cell-based and surface-based method. From Table 5.8, we observed that for the first three values including *maximum*, *minimum* and *mean*, the results by the cell-based approach are closer to the ground truth, compared to the surface-based approach. Especially for the *maximum* value, the cell-based approach achieved the same value as the ground truth (0 *mm*), but the surfaced-based approach got 0.225 *mm*. We further checked the detailed detection results by the surface-based approach: 0.2% displacement (204 points out of the total points) are more than 0 *mm*. However, for the *standard deviation*, the surface-based approach is better than cell-based approach, like 3.81 $mm < 4.46$ *mm*. Therefore,the surface-based method is smoother compared to our cell-based method, which exactly shows one limitation of this cell-based approach: possibly losing some continuous due to the split of area into cells.

Methods	*Cell-based Method*	*Surface-based Method*	*Ground Truth*
maximum	0	0.225	0
minimum	-15.39	-14.2	-15.47
mean	-7.55	-6.56	-7.74
standard deviation	4.46	3.81	0

Table 5.8: Statistical analysis of deformation detection from the same datasets by cell-based and surface-based approach (unit: *mm*)

5.5.3 Case study I - comparison with the Results from BBHN SA

Different from the synthetic data, the real-life case studies (both the landslides in Flamatt presented in Section 5.3 and the erosion on a concrete slope presented in Section 5.4) do not have the ground-truth data. To further evaluate our results on these real-life data, we compare our results with the results provided by 3rd party sources.

For the landslides in Flamatt, BBHN SA (a private surveying company in Switzerland) also analysed this point cloud dataset. By analysing the vertical displacement of the terrain surface between Epoch I and Epoch II, they generated a contour map of the slope by using AutoCAD, as shown in Figure 5.26. The red lines are the intermediate contour lines. Each intermediate contour line represents an equal displacement and each pair of neighbouring intermediate contour lines have the same change of displacement. To better understand this map, additional yellow lines are the index contour lines that represent the displacement with the values of 0.5 *m*. The figure shows the whole contour map of the slope near the railway at Flamatt. To provide a more detailed view, Figure 5.27 is a zoom-in view of a small selected part of the whole contour map in Figure 5.26; and it highlights the bottom-left part of this slope (i.e., inside the ellipse).

Besides this contour map, we do not have any additional detailed information of the BBHN results. Nevertheless, we are able to provide a graphic comparison between this contour map with the deformation map computed by our cell-based method. We observe that the deformation map (Figure 5.13) shows similar displacement to Figure 5.26: firstly, the main subsidence of this slope is around 0 *m*; secondly, two hotspots in the middle have obvious increase, both of which

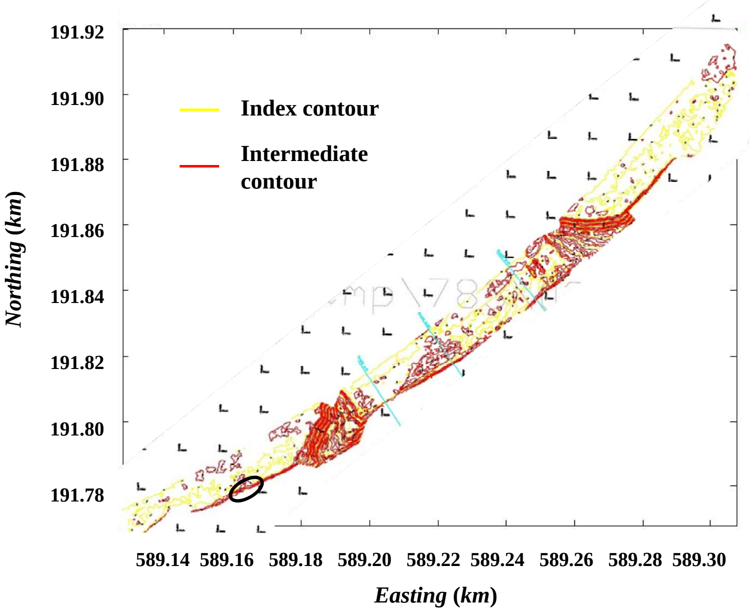


Figure 5.26: Contour map of the whole slope at Flamatt

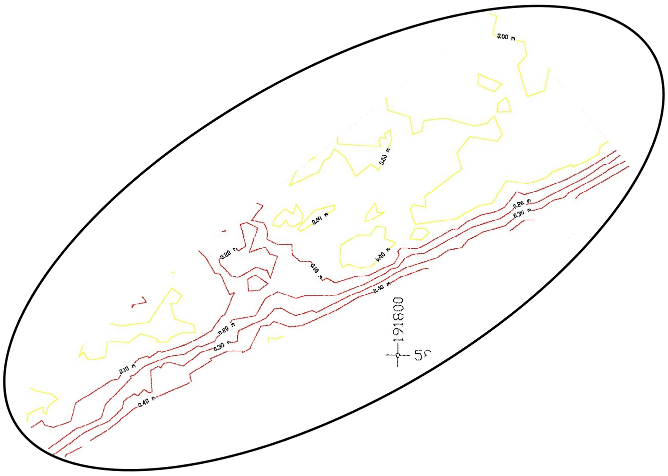


Figure 5.27: Contour map of the deformation for the zone marked in Figure 5.26 – inside the ellipse, which is enlarged by 15 times

are shown in our deformation map and the contour map from BBHN-SA; and thirdly, the lower edge of this embankment raise by 0.4 m. Without the ground-truth results, it is impossible to claim which results are better. Nevertheless, these similar features show the consistency between our approach and the BBHN results.

5.5.4 Case study II - comparison with surface-based approach

Now, we provide the deformation analysis results when the average length of triangles is set by 0.04 m, which is the best parameter studied in the previous section (from Table 5.7). We focus on two types of results from deformation detection: one is about the erosion visualisation, and the other is about deformation distribution.

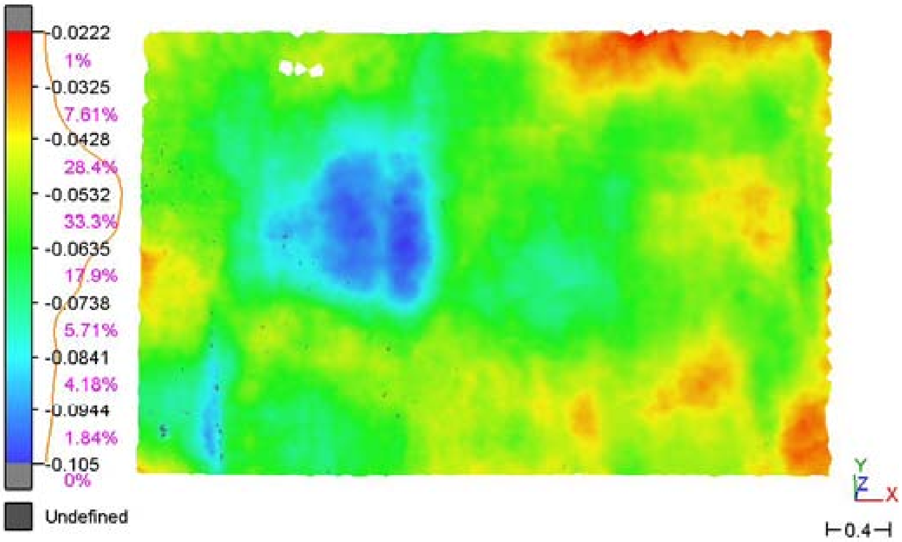


Figure 5.28: Erosion map of one area in the concrete slope by 3DReshaper

Erosion Analysis - Figure 5.28 is the final result generated by the surface-based method (3DReshaper), where the erosion changes from -0.105 m to -0.0222 m. The partial area with displacement between -0.0428 m and -0.0738 m is about 79.6% of this region.

Deformation Distribution - We analyse the displacement distributions of 3DReshaper for more statistical properties. Figure 5.29 is the histogram of all displacement in this region. The major displacement are centred around -0.06 m.

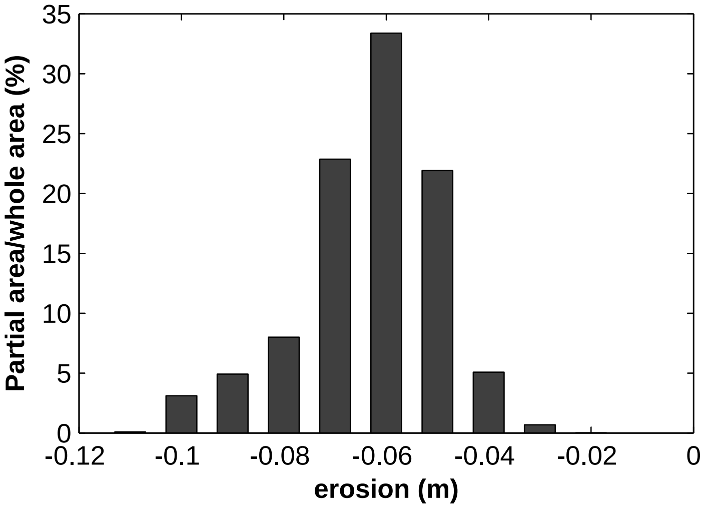


Figure 5.29: Deformation histogram by surface-based approach

With Section 5.4.2, we present the deformation (in terms of the displacement) of the same region obtained by 3DReshaper and the cell-based approach. From conceptual point of view, 3DReshaper is a *surface-based* method by building mesh surface to matching the points. Now, we compare the cell-based method with the surface-based method. Cell-based approach is directly performed on the point cloud and is aided by cells to detect deformation. In 3DReshaper, a surface-base method, a point cloud at Epoch I is firstly used to create a mesh as reference and then the second point cloud at Epoch II is used to match this surface. In this following paragraphs, we compare these displacement results of these two methods from two perspectives: one is the intuitive graphic view (comparing the erosion plot by 3DReshaper and the deformation map by cell-based approach), and the other the displacement distributions from the two methods.

Erosion Comparison - From the global visualisation, the erosion map in Figure 5.28 and the deformation maps in Figure 5.19 show very consistent results: similar areas with more or less the same displacement. The main erosion of this area (in green colour) is between -0.07 m and -0.05 m; the spot with the displacement between -0.09 m and -0.11 m (in blue colour) is described in Figure 5.28 and Figure 5.19(b).

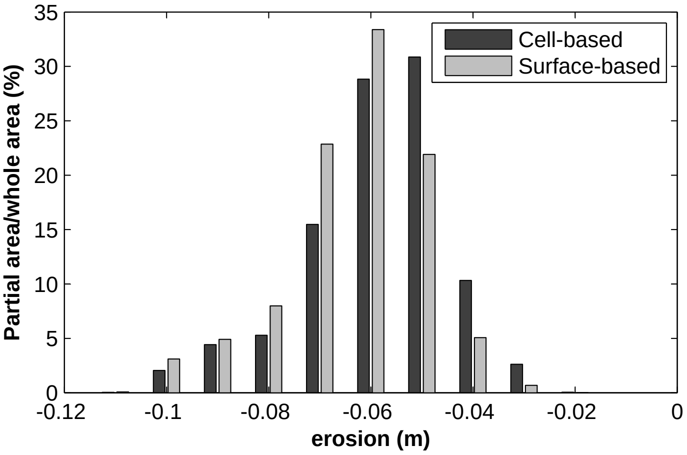


Figure 5.30: Histogram by cell-based and surface-based approach

Distribution Comparison - To further compare the erosion characteristics inside this area more precisely, we analyse the displacement distributions of our cell-based approach and the surface-based method (i.e., 3DReshaper). Figure 5.30 shows the two erosion histograms computed by our cell-based approach (dark) and by the surface-based method (grey), respectively. We can observe that the general deformation distribution is very similar. Firstly, the displacement achieved by two methods are ranging from -0.11 m to -0.02 m; secondly, the partial area with displacement between -0.07 m and -0.05 m accounts for 75.17% (cell-based approach) and 78.15% (3DReshaper) of this region. Such distribution similarity is consistent with the graphic one (erosion map vs. deformation map). The slight difference between these two results is that: in the results of our cell-based method, the most frequent magnitude of the erosion level is -0.05 m, while it is -0.06 m by using the surface-based method.

5.6 Summary

This chapter presented the third major contribution of this thesis, i.e., *evaluation of cell-based approach*. The cell-based approach was tested and evaluated by three point cloud datasets including (1) synthetic data generated by computer program; (2) real-life LiDAR data of landslide in Flamatt; and (3) real-life LiDAR data of erosion on a concrete slope. In addition, we also built some practical experiments, and collected point cloud data of the rockfall scenarios, which will be used in Chapter 6 for finding cell correspondence.

Through the synthetic point clouds, we compared the cell-based approach with non-cell based traditional method. We presented its advantages in deformation detection of distinct partial areas of the object. It is worth noting that we showed better accuracy using cell-based detection compared to non-cell based detection, based on the ground truth from the simulated data. Our cell-based method supports meta-deformation like displacement, angle variation and rigid transformation. Besides the synthetic data, we considered two real-life deformation cases. For the first case – landslide in Flamatt, we applied our cell-based approach and compared it with the results from a surveying company. For the second case – erosion on a concrete slope in Valais, we performed analysis by the cell-based approach and demonstrated a comparison with surface-based method (e.g., 3DReshaper, a commercial software for point cloud processing). For both real-life cases, the cell-based approach showed the consistent results with the analysis from this surveying company and commercial software. From the evaluation of various datasets, we concluded that the cell-based approach can be a complementary method of traditional and current processing techniques.

Chapter 6

Extension of Cell-based Approach

Stay hungry, Stay foolish.

Steve Jobs (1955-2011)

6.1 Introduction

This chapter presents some important extensions to our cell-based approach, providing more rich functionalities for deformation monitoring via 3D point clouds. The extensions are corresponding to different steps in the cell-based deformation computation framework presented in Chapter 4, particularly on the steps of split and detection.

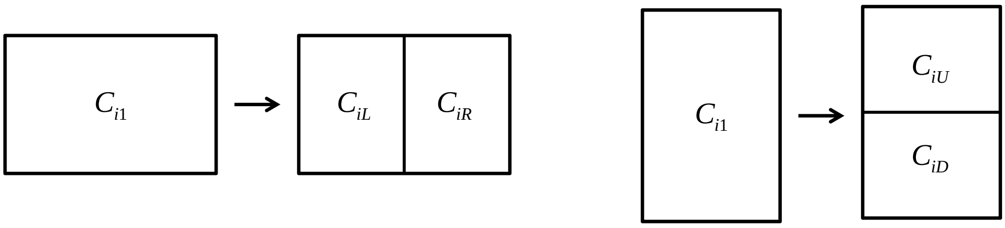
This chapter is organised as follows: Section 6.2 discusses the strategy of designing dynamic cell sizes for building meta-deformation, as an extended split method. Section 6.3 additionally discusses another research issue about “finding cell correspondence” when large deformation causes most points in one cell move to another cell. Section 6.4 proposes a further research issue about “movement tracking with data-driven cell” when deformed parts have an arbitrary boundary that cannot use a regular cell shape. Section 6.5 presents a rich detection method that is able to adaptively select a suitable model for each cell. We also provide the extension of computing deformation evolution in Section 6.6 where more than two epochs of point cloud datasets are available. Finally, Section 6.7 summarises this chapter.

6.2 Dynamic Cells

In Section 4.3 about the *split* step in the cell-based deformation computing framework, we have already presented a method that divides the space of the monitored object into uniform cells. This is a simple static method since all cells are fixed before detecting meta-deformation, based on a given cell size $\langle length, width, height \rangle$ in advance. All cells have a uniform size. This method has some obvious disadvantages, e.g., (1) All cells have the same size, while they do not have the same number of point cloud data points inside, e.g., some are quite sparse while others may be very dense. In such case, the quality of meta-deformation built for a cell largely depends on how many points inside the cell. (2) In particular, the data points in the corresponding cell at two different epochs maybe have very different status, which could also affect the final meta-deformation results. (3) Additionally, although we have studied on the sensitivity of using different cell size, it is challenging to find the best cell size. If the cell size is too small, there could be a huge amount of cells, which could cause large computation cost for building meta-deformation for so many cells (i.e., overestimated-split); on the contrary, if the cell size is too large, there might be too many points in one cell, then one single meta-deformation model will be difficult to capture complete deformation information (i.e., underestimated-split). Therefore, the first major extension on previous works is to provide an enhanced split strategy that is able to support dynamic cells. Compared with simply split using uniform cells, the strategy of using dynamic cells can provide various suitable sizes for different cells, to avoid both problems of overestimated-split and underestimated-split.

To generate *dynamic cells*, we design a *top-down split* strategy to incrementally divide the space into cells with different sizes, considering the geometric features of points inside the cell. In this thesis, we apply the modelling errors (e.g., a surface regression on the points in a given cell – the same model used in Chapter 4 to determine the cell centre) to determine whether the cell needs to be further split or not. If the cell modelling has big errors at one of the two epochs (see Equation 6.1), this cell will be divided into two cells; otherwise, we do not split this cell. For the detailed procedure of splitting a cell, this is built on the cell characteristics, e.g.,: (1) If cell length is longer than cell width, the cell is split into the left and right sub-cells, namely $C_i \rightarrow (C_{iL},\ C_{iR})$, as shown in Figure 6.1(a). (2) If cell width is longer than length, the cell is split into the up and down sub-cells, namely $C_i \rightarrow (C_{iU},\ C_{iD})$, as shown in Figure 6.1(b).

$$\begin{aligned} error(\mathcal{M}(cell_i^{(I)})) > \Delta \quad or \\ error(\mathcal{M}(cell_i^{(II)})) > \Delta \end{aligned} \tag{6.1}$$



(a) $length > width$ (b) $length < width$

Figure 6.1: Split strategy

Algorithm 6.1 summarises the detailed procedure of generating dynamic cells. This is comprised of two main parts, i.e., "initialisation" and "iteration":

Step 1. *Initialisation* (line 2 - 14) – We firstly initialise the split process and set up a couple of important variables. We use $finalCellSet$ to save the final output, i.e., all cells without any needs to be further split, use *cellQueue* to manage the candidate cells that might require further split, use Δ as the model error threshold to determine whether we should split the candidate cell or not. We load all data points into $\mathbf{P}^1$ and $\mathbf{P}^2$, corresponding to two epochs, respectively. Afterwards, we detect the max bound of all points in Epoch 1 and Epoch 2. Based on this bound, we build the first candidate cell (*cell*) containing all points, and put *cell* into *cellQueue*.

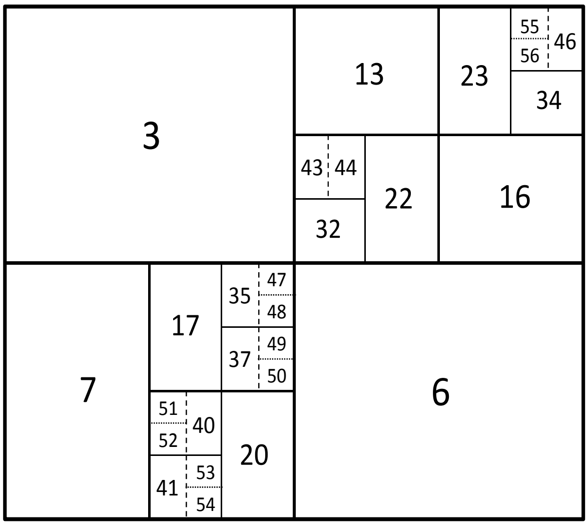
Step 2. *Iteration* (line 15 - 32) – At each iteration step, we take the first candidate cell from the *cellQueue*, modelling the point cloud dataset in the cell using a surface model (e.g., building a fitting plane via SVD or RANSAC) for both epochs, and receive the fitting error of the two fitted surfaces at two epochs (i.e., $error1$ and $error2$). If one of the two errors is larger than the pre-defined threshold Δ, the cell needs to be split

Algorithm 6.1: *dynamicCells*

```
   Input: Two files of original point cloud datasets (filename1, filename2)
   Output: Dynamic cells $finalCellSet$
1  begin
2      $finalCellSet \leftarrow \varnothing$; // initialise the output cellset
3      /* load all points of each file into $\mathbf{P}^1$ and $\mathbf{P}^2$, respectively */
4      $\mathbf{P}^1 = \{p_1^1, \cdots, p_N^1\} \leftarrow$ loadFile(filename1);
5      $\mathbf{P}^2 = \{p_1^2, \cdots, p_N^2\} \leftarrow$ loadFile(filename2);
6      /* find the bound of two point clouds $\mathbf{P}^1$ and $\mathbf{P}^2$ */
7      $boundP \leftarrow \{min_x, min_y, max_x, max_y\}$;
8      /* create one cell for containing all of the points in $\mathbf{P}^1$ and $\mathbf{P}^2$ */
9      $cell \leftarrow$ createCell($\mathbf{P}^1, \mathbf{P}^2$);
10     /* Initialise a cell queue*/
11     $cellQueue \leftarrow \varnothing$;
12     cellQueue.add(cell);
13     /* define the error threshold for accepting the current cell size */
14     $\Delta \leftarrow$ define(threshold);
15     while $cellQueue \neq \varnothing$ do
16         $cell \leftarrow$ cellQueue.poll( ); // take the first cell from the queue
17         /* build two models to represent $\mathbf{P}^1$ and $\mathbf{P}^2$, respectively */
18         $error1 \leftarrow$ model($\mathbf{P}^1$, cell);
19         $error2 \leftarrow$ model($\mathbf{P}^2$, cell);
20         if $error1 > \Delta \,||\, error2 > \Delta$ then
21             $length \leftarrow max_x - min_x$; // get length of cell $C_i$
22             $width \leftarrow max_y - min_y$; // get width of cell $C_i$
23             if $length > width$ then
24                 $\{cell_L,\ cell_R\} \leftarrow$ split(cell); // divide cell into two parts (left and right)
25                 cellQueue.add($cell_L$); // add the split cells into cellQueue
26                 cellQueue.add($cell_R$);
27             else
28                 $\{cell_U,\ cell_D\} \leftarrow$ split(cell) ; // divide cell into two parts (up and down)
29                 cellQueue.add($cell_U$); // add the split cells into cellQueue
30                 cellQueue.add($cell_D$);
31         else
32             finalCellSet.add(cell); // add non-split cell into final cellset
33     return finalCellSet // cells with dynamically generated sizes
```

using the strategy in Figure 6.1, i.e., either left-right split or up-down split based on the comparison of cell length and width; then the two sub-cells will be put into the cell queue *cellQueue* as the candidate cells for future check. If the two errors are both not larger than Δ, current cell is not necessary to be split, and will be added to the final result $finalCellSet$. The iteration step will be terminated until all of the cells in *cellQueue* are detected, i.e., the *cellQueue* becomes empty.

To further explain this algorithm, we provide an illustrative example in Figure 6.2. Initially, the whole space coving the entire monitored object at both epochs is considered as the first and biggest cell ($cell_0$, numbering as “0”). The modelling of point clouds in $cell_0$ does not contain small errors at both epochs, therefore $cell_0$ is split into $cell_1$ and $cell_2$ (a left-right split). Afterwards $cell_1$ and $cell_2$ are similarly split into two groups of sub-cells ($cell_3$, $cell_4$) and ($cell_5$, $cell_6$), respectively; $cell_1$ and $cell_2$ both apply up-down split. At this level, the modelling of $cell_3$ and $cell_6$ have small errors for both epochs (i.e., not satisfying the condition of Equation 6.1), therefore $cell_3$ and $cell_6$ do not need to be further split. On the contrary, $cell_4$ and $cell_5$ still need to be split to get smaller modelling errors on new split cells. We continue this iterative procedure. Finally, we reach the last state of Figure 6.2(a). In total, the space is divided into 28 final cells with 7-levels (i.e., the 7 different sizes), as summarised in Figure 6.2(b).



(a) *Split cells* (*the first and biggest cell is entitled* 0)

Level	***N_cell***	***Cell name***
1	2	3, 6
2	1	7
3	2	13, 16
4	4	17, 20, 22, 23
5	2	32, 34
6	7	35, 37, 40, 41, 43, 44, 46
7	10	47, 48, 49, 50, 51, 52, 53, 54, 55, 56

(b) *Final cellset*

Figure 6.2: An example of dynamic cell split

It is worth noting that the threshold (Δ) in Algorithm 6.1 is a very important parameter to determine whether a cell needs to be further split or not. Like many parametric setting issues, this is a challenging task. In practise, the threshold (Δ) typically depends on the application scenario, and needs to be tuned accordingly. Nevertheless, the objective of using dynamic cells can guarantee that all cells have similar and reasonable data points inside, such that a single model for the cell is optimised.

6.3 Cell Correspondence

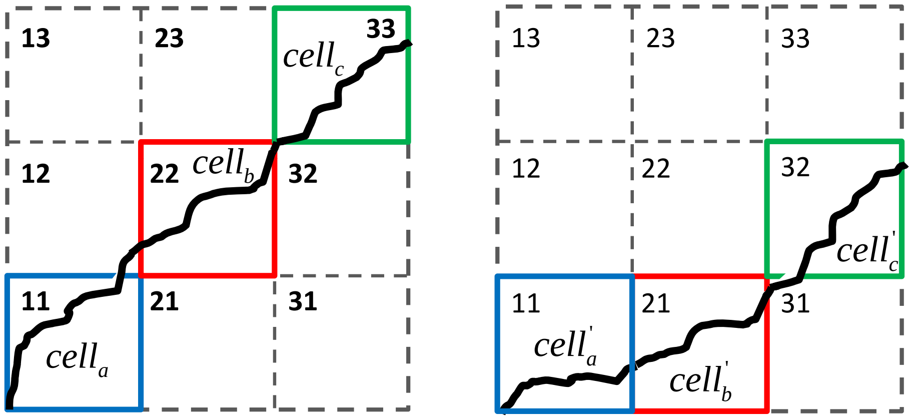
Up to now, we have presented techniques about using static cell split (uniform cell size) or building dynamic cells (various cell size) in our cell-based deformation computation. There is a common assumption that there exists a fixed cell correspondence for point clouds between Epoch I and Epoch II, i.e., $cell_i^{(I)} \rightarrow cell_i^{(II)}$ with the same ID i. The meta-deformation of $cell_i$ is computed based on comparing the two data clouds inside the cell at the two epochs. This kind of detection is suitable for small changes, which means that the points in the cell after deformation mainly stay in the same cell in general; namely, points in $cell_i^{(I)}$ largely stay in $cell_i^{(II)}$ as well after deformation. However, some deformation might be large for some cells, the deformable points moved from one cell to another cell (maybe neighbouring cell or maybe a cell far away). Therefore, our next extension is to develop a flexible searching method to automatically find the cell correspondence, not simply using a fixed $i \rightarrow i'$ correspondence.

To better understand this problem and explicitly explain the research challenge, Figure 6.3 provides an illustrative example. The bold curve represents the slope that has deformation from Epoch I to Epoch II. The slope is covered by a large area that is divided into 9 cells at both epochs using a simple uniform cell split strategy; thus, we generate 9 cells in total, i.e., from $cell_{11}$, $cell_{12}$ to $cell_{33}$. There are three main valid cells that cover the slope at Epoch I, i.e., $cell_{11}$, $cell_{22}$, and $cell_{33}$, which are also denoted as $cell_a$, $cell_b$, and $cell_c$, respectively. In terms of building meta-deformation using corresponding cells, we cannot build a simple direct $i \rightarrow i'$ correspondence like "$cell_{11}^{(I)} \rightarrow cell_{11}^{(II)}$", "$cell_{22}^{(I)} \rightarrow cell_{22}^{(II)}$", "$cell_{33}^{(I)} \rightarrow cell_{33}^{(II)}$". Such direct $i \rightarrow i'$ correspondence in this scenario is wrong: slope in $cell_a$ (i.e., $cell_{11}$) at two epochs keeps the right correspondent points (see blue rectangles); however, different from $cell_a$, the two other cells ($cell_b$ and $cell_c$) do not have such direct correspondence relationship. Points in $cell_b$ move to the down cell $cell_b^{'}$ in red rectangle; and points in $cell_c$ also change to the down cell $cell_c^{'}$ in green rectangle. Therefore, the correct cell correspondence is "$cell_{11}^{(I)} \rightarrow cell_{11}^{(II)}$", "$cell_{22}^{(I)} \rightarrow cell_{21}^{(II)}$", "$cell_{33}^{(I)} \rightarrow cell_{32}^{(II)}$". In summary, we observe that some cells have the direct cell correspondences, while others change the correspondence. This is quite normal in many real-life scenarios. Therefore, for a complex deformation scenario like this, the research challenge is to find the correct cell correspondence and to get the right meta-deformation parameters.

To solve this problem, we develop a greedy algorithm to find such cell correspondences, which is a local matching problem. The detailed algorithm is presented in Algorithm 6.2. This algorithm has two main steps:

Step 1. Identify important cells in Epoch I that cover the deformation object, e.g., $cell_a$, $cell_b$, and $cell_c$ in Figure 6.3. By this step, we can extract key cells, and only focus on finding cell correspondence of these important cells. This could significantly reduce our searching time. For example, the scenario in Figure 6.3(a) only has 3 important cells out of 9 cells.

Step 2. For each key cell ($cell_i^{I}$) at Epoch I, identify a set of candidate cells ($\{cell_j^{II}\}$) in Epoch II that might be the corresponding cells. Afterwards, we build a matching function $\mathcal{F}$. As shown in Equation 6.3, $\mathcal{F}$ could be a joint meta-model like ICP to find out the



(a) Epoch I (b) Epoch II

Figure 6.3: Cell correspondence between Epoch I and Epoch II

Algorithm 6.2: ***cellCorrespondence***

```
Input: Two point cloud datasets in cells ($CELL^I = \{cell^I_m, \cdots, cell^I_n\}$,
       $CELL^{II} = \{cell^{II}_p, \cdots, cell^{II}_q\}$)
Output: cell correspondence pairs $\langle cell^I_i, cell^{II}_j \rangle$
begin
    $CELL_{deform}$ ← getMainCells($CELL^I$); // get the main cells including deformation
    foreach $cell^I_i \in CELL_{deform}$ do
        /* find a set of candidate corresponding cells at Epoch II */
        $candidateCells$ = subSet($CELL^{II}$);
        /* build meta-model for each cell in cell */
        metaModel($\{cell^{II}_p, \cdots, cell^{II}_q\}$);
        /* perform comparison to search the correspondent cell in candidateCells */
        $cell^{II}_j \leftarrow \underset{cell^{II}_j \in candidateCells}{argmin} \mathcal{F}(cell^I_i, cell^{II}_j)$ ;
        get one cell correspondence $\langle cell^I_i, cell^{II}_j \rangle$
    return all corresponding cell pairs
```

transformation parameters between $cell^I_i$ and $cell^{II}_j$, or an independent meta-model ($\mathcal{M}$) for both cells, e.g., the point cloud surface regression that we used a lot in the previous chapters and sections for computing meta-deformation. After checking all candidate $cell^{II}_j$, we pick $cell^{II}_j$ that can achieve the most similar modelling parameters from $\mathcal{F}(cell^I_i, cell^{II}_j)$; we claim such $cell^{II}_j$ as the corresponding cell of $cell^I_i$.

$$correspondence(cell^I_i) \leftarrow \underset{cell'_j \in candidateCells}{argmin} \mathcal{F}(cell^I_i, cell^{II}_j) \tag{6.2}$$

$$\mathcal{F} = \begin{cases} metaModel(cell^I_i, cell^{II}_j) \\ ||metaModel(cell^I_i) - metaModel(cell^{II}_j)|| \end{cases} \tag{6.3}$$

6.3.1 Practical data collection

We test the proposed algorithm on a dataset collected by practical settings in an experimental environment. We design practical experiments to simulate typical deformation scenarios, and use terrestrial LiDAR to capture the point cloud datasets. Our practical experiments focus on the scenarios of slope movement. Slope movement is a typical and complicated deformation process, which can be affected by geomorphic factor (such as slope gradients, overburden depth, and structural rock properties), hydrologic factor (like water content and soil pore water pressure) and vegetative factor. Landslide processes can be classified into six dominant types including *falls*, *creep*, *slumps and earthflows*, *debris avalanches and debris flows*, *debris torrents*, and *bedrock failures* [Var78]. However, these landslides descriptions do not provide precise geometrical changes of the materials on the slope. In fact, each type of landslides is not one kind of single geometrical change, but a combination of some micro geometrical changes including displacement, translation and rotation, change of scale, etc. According to the location of the changing object, we can divide the complete monitored area into two parts: (1) an object on the slope and (2) the slope itself. Therefore, we design two experiments: (1) simulating the movement of a specific object on the slope, where the deformation focus is on the object not the slope; and (2) simulating the movement of a slope itself, which is caused by extra forces such as water or fallen materials designed in our practical experiments to imitate real-life deformation.

Movements of an object on the slope – Movements such as *displacement, rotation* and *translation* are the general deformation types in many real-life cases, e.g., landslide process. It is a fundamental step to simulate these basic deformation types and obtain point clouds for subsequent analysis. The objective of this experiment is to simulate movements of an object on the slope from Epoch I to Epoch II. In this situation, position of an object is changed in terms of a specific movement at different epochs.

Experiment design – We use an object *Obj* (e.g., a rock) to simulate its movement on the slope S as shown in Figure 6.4. To better describe the movements of this object, we built a suitable coordinate system of this object: Z-axis is the same as the the normal of this slope to the up direction; Y-axis is the perpendicular to Z-axis and is the same as the up direction of the slope; X-axis is perpendicular to Y- and Z- axis and horizontal to the slope. This coordinate system is depicted in Figure 6.4.

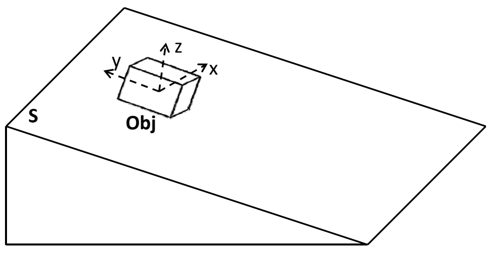
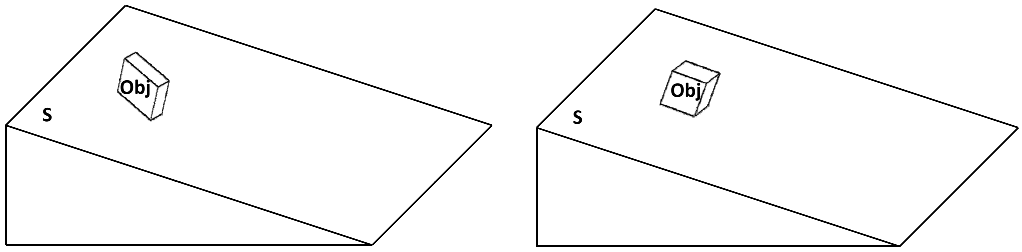


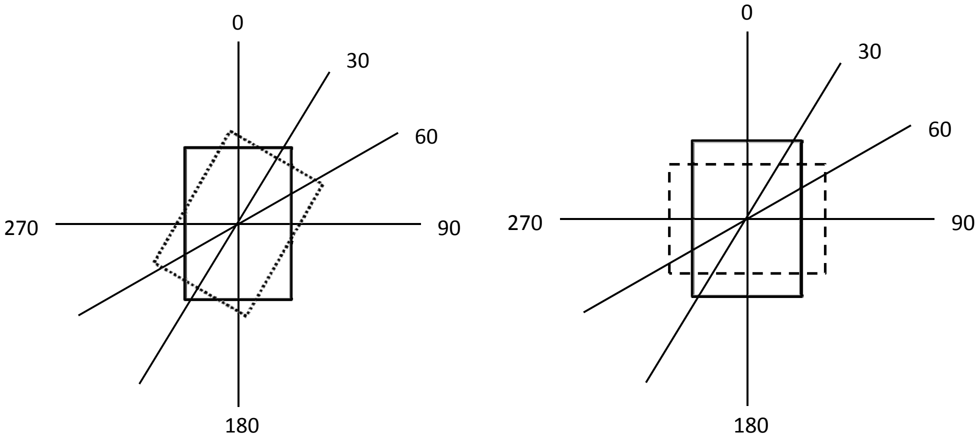
Figure 6.4: An object on a slope

The rotation of this object is designed as below. This object starts to rotate around its origin and Y-axis with small angle (30°). Figure 6.5(a) show the object's rotation on the slope and Figure 6.6(a) show the rotation of this object from the top-down view. After three times rotation with small angle, the position of this object is described in Figure 6.5(b) as well as the top-down view shown in Figure 6.6(b). Afterwards, the object is rotated with big angle (90°) three times until it is returned to the initial position.



(a) The position of this object after rotated 30° (b) The position of this object after rotated 90°

Figure 6.5: Rotation of an object on a slope. Figure 6.5(a) is the position of this object on the slope after rotating 30°; Figure 6.5(b) is the position of this object after rotated 90°.



(a) Rotation with small angle 30° (b) Rotation with big angle 90°

Figure 6.6: Rotation of an object on a slope (Top view). Figure 6.6(a) is the position of this object on the slope after rotating 30°; Figure 6.6(b) is the position of this object after rotated 90°.

Additionally, we simulate the translation of this object on the slope. The object translates from the initial position to the bottom of the slope (see Figure 6.7).

Experiment Setup - We perform this experiment under the support of and collaboration with the Institute of Geomatics and Risk Analysis (IGRA), University of Lausanne (UNIL)[1]. During the experiment of this rockfall simulation, the point cloud datasets of the monitored object are captured by VIVID 9i (see Figure 6.8), a terrestrial laser scanner produced by Konica Minolta company[2]. This Vivid 9i laser scanner applies the triangulation light block method, and its scan range is $[0.6 - 1.0m]$ in standard mode and $[0.5 - 2.5m]$ in in extended mode. The precision

---

[1] http://www.unil.ch/igar

[2] http://www.konicaminolta.com/instruments/products/3d/non-contact/vivid9i/index.html

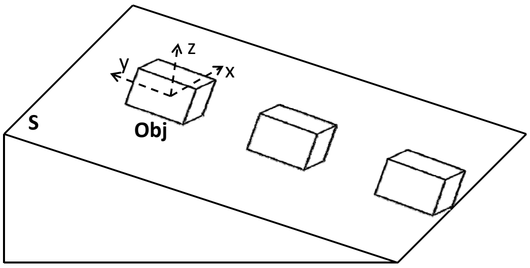


Figure 6.7: Translation of an object on a slope

of x-, y-, and z- coordinates is ±0.05 mm (using TELE lens at distance of 0.6 m, with field calibration system, Konica Minolta's standard, at 20°C). It takes 2.5 seconds for a scan and approximately 1.5 seconds to transfer the data to the host computer.

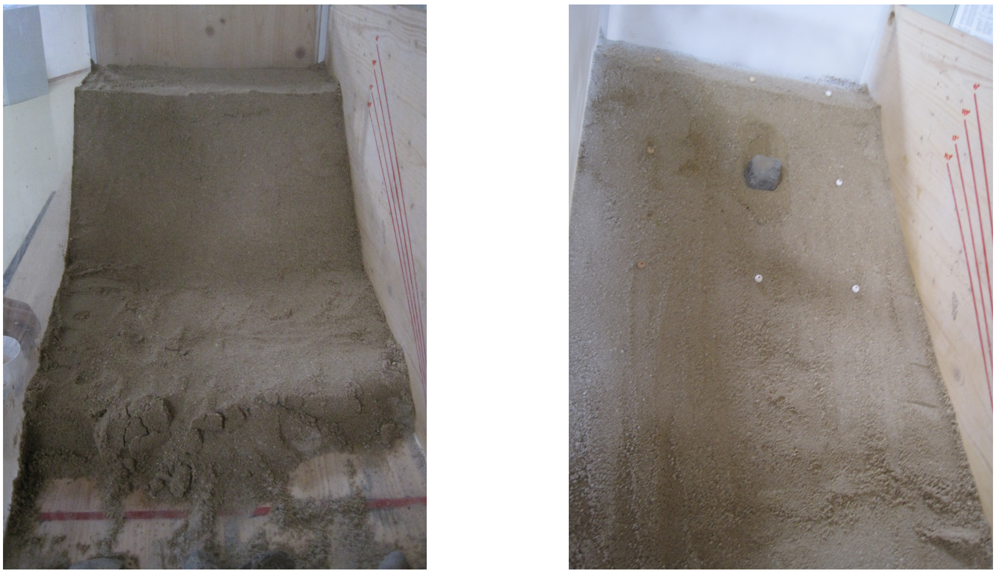
Figure 6.8: KONICA MINOLTA VIVID 9i laser scanner

Another equipment is called "Sandbox", which is chosen to set up the designed experiments. Figure 6.9(a) shows the components of Sandbox. It is a box with three sides made of wood and one side made of glasses to support easy observation. There is sand inside this box which can form different shapes as needed, e.g., a sand slope with different angles, which are marked in red line at the right side of the board.

We select one stone as the monitored object and put it on the sand slope inside the sandbox, as shown in Figure 6.9(b). The initial position of this object and the slope is scanned, and we get the first epoch of point cloud data. Afterwards, we give forces to the object and let it rotate along it's own Y-axis and slide along the slope until it moves to the bottom of this slope. During this period, we capture four epochs of point cloud datasets.

Movements of a slope – In another practical experiment, we simulate the movement of the slope itself. In such case, the outer force is not performed on an object (like the stone in the previous experiment), but on the whole slope. Such force would be water (e.g., storm) and big rock-fall, for simulating the scenarios of landslides.

Experiment design - Figure 6.10 is a box containing sand at the left side. The size of this box is 1 $m \times 0.5$ $m \times 0.5$ m. Sand is settled to form a slope and angle between this slope and the bottom of the box can be adjusted as a pre-defined angle, e.g., 45° in Figure 6.10.



(a) Sandbox (b) An object added in Sandbox

Figure 6.9: Sandbox. Figure 6.9(a) is the Sandbox; Figure 6.9(b) is the Sandbox with an added object – a stone.

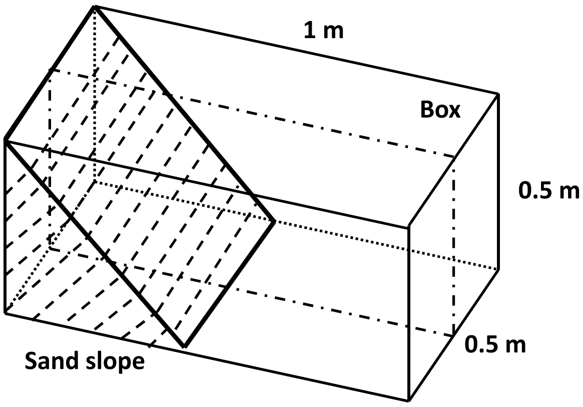
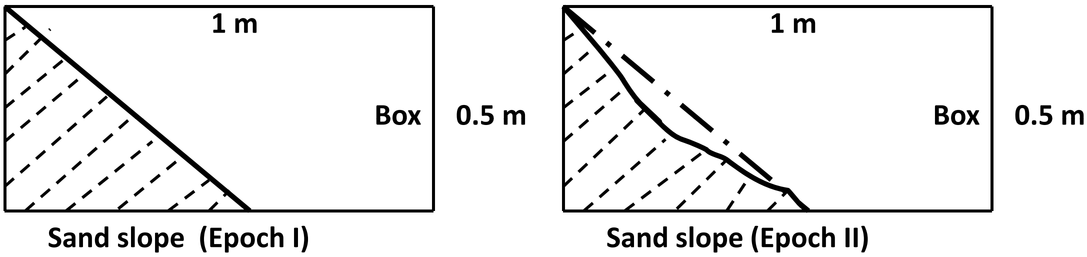


Figure 6.10: A sand slope in a box

We give some artificial forces from upside of this slope and cause changes of the surface of this slope. Considering various environmental phenomena in nature, two sorts of forces caused by water and fallen stones are applied on this slope. The strength of external forces is controlled by the volume of water/stones and the pressure given to these material. The profile of this slope along the long side of outside box (shown by dashed rectangle in Figure 6.10) are given by Figure 6.11. Artificial forces are given to the designed slope between two LiDAR scan measurements (i.e., Epoch I and Epoch II). Figure 6.11(a) is the profile of the original surface at Epoch I; solid line is the shape of the surface before giving forces. Figure 6.11(b) is the profile of the deformed surface at Epoch II. Solid curve is the shape of the surface after giving forces; and the dashed line is the original shape at Epoch I.



(a) Profile of sand slope at Epoch I

(b) Profile of sand slope at Epoch II

Figure 6.11: Changes of the slope surface at Epoch I and Epoch II

6.3.2 Finding cell correspondence

After collecting our practical datasets collected in the previous experimental setting, we design relevant algorithms and provide the detailed implementations for finding such cell correspondence in this dataset. As shown in Figure 6.12, we projected the four epochs of point cloud datasets in one picture (in order to have a global view about deformation), and obviously we observed that the four rock positions at the four epochs (from Epoch 1 to Epoch 4) are in different cells. Therefore, we need to apply the method (i.e., Algorithm 6.2) to find the corresponding cells that contain the rock in each epoch, i.e., $cell_a^{(1)} \rightarrow cell_b^{(2)} \rightarrow cell_c^{(3)} \rightarrow cell_d^{(4)}$.

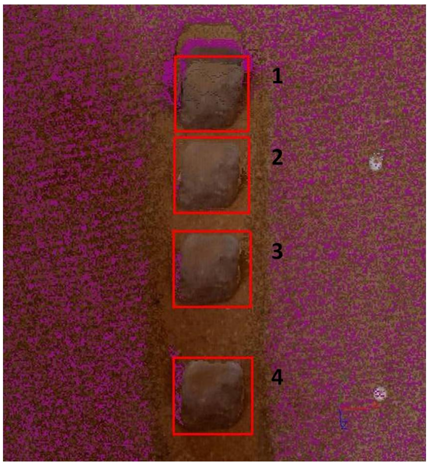


Figure 6.12: Practical experiment of a rockfall deformation on a slope (front view)

We choose point cloud datasets of this moving stone at Epoch 1 and Epoch 3 to test our cell-correspondence algorithm. The objective is to find the right cell position of this stone at Epoch 1 and Epoch 3. Firstly, we need to fix the initial position (i.e., the main cell) of the interesting target (i.e., the stone in this case) in Epoch 1; afterwards, based on the points inside each key cell, we search the corresponding candidate cells at Epoch 3; finally, we find the most suitable cell in the candidate cell as the final corresponding cell. There are two important experimental issues in Algorithm 6.2, i.e., (1) choosing the candidate cells (*candidateCells*) and (2) the meta model function (**metaModel**) to test the similarity of cells.

1. For the *candidateCells* in Algorithm 6.2, we divide the area into uniform cells (6 rows

× 3 columns = 18 cells in total), as shown in Figure 6.13. Yellow points in the middle cell of the first row and the purple points in the middle of the fourth row are the two corresponding cells that contain the data points covering this stone. For the detailed candidate corresponding cells, we consider all of the 18 cells. Of course, when checking a large area with a large number of cells, we can build a searching area contains possible candidate cells, rather than searching all cells. In practise, we typically only consider the direct corresponding cell and its neighbouring cells as the candidate cells.

2. For the meta-model in Algorithm 6.2, we apply the ICP method to model the data points in two cells (i.e., the $||ICP(cell_i^I, cell_j^{II})||$), and use two important outputs in ICP to check the similarity of the two cells in terms of data points inside. The two outputs of ICP are the number of ICP iteration (N) and the final root mean squared distance (MSD). Of course, the smaller of these two values, the two cells are more close and have higher probability to be the corresponding cells.

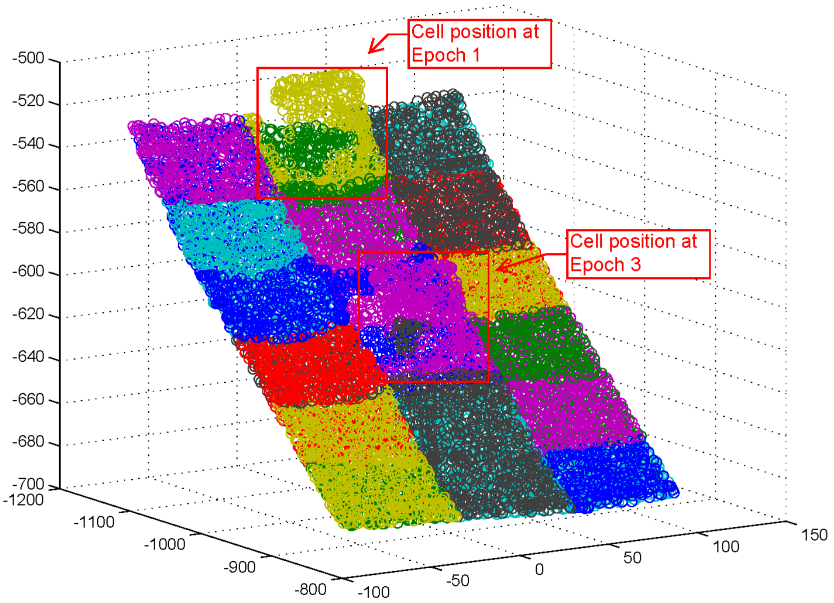


Figure 6.13: Split cells of a monitored slope at Epoch 1 and Epoch 3

For each cell at Epoch 1, we compare this cell with all 18 cells in Epoch 3. In this case, we apply ICP method to estimate the rigid transformation between the points at two cells. In order to find the initial and subsequent position of the stone at two epochs, we extract the iteration number of ICP (N) and the MSD (Mean Squared Distance) of points between 'before' and 'after' the transformation. The matrix in Table 6.1 shows the N values for two cells, i.e., $value_{ij}$ means building ICP from $cell_i$ at Epoch 1 to $cell_j$ at Epoch 3 requires $value_{ij}$ times to terminate. We observe that the iteration numbers of ICP in the diagonal of the table are significantly lower compared to other numbers in this table. For example, all the diagonal values are less than 10, and only two values is bigger than five; while, 99.4% of other non-diagonal values are bigger than 10. This means most cells have direct cell correspondence, i.e., "$cell_i^I \rightarrow cell_i^{II}$", which is true as there is only one non-direct cell difference that contains the stone, and other parts kept stable during this movement. These two cells are $cell_2$ at Epoch I and $cell_{11}$ at Epoch 3, which is consistent with the plot in Figure 6.13. This is consistent with the ground truth that we have cell correspondence: $cell_2^I \rightarrow cell_{11}^{II}$.

	1	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*	*13*	*14*	*15*	*16*	*17*	*18*
1	2	17	21	16	18	22	18	18	23	20	15	20	19	18	24	16	23	22
2	11	10	12	15	11	18	15	15	22	19	10	24	18	15	30	20	15	23
3	20	21	2	31	19	21	19	17	20	20	14	19	31	17	17	28	22	17
4	26	20	26	3	16	20	24	19	31	23	15	22	23	19	24	22	26	26
5	31	15	20	16	4	18	21	29	31	22	12	28	31	22	25	22	24	25
6	27	20	21	22	15	3	20	30	15	22	14	21	24	16	20	20	18	17
7	21	26	22	17	18	17	5	17	24	19	14	26	20	19	23	22	25	23
8	21	20	26	29	14	29	16	4	18	29	11	23	25	15	23	26	19	21
9	27	24	17	30	17	20	23	18	5	21	15	19	21	14	19	20	21	17
10	19	18	18	25	16	21	25	20	26	3	14	18	18	20	25	17	24	30
11	24	16	22	19	16	24	31	28	23	18	10	21	22	14	31	30	18	30
12	30	19	20	24	21	17	18	23	18	16	10	4	19	17	17	19	21	19
13	22	23	24	19	18	17	19	18	20	23	15	22	3	15	22	18	27	24
14	18	20	19	24	19	27	23	20	18	20	11	31	21	4	18	27	20	31
15	25	18	18	25	21	17	23	23	22	25	13	21	20	13	2	22	23	23
16	19	19	20	20	19	26	23	21	28	19	14	25	18	23	31	2	22	28
17	28	15	19	21	17	22	24	15	22	24	12	23	31	14	23	21	2	21
18	21	18	15	22	20	20	23	25	17	23	15	20	24	15	23	21	18	2

Table 6.1: Iteration number of ICP (N) between two cells at Epoch 1 and Epoch 3 ($Index_{cell}^1$ vs $Index_{cell}^3$)

In addition, Table 6.2 provides the MSD matrix between cells at two epochs. Similar to Table 6.1, the index i and j mean the ICP meta-model from $cell_i$ at Epoch 1 to $cell_j$ at Epoch 3, while the $value_{ij}$ means the mean squared error (MSD) for such ICP rigid transformation. The average value of all MSD in the matrix is 2.82 mm. For row 2 ($cell_2$ at Epoch 1), the average MSD in this row is 5.4 mm; similarly, we found column 11 ($cell_{11}$ at Epoch 3) has average MSD 6.35 mm. Compared to other cells with smaller average MSD, we can claim that $cell_2$ and $cell_{11}$ are mostly likely the cells containing the stone. $value_{2,11}$ is a small value in column 11 but not for row 2; ideally, $value_{2,11}$ should be the smallest value for both cases. But in our practical experiment, the stone has a very regular shape (like a cube), which is easy to be wrongly matched to the slope (other cells) by using ICP. Nevertheless, based on the average MSD, we are still able to find the cells containing the stones and identify such cell correspondence, i.e., $cell_2^I \rightarrow cell_{11}^{II}$. The meta-deformation of cell is rigid transformation of this stone as below:

$$[R|t] = \begin{bmatrix} -0.511 & -0.558 & -0.654 & -1980.472 \\ -0.217 & 0.820 & -0.530 & -527.690 \\ -0.831 & 0.129 & 0.540 & -1011.324 \end{bmatrix}.$$

It is worth noting that this solution is based on a local optimisation: we independently find out the most suitable corresponding cell for the given cell at Epoch I ($cell_i^I$) – from the cells at Epoch II ($cell_j^{II}$) , see Equation 6.2. As a complex deformation scenario, the independent local optimisation cannot guarantee we find the best cell correspondences for all cells at Epoch I. For example, both $cell_a^I$ and $cell_b^I$ at Epoch I are corresponding to $cell_j^{II}$ at Epoch II as

	1	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*	*13*	*14*	*15*	*16*	*17*	*18*
1	1.81	2.88	2.10	2.04	3.77	2.21	2.26	2.62	2.08	2.28	6.60	2.14	2.01	4.77	2.01	1.91	2.13	1.99
2	6.34	9.09	5.02	5.85	4.96	5.25	6.28	4.70	4.89	4.85	6.09	4.56	4.90	5.65	4.48	4.92	4.89	4.50
3	2.21	2.65	1.80	2.45	2.74	2.11	2.36	3.50	2.12	2.42	5.93	2.16	2.44	4.03	2.21	2.62	2.41	2.12
4	1.93	2.85	2.06	2.06	2.91	2.17	2.14	2.59	2.22	2.35	6.35	2.49	2.01	3.99	2.13	2.02	2.24	1.97
5	2.46	5.14	2.47	2.38	2.64	2.12	2.22	2.15	2.38	2.25	6.09	2.15	2.05	2.07	2.12	2.12	2.16	2.10
6	2.10	3.01	2.21	2.27	3.22	2.09	2.59	2.93	2.38	2.59	6.39	1.96	2.17	3.92	1.92	2.90	2.37	1.92
7	2.02	2.88	2.16	2.40	3.39	2.54	2.15	2.49	2.15	2.36	6.41	2.48	2.08	3.98	2.64	1.86	2.57	2.03
8	2.23	3.03	2.24	2.23	3.30	2.10	2.42	2.53	2.33	2.66	6.44	2.20	2.09	3.94	1.99	2.06	2.18	2.09
9	1.94	3.10	2.28	2.14	3.26	2.11	2.30	2.72	2.03	2.76	6.71	1.98	2.18	4.20	1.86	2.66	2.59	1.86
10	2.03	2.85	2.75	2.13	2.93	2.12	1.95	2.49	2.21	2.12	6.25	2.05	2.01	4.25	2.07	2.14	2.32	1.92
11	2.00	2.95	2.33	2.47	3.09	2.64	2.36	2.63	2.31	2.45	5.99	1.99	2.38	4.01	2.04	1.96	2.19	1.93
12	2.28	3.19	2.14	2.33	3.39	2.20	2.48	3.09	2.28	2.46	6.34	2.03	2.31	4.20	2.15	2.51	2.51	2.05
13	1.84	2.96	2.02	2.24	3.07	2.70	2.18	2.32	2.38	2.13	6.44	2.06	1.77	4.28	1.89	1.82	2.02	1.97
14	2.08	2.94	2.32	2.01	3.06	2.42	2.22	2.53	2.39	3.12	6.29	2.01	2.09	4.00	2.02	1.83	2.08	2.09
15	2.33	2.96	2.21	2.40	3.12	2.16	2.39	2.97	2.02	2.62	6.70	1.98	2.17	4.34	1.75	2.10	2.19	1.81
16	2.01	2.91	2.33	2.11	3.12	2.24	2.14	2.52	2.16	2.31	6.64	1.99	1.97	3.96	2.03	1.73	2.05	1.89
17	1.93	3.19	2.90	2.54	3.25	2.22	2.53	2.79	2.30	2.60	6.32	2.82	2.58	4.19	2.10	2.00	1.85	1.96
18	2.66	3.15	2.33	2.42	3.28	2.14	2.51	2.99	2.26	2.52	6.32	2.07	2.18	4.25	1.94	2.07	2.11	1.68

Table 6.2: Mean squared distance between two cells at Epoch 1 and Epoch 3 ($Index_{cell}^{1}$ vs $Index_{cell}^{3}$)

the most suitable cell based on local optimisation; however, there might be a constraint that $cell_a^I$ and $cell_b^I$ cannot be matched to one single cell at Epoch II (like the scenario in Figure 6.3). In addition, for most deformation scenarios, neighbouring cells at Epoch I should be also neighbourhood cells at Epoch II. These constraints request a global optimisation strategy for finding cell correspondence. In addition, real-life scenarios can be even more complex: e.g., (1) in contrast to the previous assumption, multiple cells at Epoch I can be deformed to a single cell at Epoch II; and vice versa, one single cell at Epoch I can be deformed into multiple cells at Epoch II; (2) neighbouring cells at Epoch I are not neighbouring any more after deformation at Epoch II. These complicated scenarios require more advanced cell-corresponding search methods rather than a single cell matching. In this thesis, we focus on finding cell correspondence using local optimisation and test it with a practical point cloud dataset, and global optimisation considering more complicated real-life constraints is the outlook of this thesis work.

6.4 Movement Tracking with Data-driven Cell

In Section 6.3, we have already enhanced the cell correspondence searching method, from direct correspondence (i.e., $cell_i^{(I)} \rightarrow cell_i^{(II)}$ with the same ID i) to the non-direct correspondence (i.e., $cell_i^{(I)} \rightarrow cell_j^{(II)}$ with different ID i and j). During this searching procedure, the space containing the object is still split into the same cell structure, i.e. the uniform cells with regular shape. This is a restriction of this method, based on an important assumption that the deformed parts are in a single cell (i.e., 0 cell movement), or an exact multiple of cells (i.e., N cells movement, where N is a natural number). However, this assumption is quite difficult to be satisfied in the real cases. Firstly, the deformed parts are usually not regular (non-rectangle, irregular shape of moving object, like an irregular stone); Secondly, the deformed parts may move into several different cells or many cells move to one cell, which increases the difficulty of deformed part detection. Thirdly, the movement distance of this deformed part may not exactly N times of the cell size, which might be smaller than a single cell (e.g., 0.3 cell size) or bigger than N cells but less than N+1 cells (e.g., 3.4 cell size). Therefore, we develop a *data-driven cell* to track object movement with deformation, in order to automatically detect the *boundary* and *position* of the deformed object; such data-driven cell also supports irregular cell shape.

In Section 3.4.4 about hybrid modelling of deformation, we have already introduced the "meta-cell" to better describe the boundary of the deformed object and to automatically detect the boundary of the deformed object with an irregular cell. The detailed procedure of movement tracking of an irregular object is summarised in Algorithm 6.3. The algorithm has the following three main steps:

Step 1. *Deformed parts (deformation boundary) identification* (line 2 - 7) – We firstly split the space into meta-cells (meta-cells here are much smaller than previous rectangle cells, and they are in analogy with pixels in image analysis). Secondly, we identify the meta-cells containing the points of the deformed parts (e.g., the stone in our previous practical experiment data collection) at the initial epoch. These meta-cells are grouped together as the deformed parts, which could be an arbitrary boundary (irregular shape like the rectangle in previous cells), see Figure 3.11 in Section 3.4.4.

Step 2. *Movement tracking* (line 8 - 10) – For point clouds at the subsequent epochs, we track the deformed parts and detect its position and boundary at each epoch continuously. There is an important sub-function, i.e., "*findCellBound*". This function is designed to identify the deformation boundary and to build an irregular cell (see Algorithm 6.4). In movement tracking, the previous-epoch deformed parts boundary is regarded as an irregular cell and used to find the boundary of the deformed part at current epoch. We firstly get the points in this irregular cell at current epoch. Afterwards, we move this cell along different direction via the unit of a meta-cell size, until finding the new boundary of the deformed part (i.e., the most suitable matching). In this step, we apply angle variation to determine the deformed parts at current epoch. Finally, we achieve the position and the boundary of the deformed parts at current epoch.

Step 3. *Deformation computation* (line 11 - 14) – After we obtain the position and boundary of deformed parts at each epoch, we compute the deformation parameters of these parts. The deformation parameter can be the rotation and translation of the moving mass. For the neighbouring epochs, we calculate the centres of two masses, connect them, and form a vector. Based on this vector, we can further calculate the rotation angle around X-, Y- and Z- axis and translation along these three axes as well.

The utilisation of "*meta-cell*" can improve the accuracy and efficiency of searching deformed part position. From the point of view of accuracy, the moving distance is not a multiple of "cell" sizes but a multiple of "meta-cell" size, when searching the correct position of the deformed parts. In general, meta-cell is much smaller than the cell size in previous sections; therefore, it can improve the accuracy of correspondence searching. From the point of view of efficiency, the

Algorithm 6.3: *MovementTracking*

```
Input: Multiple files of original point cloud datasets ($filename1, \cdots, filenameN$)
Output: Deformed part at all epochs ($CELL_i^1, \cdots, CELL_j^N$); Deformation parameters for
        deformed part at all epochs ($D_{1,2}, \cdots, D_{N-1,N}$)
begin
    /* load the points of the first file into $\mathbf{P}^1$ */
    $\mathbf{P}^1 = \{p_1^1, \cdots, p_N^1\} \leftarrow$ loadFile($filename1$);
    /* get the meta cells for the whole space at Epoch 1 */
    $metaCell \leftarrow$ getMetaCells($\mathbf{P}^1$);
    /* get the deformed part boundary at Epoch 1 */
    $CELL^1 = \{metaCell_{j,m}^1, \cdots, metaCell_{k,n}^1\} \leftarrow$ getDeformationBound($metaCell$);
    foreach $filenamei$ do
        /* get the deformation boundary $CELL_j^i$ at Epoch i*/
        $CELL_j^i \leftarrow$ findCellBound($filenamei, CELL_j^{i-1}$);
    /* get the deformed part centre at all epochs */
    $(centreMass^1, \cdots, centreMass^N) \leftarrow$ mean($CELL_i^1, \cdots, CELL_j^N$) ;
    /* calculate the deformation parameters based on the deformation boundary at all epochs */
    $(D_{1,2}, \cdots, D_{N-1,N}) \leftarrow$ deformation($centreMass^1, \cdots, centreMass^N$) ;
    return Deformed part at all epochs ($CELL_i^1, \cdots, CELL_j^N$); Deformation parameters for
    deformed part at all epochs ($D_{1,2}, \cdots, D_{N-1,N}$)
```

Algorithm 6.4: *findCellBound*

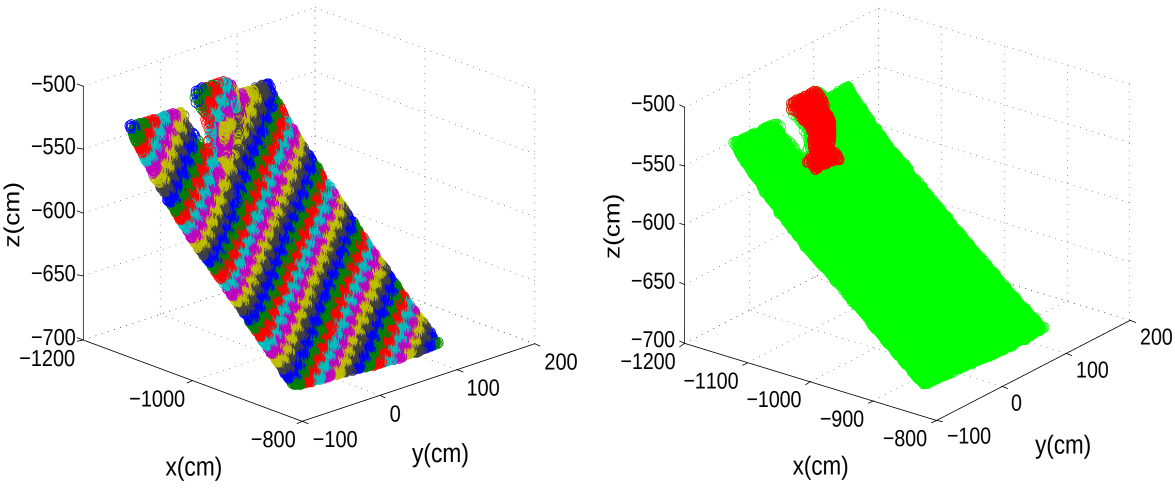
```
Input: A file of original point cloud datasets at Epoch i ($filenamei$), former deformed part
       boundary ($CELL_j^{i-1} = \{metaCell_{j,m}^{i-1}, \cdots, metaCell_{k,n}^{i-1}\}$)
Output: Deformed part boundary at Epoch i ($CELL^i$)
begin
    /* load the points of the file $filenamei$ into $\mathbf{P}^i$ */
    $\mathbf{P}^i = \{p_1^i, \cdots, p_N^i\} \leftarrow$ loadFile($filenamei$);
    /* get the points at Epoch i with the same bound as $CELL^{i-1}$ */
    $CELL_j^i = \{metaCell_{j,m}^i, \cdots, metaCell_{k,n}^i\} \leftarrow$ getFormerBound($CELL_j^{i-1}$);
    while $j < N_y$ & $angleVariation < threshold$ do
        /* build the $newCELL$ by moving $CELL_j^i$ on the space $\mathbf{P}^i$, with one meta-cell size as
        each moving distance*/
        $newCELL \leftarrow CELL^i + metaCell$;
        /* calculate the angle variation between $newCELL$ and $CELL^i$ */
        $angleVariation \leftarrow$ detectCELL($CELL^i, newCELL$);
        $CELL_j^i \leftarrow newCELL$;
    return find the cell boundary $CELL_j^i$ at Epoch i
```

object space does not need to be split again in different cell structure, when loading the points at the subsequent epoch. Alternatively, this space is just divided once into meta-cells at the initial epoch and is kept as same for the point cloud at the subsequent epoch. The position of the changed part is detected based on the meta-cell adjustment.

In addition, this data-driven cell is able to support irregular cell shape, not only the regular rectangle/cube cells in previous sections. Furthermore, this strategy can utilise other advanced properties of the point clouds besides the spatial positions of the points, like the roughness of the slope, RGB colour information of the points, etc. For instance, angle variation, as a graphic feature, is used to decide the correct position of the cell.

Concrete Experiments of Movement Tracking – In Section 6.3, we have already presented the experimental results of finding cell correspondence using the practical experiment datasets at two epochs. In fact, we have four point cloud datasets collected by this practical experiment. Now, we want to utilise this four-epoch datasets to test the movement tracking algorithm using the data-driven cell. In this scenario, there is a stone moving on the slope. The main objective of this experiment is to track the movement of this stone on four-epoch point clouds. This process consists of three crucial steps, i.e., *deformed parts identification*, *movement tracking* and *deformation computation*. The detailed experimental results are presented as follows.

Deformed parts identification. This is the initialisation procedure to detect the position and boundary of the tracking object from the point cloud at Epoch 1. Figure 6.14(a) shows the point cloud dataset at Epoch 1 with space split into meta-cells. Therefore, we need to determine which meta-cells contain the points of the tracking object (e.g., the stone with irregular shape, not pure cube or rectangle), based on its features like geographic property, colour information, etc. Here, according to the feature variation in the bound, we are able to find the irregular cell (i.e., a group of meta-cells) that contains this stone. From Figure 6.14(b), we can observe that the green part of this point cloud is the slope at Epoch 1; while the red part is the moving stone at Epoch 1.

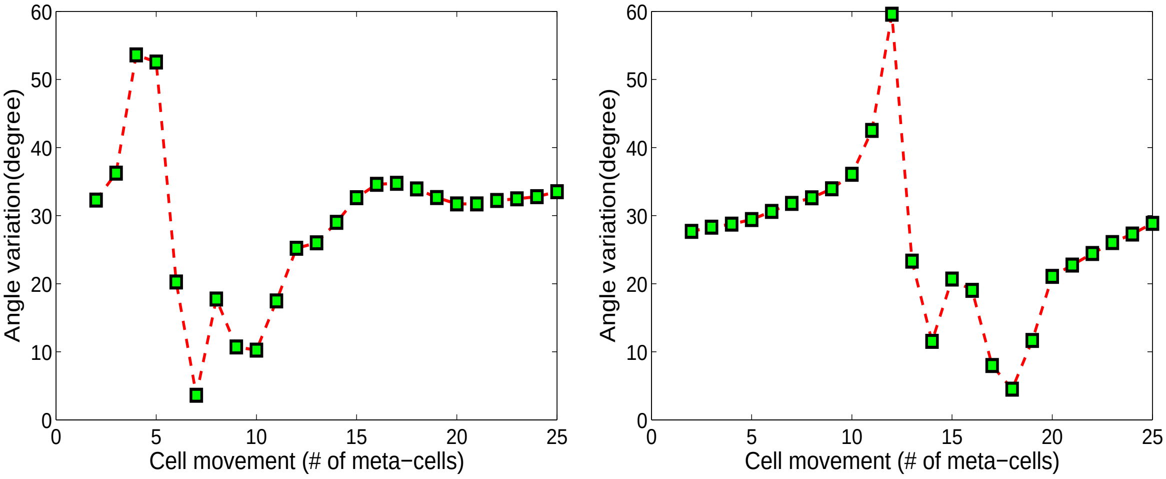


(a) Meta cells of point cloud at Epoch 1 (b) Deformed parts detected at Epoch 1

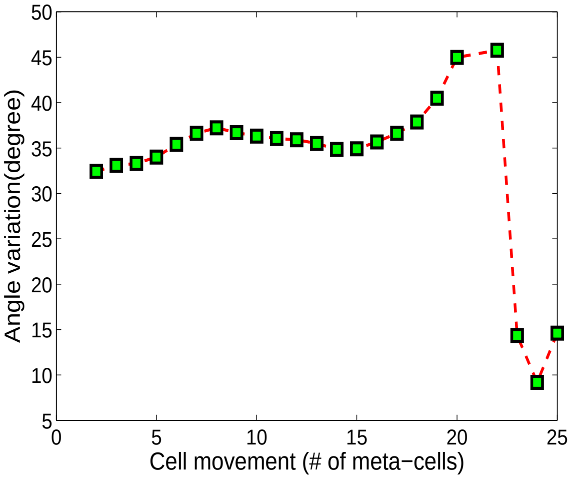
Figure 6.14: Object detection at Epoch 1

Movement tracking. After achieving the initial position of this stone (as an irregular cell) at Epoch 1, we start to determine its position at subsequent epochs using the point clouds. In general, the position of this stone (i.e., the initial irregular cell) at Epoch K provides a starting point to search this stone at Epoch K+1; the algorithm moves the cell step by step, until reaching the best matching between two corresponding cells. The step unit is the size of meta-cell. Considering the rock fall scenario, the stone usually moves up-down along the slope. In this experiment, we just move the cell along the down direction (e.g., $Y-$axis), not along the horizontal direction (e.g., $X-$axis).

We apply the *angle variation* metric to decide whether the object is matched or not. Angle



(a) Detection at Epoch 2 (b) Detection at Epoch 3



(c) Detection at Epoch 4

Figure 6.15: Object detection using angle variation

variation is the difference between the normals of two point sets. One set is the point cloud found at the previous epoch; another one is the points inside the cell at current epoch. Figure 6.15 shows the angle variation plots, when detecting the irregular cell from the point cloud at Epoch to the point cloud at Epoch 2, 3 and 4, respectively. We can observe that, (1) in general, the three sub-figures have large angle variations - which means the cells are not correspondent each other; (2) with the cell movement (i.e., moving along how many meta-cells), there is a minimum valueof angle variation, which indicates that the two cells are correspondent. Figure 6.15-(a) indicates that from Epoch 1 to Epoch 2 it requires about 7 steps, Figure 6.15-(b) indicates that from Epoch 1 to Epoch 3 it requires about 18 steps, and Figure 6.15-(c) indicates that from Epoch 1 to Epoch 4 it requires about 24 steps. In addition, Figure 6.16 provides the detailed plot of these four cells at the four epochs, respectively. In other words, these four cells are the position of the stone at each individual epoch – this is the detailed information of tracking such stone movement.

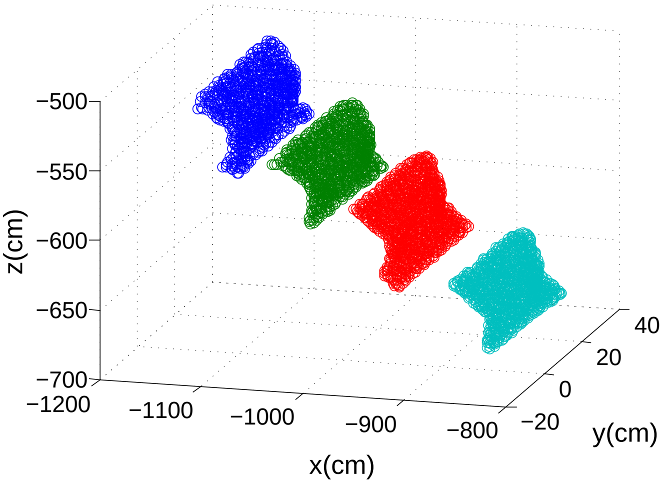
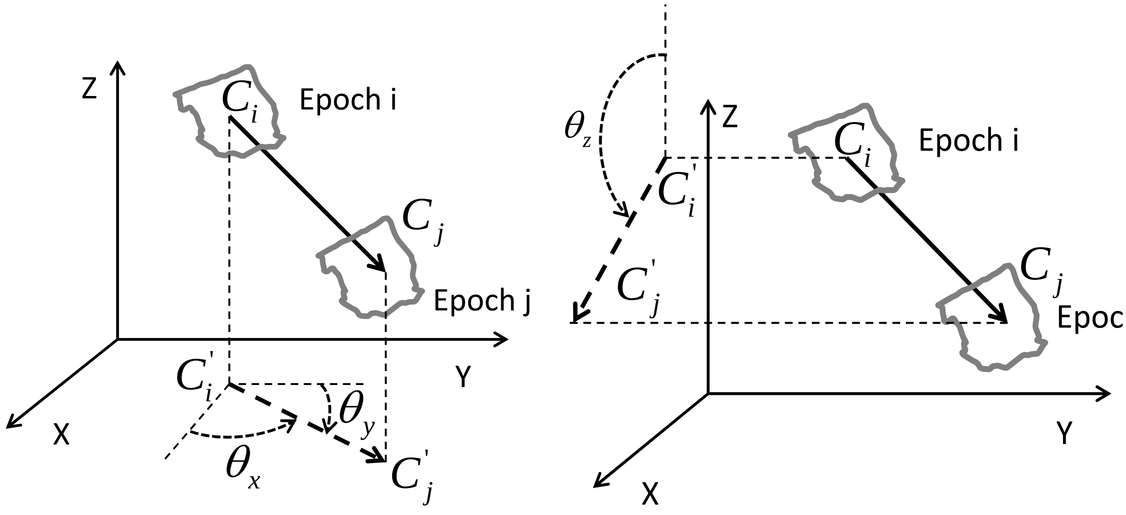


Figure 6.16: Object movements from Epoch 1 to Epoch 4 (only show the moving object)

Deformation computation. After tracking the stone from Epoch 1 to Epoch 4, we can calculate the detailed deformation parameters of each movement. As shown in Figure 6.17, C_i and C_j are the centre of deformed parts (i.e., irregular cell) at Epoch i and Epoch j, respectively. Vector $\overrightarrow{C_iC_j}$ is the deformation vector of this mass between Epoch i and Epoch j. In order to better compare the rotation angle in the same reference plane, we project the $\overrightarrow{C_iC_j}$ vector into the xy plane as $\overrightarrow{C_i'C_j'}$ (see Figure 6.17(a)); and also project the $\overrightarrow{C_iC_j}$ vector into the xz plane as $\overrightarrow{C_i'C_j'}$ (see Figure 6.17(b)). Therefore, θ_x, θ_y and θ_z are the rotation angle of vector $\overrightarrow{C_iC_j}$ rotating around the X-, Y-, Z- axis, respectively. We can utilise these three angle θ_x, θ_y and θ_z to describe the rotation of the $\overrightarrow{C_iC_j}$ vector from Epoch i to Epoch j.



(a) Rotation around x- and y- axis

(b) Rotation around z- axis

Figure 6.17: Rotation of deformation vector around the X-, Y- and Z- axis between Epoch i and Epoch j

Table 6.3 shows the rotation angles of this stone (i.e., the irregular cell) around the X-, Y- and Z- axis from Epoch 1 to Epoch 4. We can see that the rotation of the stone around the

X- and *Y*- axis changed slightly (less than 0.3°); and the rotation around the *Z*- axis changed larger (about 2°). In addition, Table 6.4 shows the translation of this stone along the *X*-, *Y*- and *Z*- axis from Epoch 1 to Epoch 4. It had a big movement along the *X*- and *Z*- axis and a small movement (less than 1 *cm*) along the *Y*- axis.

$\theta(\circ)$	$E1 \rightarrow E2$	$E2 \rightarrow E3$	$E3 \rightarrow E4$	$E1E2 \rightarrow E2E3$	$E2E3 \rightarrow E3E4$
θ_x	0.0498	0.2463	0.4679	-0.1965	0.2216
θ_y	89.9502	89.7537	89.5321	-0.1965	-0.2216
θ_z	115.5162	116.2956	114.2666	0.7794	-2.02900

Table 6.3: Rotation angle of deformation vector around the *X*-, *Y*- and *Z*- axis between two neighbouring epochs (from Epoch *1* to Epoch *4*)

$\Delta(cm)$	$E1 \rightarrow E2$	$E2 \rightarrow E3$	$E3 \rightarrow E4$	$E1E2 \rightarrow E2E3$	$E2E3 \rightarrow E3E4$
Δ_x	76.2332	77.4393	96.8693	1.2061	19.43
Δ_y	-0.0662	-0.3329	-0.7912	-0.2667	-0.4583
Δ_z	-36.3879	-38.2655	-43.6702	-1.8776	-5.4047

Table 6.4: Translation of the stone (i.e., irregular cell) along the *X*-, *Y*- and *Z*- axis between two neighbouring epochs (from Epoch *1* to Epoch *4*)

6.5 Adaptive Model

In addition to the extension of using dynamic cells and building cell correspondence for the split step in our cell-based deformation monitoring, we attempt to improve the step of "detect" for achieving better meta-deformation. We have presented several meta-deformation estimation methods in Section 4.4. However, after choosing a model for detecting meta-deformation, this model is fixed and used for the whole "detect" step for all cells. There is no way to switch to another model during the detection step. In this section, we develop *adaptive model* as a flexible choice for the detection of meta-deformation, i.e., different cells can independently choose different meta-model that is mostly suitable for this cell.

The main objective of building "adaptive model" is able to provide multi-model choice when detecting meta-deformation, which means that we can select the best one from several model candidates for each individual cell. This strategy can provide an on-the-fly way to choose the most appropriate model in order to improve the detection quality of the meta-deformation for each cell. The detailed procedure of applying adaptive model for the detection of cell meta-deformation is as follows:

Step 1. *Initialisation* - identify some candidate models for building meta-deformation (e.g., a simple linear model $\mathcal{M}_1$ – see Equation 6.4, a polynomial regression model $\mathcal{M}_2$ – see Equation 6.5), and define the error threshold (δ) to determine how to choose from the candidate models and find the most suitable one;

Step 2. *Test with a simple model* - apply a model (e.g., a simple linear regression model) for the points inside the cell and extract the modelling error (e.g., the mean squared distance between the fit surface and the real points).

Step 3. *Check whether a simple model is good enough* - compare the modelling error with the defined error threshold δ. If error for one of the epoch is larger than δ, then we select an alternative better model (e.g., a polynomial regression model) to model this cell.

Step 4. *Iteration for all cells* - Step 2 and Step 3 are iterated until finding the suitable model (between $\mathcal{M}_1$ and $\mathcal{M}_2$) for all cells. It is worth noting that we can provide more alternative models ($\mathcal{M}_3 \cdots$) in real-life experiments.

$$z = m_1^{(1)} x + m_2^{(1)} y + m_3^{(1)} \tag{6.4}$$

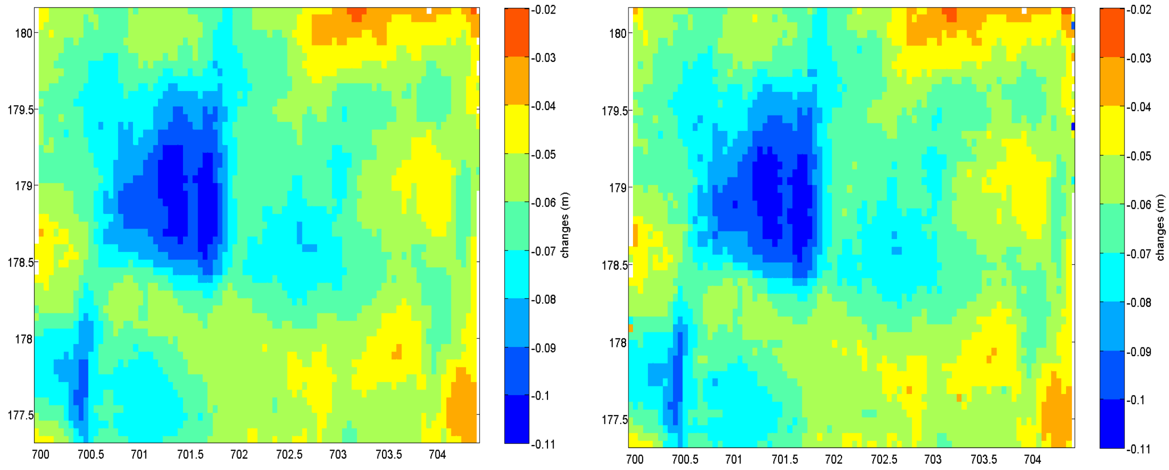
$$z = m_1^{(2)} x^2 + m_2^{(2)} xy + m_3^{(2)} x + m_4^{(2)} y^2 + m_5^{(2)} y + m_6^{(2)} \tag{6.5}$$

Concrete Experiments of using Adaptive Model – Now, we test our "adaptive model" method on real-life point cloud datasets, i.e., the erosion case study in Section 5.4. To simplify the experiment, we build adaptive modelling using two models - $\mathcal{M}_1$ and $\mathcal{M}_2$. $\mathcal{M}_1$ is a plane model expressed by Equation 6.4; and $\mathcal{M}_2$ is a second-order polynomial model written as Equation 6.5. $\mathcal{M}_1$ is a simpler model with less complexity compared to $\mathcal{M}_2$. Therefore, the selection strategy in our adaptive modelling is to use a simpler model ($\mathcal{M}_1$) first if the error of simpler model can satisfy the requirement, otherwise we apply a stronger model ($\mathcal{M}_2$) for the cell. The requirement is: whether the mean squared distance between the approximation using the fitted surface and the real points in the cell is smaller than a given error threshold δ. In our experiment, we study the sensitivity of choosing different δ values, e.g., 0.004, 0.003, 0.002, 0.0012, 0.001, 0.0008, 0.0006 (m), as shown in Table 6.5.

Error threshold	0.004	0.003	0.002	0.0012	0.001	0.0008	0.0006
$\#_{Model1^{Cell}}$	5058	5057	4917	2454	1249	219	2
$\#_{Model2^{Cell}}$	0	1	141	2604	3809	4839	5056
Total error of all cells (Epoch 1)	4.6876	4.6873	4.6744	4.4653	4.3759	4.3142	4.3065
Total error of all cells (Epoch 2)	5.7634	5.7630	5.7216	5.1994	4.9673	4.7880	4.7534
Total error of all cells (Epoch 1+Epoch 2)	10.4510	10.4503	10.3960	9.6647	9.3432	9.1022	9.0599

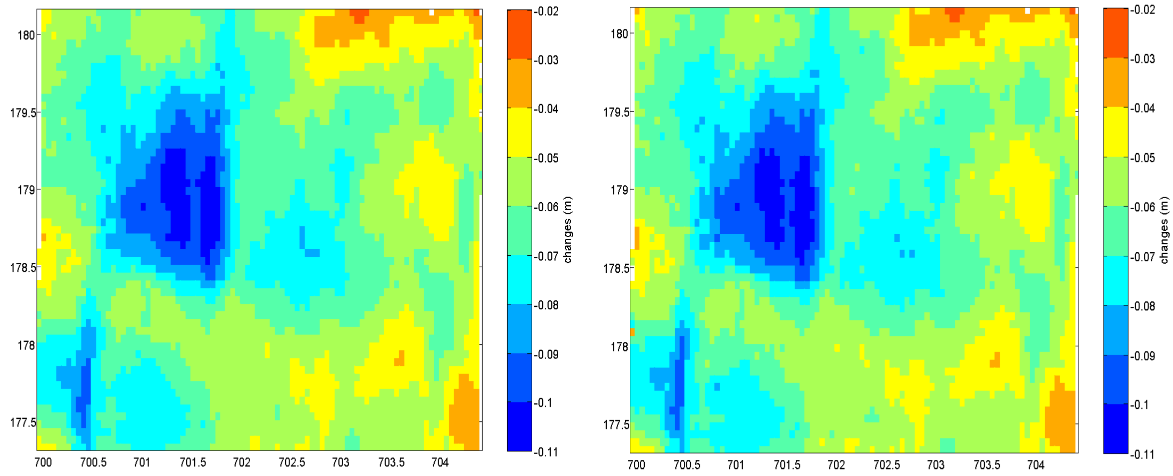
Table 6.5: Statistical analysis of deformation map estimated by adaptive model (the unit in the table is *meter*, *cellsize* : 0.05 m)

In this table, we provide detailed statistical analysis of deformation map that are computed by the adaptive model selection. As shown in the second row ($\#_{Model1^{Cell}}$ – the number of cells that choose $\mathcal{M}_1$) and the third row ($\#_{Model2^{Cell}}$ – the number of cells that choose $\mathcal{M}_2$), with the decrease of error threshold δ, there are more cells using $\mathcal{M}_2$ to estimate meta-deformation; and vice versa, many cells go for $\mathcal{M}_1$. Since the estimation model is used for points in cells at both Epoch I and Epoch II, we sum up the total error of all the cells at two epochs. We compute three kinds of errors, i.e., (1) the total error of all the cells at Epoch 1, (2) the total error of all the cells at Epoch 2, and (3) the sum of total errors both at Epoch 1 and Epoch 2; these three errors are shown in the last three rows in Table 6.5. We observe that all errors decrease with the reduction of error threshold δ. In addition, we compute the total errors of all cells at Epoch I and Epoch II using a static model: for using $\mathcal{M}_1$, the errors at Epoch I and Epoch II are 4.6876 m and 5.7634 m, respectively (the sum is 10.4510 m); for using $\mathcal{M}_2$, the errors at Epoch I and Epoch II are 4.3065 m and 4.7530 m, respectively (the sum is 9.0595 m). Obviously, the total error is: $error(\mathcal{M}_2) < error(adaptive\ model) < error(\mathcal{M}_1)$, because that a higher order model can fit the surface more accurately and has a lower fitting residuals. The possible question here is that why we do not choose $\mathcal{M}_2$ for all cells as the total modelling error could be less: the reason is that building $\mathcal{M}_2$ is more time-consuming than building $\mathcal{M}_2$ (see Equation 6.4 and Equation 6.5). For all cells, the good strategy is to choose the most suitable models rather than the best fitting model. The choice of mode needs to balance the accuracy and the computation



(a) Model 1: linear

(b) Model 2: 2nd order polynomial



(c) Adaptive model ($\delta = 0.002$)

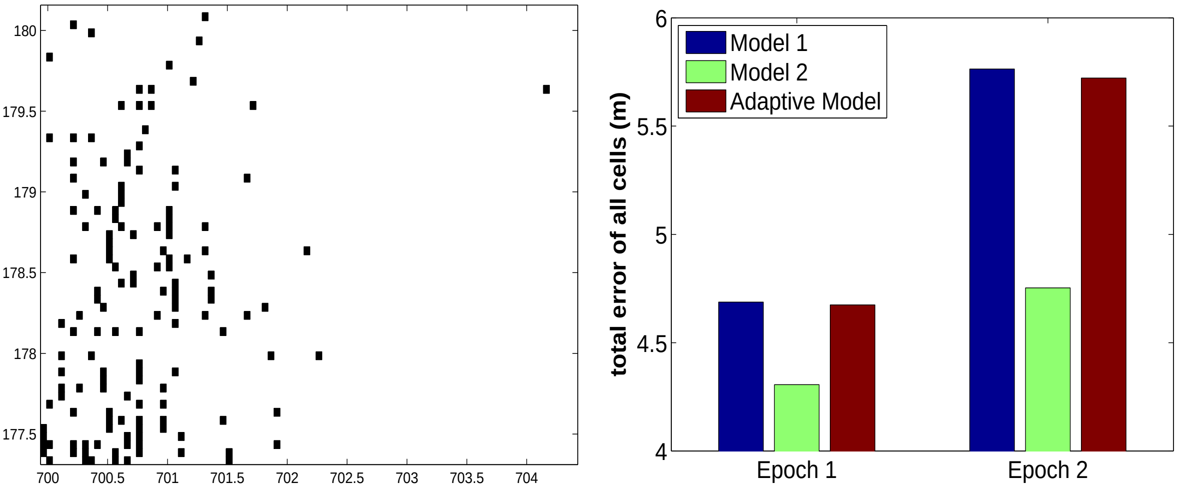
(d) Adaptive model ($\delta = 0.0012$)

Figure 6.18: Deformation map between Epoch 1 and 2 estimated by a uniform model (either linear or order-2 polynomial) and adaptive models. ($cellsize : 5\ cm$)

cost, and identify the most suitable one for each individual cell.

To further show the illustrative results, we provide the final deformation maps using different model strategies. Figure 6.18(a) shows the four deformation maps, which are individually computed (a) using fixed $\mathcal{M}_1$, (b) using fixed $\mathcal{M}_2$, (c) using adaptive model with threshold ($\delta = 0.002\ m$), and (d) using adaptive model with threshold ($\delta = 0.0012\ m$), respectively. Based on these different regression models, we estimate the centre of point clouds in each cell, compute the deformation in terms of centre displacement at two epochs, and get the final deformation map. In general, the deformation map produced by these different meta-deformation modelling strategy are quite similar (as shown in Table 6.5, the errors of using $\mathcal{M}_1$ and $\mathcal{M}_2$ are in a small range between [9.0-10.5] with small difference); the reason could be linear and order-2 polynomial did not provide much difference for this datasets. Nevertheless, we could observe that (1) the results of $\mathcal{M}_1$ is smoother compared to $\mathcal{M}_2$; on the other word, $\mathcal{M}_2$ has more detailed and different results in some parts; (2) the adaptive model is somehow between $\mathcal{M}_1$ and $\mathcal{M}_2$; (3) by using larger threshold δ, the adaptive model becomes closer to $\mathcal{M}_1$.

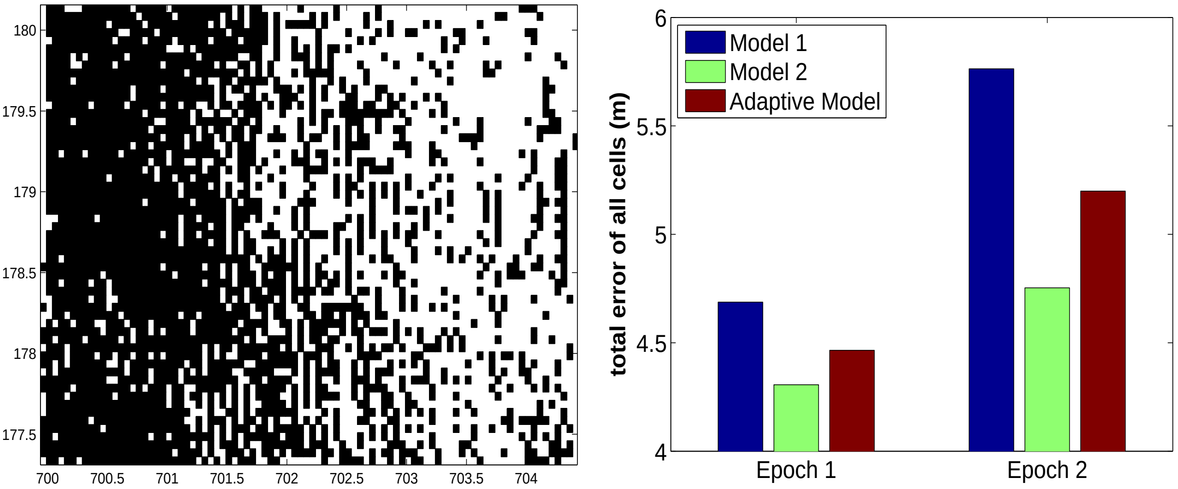
In order to further identify the detailed selection between the two models ($\mathcal{M}_1$ and $\mathcal{M}_2$) for each cell using the adaptive modelling strategy, Figure 6.19(a) provides an illustrative result on the deformation map to highlight which cells using which model. In the figure, the black rectangles indicate that the cell used the complex model ($\mathcal{M}_2$), while the write ones indicate using the simple model ($\mathcal{M}_1$) for modelling the points in the cell. There are 141 cells (2.79%) using $\mathcal{M}_2$ and the rest choosing $\mathcal{M}_1$. We observe that most of $\mathcal{M}_2$ selection is on the left side of the object, where there are some larger deformations/subsidence. In addition, Figure 6.19(b) shows the total error of cells with adaptive model at Epoch I and Epoch II, compared to using a fix model, either $\mathcal{M}_1$ or $\mathcal{M}_2$.



(a) Adaptive model index (□: model 1; ■: model 2) (b) Error statistics of deformation map estimated by adaptive model

Figure 6.19: Statistical analysis of deformation map between Epoch 1 and 2 estimated by an adaptive model ($\delta = 0.002\ m,\ cellsize = 5\ cm$)

We have the similar plots that are using adaptive model using smaller threshold ($\delta = 0.0012\ m$). We observe that there are more cells using $\mathcal{M}_2$, as shown in Figure 6.20(a), where



(a) Adaptive model index (□: model 1; ■: model 2) (b) Error statistics of deformation map estimated by adaptive model

Figure 6.20: Statistical analysis of deformation map between Epoch 1 and 2 estimated by an adaptive model ($\delta = 0.0012\ m,\ cellsize = 5\ cm$)

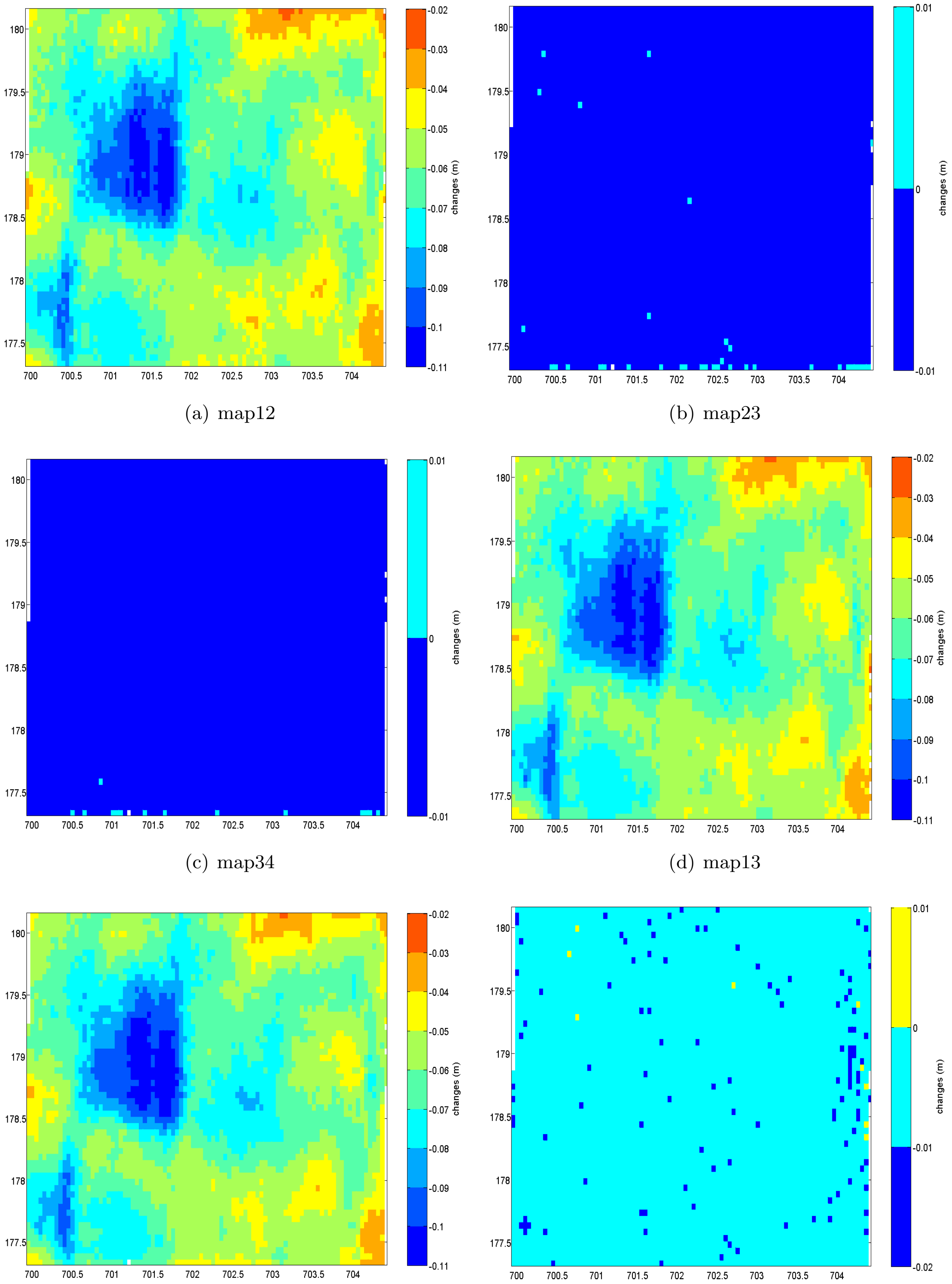
2604 (51.48%) cells choose $\mathcal{M}_2$, which is significantly larger than 141 (2.79%) cells when adaptive model threshold is $\delta = 0.002\ m$. Similarly, we plot the total modelling errors for the two epochs, and observe that the adaptive model error becomes closer to $\mathcal{M}_2$.

In summary, the strategy of adaptive modelling can automatically select the most suitable model for detecting meta-deformation for each individual cell. A model can be complex (with high accuracy) or simple (with low accuracy). To effectively analyse the tradeoff between modelling accuracy and the model complexity (i.e., the time cost), adaptive modelling aims at providing the best model for each individual cell dynamically. It is worth noting that like many parametric setting tasks, the choice of threshold ("δ") is also quite important in adaptive modelling. This parameter also depends on application knowledge and needs to be tuned accordingly.

6.6 Evolution of Deformation Map

As the final extension of cell-based deformation monitoring via 3D point clouds, this section focuses on detecting the deformation evolution, which means a long-term deformation analysis based on several different times of point cloud measurements (i.e., at least more than two epochs). In Section 3.4 as well as other relevant sections, we present the concept of "deformation map" for describing the deformation between two epochs. This deformation map is computed from two-epoch point cloud datasets. This section studies the evolution of deformation map, which can be considered as the evolution of deformation for a long term deformation monitoring.

In Section 5.4, we have already discussed a real-life deformation scenario (i.e., erosion on a concrete slope) with four epochs of LiDAR point cloud datasets. More detailed description of such point cloud datasets can be referred to Section 5.4. To identify the evaluation of deformation map, we can recursively compute the deformation map between two closest measurements, i.e.,



(a) map12

(b) map23

(c) map34

(d) map13

(e) map14

(f) map24

Figure 6.21: Deformation map between Epochs 1, 2, 3 and 4 with cell size 5 *cm*

the deformation maps of $\{Epoch1 \rightarrow Epoch2\}$, $\{Epoch2 \rightarrow Epoch3\}$, $\{Epoch3 \rightarrow Epoch4\}$. To simplify the notation, we call them $map12$, $map23$, and $map34$. By using cell size as 5 cm and the meta-model of "*using minimal Z to approximate vertical displacement for each cell*" (see Section 4.4 about detecting meta-deformation for more details), we compute the evolution of deformation maps (see Figure 6.21(a), Figure 6.21(b), Figure 6.21(c), corresponding to $map12$, $map23$, and $map34$, respectively). We observe that most of the changes are between Epoch I and Epoch II, i.e., $map12$; whilst, $map23$ and $map34$ are relatively stable without any significant deformation, and particularly $map34$ almost has no deformation in the last period between Epoch 3 and Epoch 4.

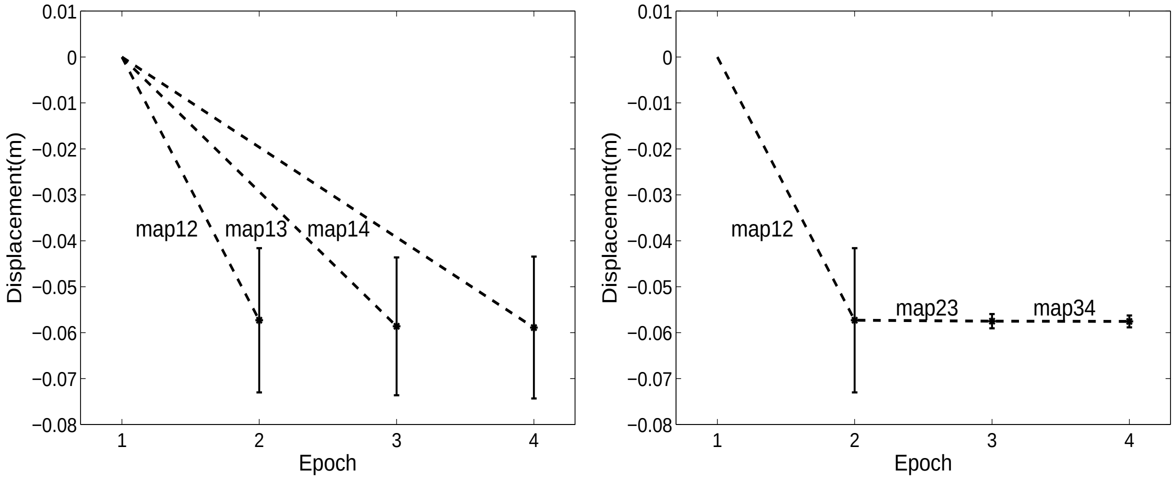
To further check this, we additionally compute the deformation between several pairs of epochs such as $\{Epoch1 \rightarrow Epoch3\}$, $\{Epoch1 \rightarrow Epoch4\}$, $\{Epoch2 \rightarrow Epoch4\}$ (see Figure 6.21(d) for $map13$, Figure 6.21(e) for $map14$, and Figure 6.21(f) for $map24$). We identified the same observations: there are some significant changes in $\{Epoch1 \rightarrow Epoch3\}$ and $\{Epoch1 \rightarrow Epoch4\}$, but very slight deformation after Epoch 2 (i.e., $\{Epoch2 \rightarrow Epoch4\}$). From Figure 6.21(d) and Figure 6.21(e), we can see that the deformation map keeps similar to Figure 6.21(a): the map_{12}, map_{13}, and map_{14} are similar as the main deformation is between Epoch 1 and Epoch 2. Figure 6.21(b) (i.e., map_{23}), Figure 6.21(f) (i.e., map_{24}) and Figure 6.21(c) (i.e., map_{34}) further prove that the erosion of this region between Epoch 2 and Epoch 4 is quite small.

In addition, we compute statistical characteristics (e.g., mean and standard deviation) of the six deformation maps shown in Figure 6.21. Table 6.6 summarises these statistical measurements. For the three continuous deformation maps ($map12$, $map23$, $map34$), the average deformation between the two neighbouring epochs decreases from 0.0573 m to 0.0002 m, until 0 m. The standard deviation also decreases with the time proceeds. Additionally, we also have relevant consistent statistical measurements for the other three deformation maps, i.e., $map13$, $map14$ and $map24$.

Deformation map	map12	map23	map34	map13	map14	map24
Mean	-0.0573	-0.0002	-0.0000	-0.0586	-0.0589	-0.0005
Standard deviation	0.0157	0.0015	0.0013	0.0150	0.0154	0.0023

Table 6.6: Statistical results of deformation map with cell size 5 cm ($unit : m$)

To provide a further illustrative view on deformation evolution, we provide a new plot showing the evolution of the erosion in the deformation map in Figure 6.22. Sub-figure 6.22(a) shows the calculative deformation from Epoch 1 to three other epochs (Epoch 2, Epoch 3 and Epoch 4); and sub-figure 6.22(b) represents the direct deformation between two neighbouring epochs like $\{Epoch1 \rightarrow Epoch2\}$, $\{Epoch2 \rightarrow Epoch3\}$, $\{Epoch3 \rightarrow Epoch4\}$, which changed quickly between the first two epochs and then kept almost stable in the following two periods.



(a) Direct computation (b) Accumulated deformation

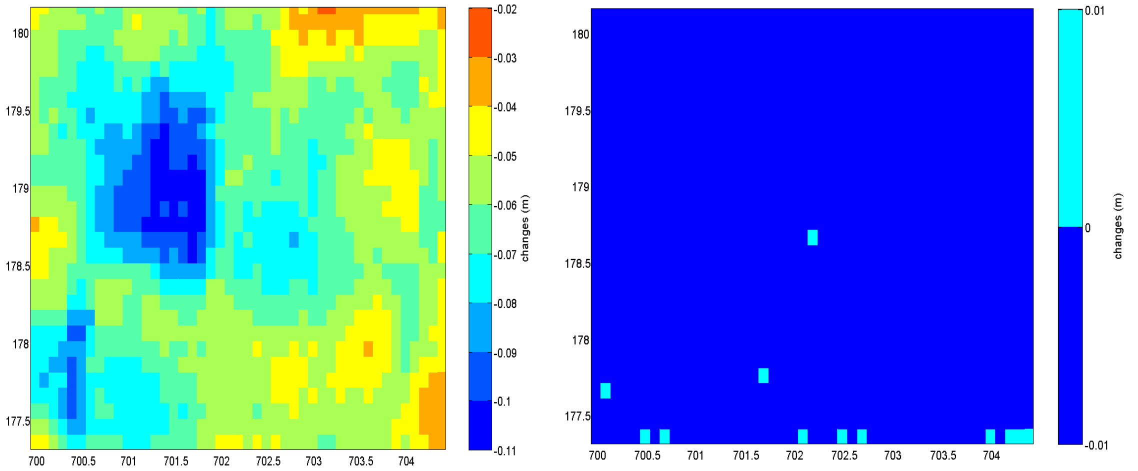
Figure 6.22: Deformation map between multi-epochs with cell size 5 *cm*

In addition to compute the deformation evolution using cell size of 5 *cm*, we additionally establish another set of experiments that use the cell size of 10 *cm*. Figure 6.23 shows the six deformation maps (i.e., *map*12, *map*23, *map*34, *map*13, *map*14, *map*24) between different epochs when applying cell size of 10 *cm*. We observe the similar results with deformation map using the cell size of 5 *cm* in Figure 6.21: a significant deformation happened between Epoch 1 and Epoch 2. After Epoch 2, there is only slight changes, e.g., *map*23 and *map*24 comparing Epoch 2 with Epoch 3 and Epoch 4, respectively.

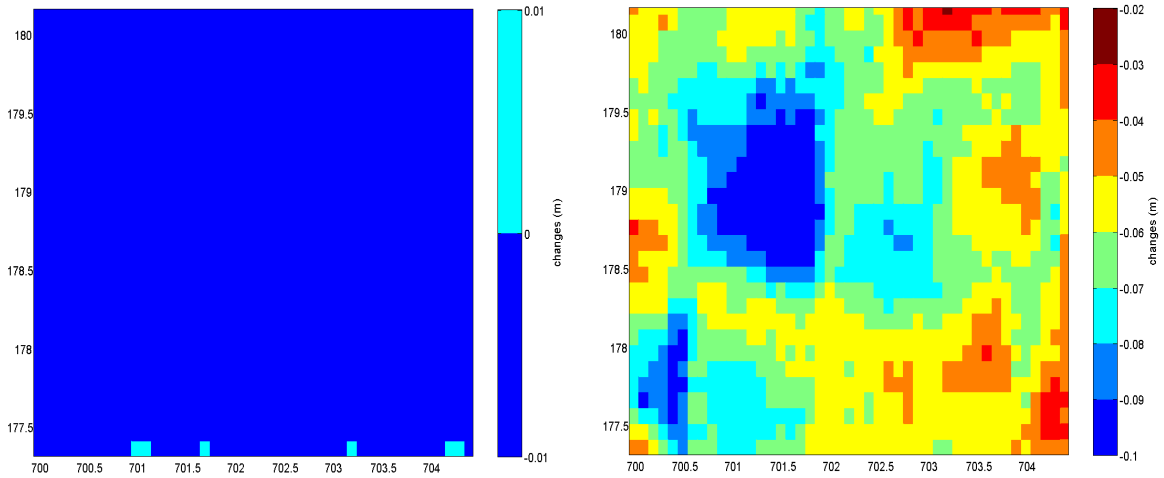
Similar to the experiments in building deformation evolution using cell size of 5 *cm*, we also compute the statistical features (e.g., mean and standard deviation) for each deformation map between any two epochs. Table 6.7 summarises this information. We observe the consistent results with Table 6.6, where the deformation between continuous epochs decreases from 0.0573 *m* (at *map*12), to 0.0002 *m* (at *map*23), and to 0 *m* (at *map*34) finally. In addition, we illustratively show the evolution of the erosion in the deformation map in Figure 6.24. Sub-figure 6.24(a) shows the calculative deformation from Epoch 1 to three other epochs (Epoch 2, Epoch 3 and Epoch 4); and sub-figure 6.24(b) represents the direct deformation between the neighbouring epochs like $\{Epoch1 \rightarrow Epoch2\}$, $\{Epoch2 \rightarrow Epoch3\}$, $\{Epoch3 \rightarrow Epoch4\}$, which changed quickly between the first two epochs and then kept almost stable in the following two periods. The plots in general are quite consistent with Figure 6.22 using cell size of 5 *cm*.

Deformation map	map12	map23	map34	map13	map14	map24
Mean	-0.0583	-0.0002	-0.0000	-0.0592	-0.0596	-0.0004
Standard deviation	0.0147	0.0016	0.0014	0.0145	0.0145	0.0022

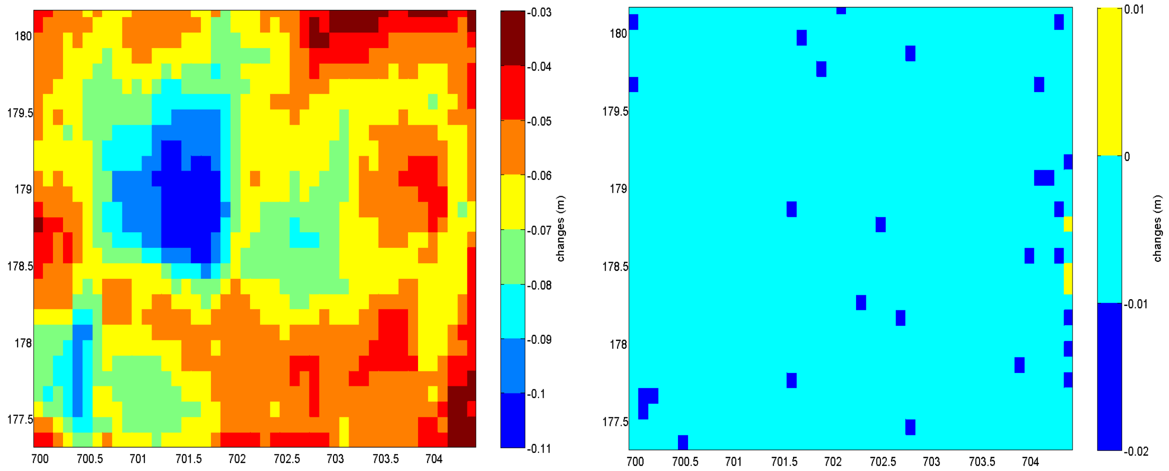
Table 6.7: Statistical results of deformation map with cell size 10 *cm* $(unit : m)$



(a) map12 (b) map23

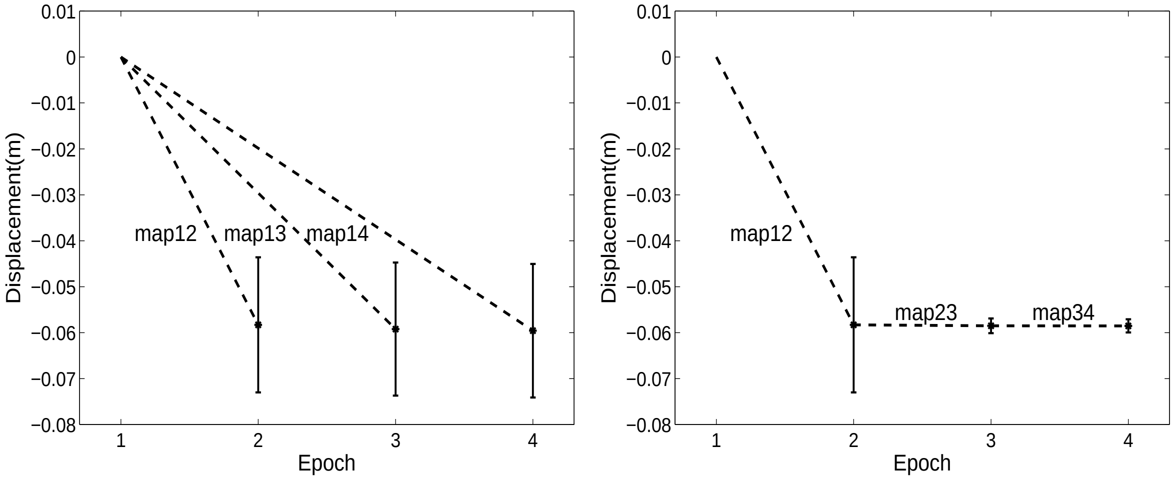


(c) map34 (d) map13



(e) map14 (f) map24

Figure 6.23: Deformation map between Epochs 1, 2, 3 and 4 with cell size 10 *cm*



(a) Direct computation (b) Accumulated deformation

Figure 6.24: Deformation map between multi-epochs with cell size 10 *cm*

6.7 Summary

This chapter provided five main extensions of our cell-based deformation computation approach, such as designing *dynamic cells*, finding *cell correspondence*, tracking *deformed parts with data-driven cell*, movement tracking with *data-driven cell*, building *adaptive model* and computing *evolution of deformation map.*

Dynamic cells is one kind of improvement from the "split" perspective, and offers a more suitable way to automatically split the space into cells with different sizes. We designed a top-down split strategy that is able to build dynamic cells on the fly, and it can take advantage of geometric features of the monitored object. Finding cell correspondence is a very challenging task, particularly in the situation where some deformation causes the data points moving from one cell to another cell. We designed a local optimisation strategy that is able to independently find the most suitable corresponding cell at Epoch II for a given cell at Epoch I. Tracking deformed parts with data-driven cell is a further improvement of "dynamic cells" and "finding cell correspondence", which can support arbitrary cell boundary and non-direct correspondence searching. We bring the *meta-cell* into this extension and propose an algorithm to track object movement on a slope using irregular cell. From the modelling perspective to build meta-deformation for cells, we additionally designed adaptive modelling strategy that can adaptively choose suitable model for each cell. Our method can start from simpler model (like 1st-order linear regression) to complex ones (e.g., 2nd-order polynomial model). This can provide a good tradeoff between modelling accuracy and model complexity, and select the most suitable model to calculate meta-deformation. The final extension is computing the evolution of deformation map that could build a long-term deformation monitoring based on several epochs of point clouds. In addition, such deformation evolution might be useful for further assessment and prediction of future deformation trend.

Chapter 7

Conclusion and Outlook

Climate change will mean more landslides.

Almost 100 experts from 14 nations, 2006

7.1 Conclusion

Deformation phenomena like landslides are one of the most crucial natural hazards that bring significant damages to Earth and human life, e.g., (1) the Rio de Janeiro landslide in Brazil on January 11, 2011, which caused 610 deaths[1], and (2) the Gansu mudslide in China on August 8, 2010, which led to 1, 471 deaths[2]. In Switzerland, landslides have resulted in enormous financial damages and mainly affected pre- and central Alpine regions [HBH09]. Landslide disasters (like ground movement, rockfall, and slope failures) along highway or railway lines become a serious issue to public transportations in people's daily life [DR08, KYS08].

Monitoring and possibly forecasting such deformation scenarios has become a tremendous important research and engineering topic in the environmental field. Traditionally, deformation monitoring techniques were mainly based on a small set of sparse monitoring points using the conventional geodetic surveying techniques; only using a few points, such geodetic techniques could hardly provide a prompt and informative deformation analysis. Recently, the increasing techniques of terrestrial LiDAR can capture a large amount of high-density three-dimensional point clouds easily and efficiently, both from the perspective of technical feasibility and financial cost. To analyse deformation using such terrestrial LiDAR equipment, this thesis studied the problem of "*Deformation Monitoring via 3D Point Clouds*". The fundamental contribution here is a novel hybrid model as well as a cell-based deformation computation approach that is able to generate rich and informative deformation automatically and incrementally. We now summarise the main research findings and technical contributions from this thesis study.

---

[1] http://en.wikipedia.org/wiki/January_2011_Rio_de_Janeiro_floods_and_mudslides

[2] http://en.wikipedia.org/wiki/2010_Gansu_mudslide

Deformation Modelling Requirements. In this thesis, we first found out the complete modelling requirements for representing deformation, based on the investigation of several different deformation scenarios such as landslide, subsidence, rockfall, and erosion. We identified that a comprehensive deformation modelling approach should cover three main components, i.e., "what", "which", and "how".

- *What*: This component is concerned with the deformation object, i.e., the monitored object under the measurement using relevant terrestrial LiDAR equipments. We divide the objects into two categories, i.e., (1) *artificial objects* like buildings, structures, bridges, dams that are built by human, and (2) *natural objects* like mountains, slopes, valleys that typically exist in Earth environment by nature. In many studies, both *artificial objects* and *natural objects* can have similar deformation analysis methods, and the difference between them is not extremely important.
- *Which*: This component focuses on the deformation types, e.g., rigid or non-rigid deformation. Rigid deformation is used for describing the absolute change of a monitored object from the complete object point of view. Particularly, "displacement" is largely used when modelling the rigid deformation, such as vertical displacement along Z axis, horizontal displacement along X or Y axis, angle displacement (i.e., rotation). Non-rigid deformation is more complex than a rigid one as it concerns relative changes inside the object. Typical non-rigid deformation types include scaling, shearing, and strain. In this thesis, we mainly focus on rigid deformation like displacement, angle variation, and their combinations.
- *How*: In addition to identify "what" object has "which" type of deformation, a complete model should include a suitable description strategy. We identified two types of assessments, namely the *quantitative* measurement and the *qualitative* measurement. The first one uses numeric values to describe the deformation whilst the second one applies meaningful terms (like "big", "small" deformation). In this thesis, we mainly focus on computing the *quantitative* deformation information by using 3D point cloud datasets captured by LiDAR.

Global Deformation Modelling vs. Hybrid Modelling. After finding the three components of deformation modelling, we identified that traditional deformation analysis methods based on geodetic surveying typically focus on only one type of model (i.e., the same deformation type - "which", as well as the same deformation measurement - "how"). This is mainly because of the limited points in traditional observation techniques. However, for monitoring real-life large object like a big slope, deformation can be heterogeneous (e.g., with varying deformation characteristics, with mixed deformation types for different partial areas). To solve this problem, we designed a new hybrid modelling method besides traditional global modelling.

- *Global Modelling*: The global deformation modelling method utilises a single deformation type for a whole monitoring area. In traditional geodetic surveying, such global modelling methods are largely used, e.g., a uniform deformation type like displacement, angle variation, scaling, shearing based on a small set of data points. In addition, the model is applied to the entire monitored object/region.

- *Hybrid Modelling*: For point cloud data from LiDAR, global modelling cannot make good use of such high-density 3D data points. Therefore, we designed a novel hybrid model, which includes "meta-deformation" for a small area (e.g., cells in our case), "sub-deformation" for a partial area where the deformation is more or less uniform or consistent, and "deformation-map" for describing the complete region. With such a hybrid deformation model, we are able to provide a more informative description of heterogeneous deformation for a large monitored region. Furthermore, the deformation can be computed incrementally according to this hybrid modelling.

Cell-based Deformation Computation. To effectively utilise the hybrid deformation modelling for a large region and to efficiently compute the deformation map, we design a novel and self-contained cell-based deformation computing framework. With this framework, we are able to automatically and incrementally compute the deformation information, from the meta-deformation for cells, to the sub-deformation for partial areas, and finally to the deformation map for the complete region. This framework has three main steps (technical components), namely, "split", "detect", and "merge".

- *Split*: By using R-tree based spatial data indexing and access techniques, our framework can split the large region into cells and access the data points in each cell very efficiently. Instead of retrieving the complete data each time, we can reorganise the data in terms of using cells. This can make our subsequent deformation computing process much easier, faster and convenient.
- *Detect*: For each cell, we design several algorithms for detecting the uniform deformation using the data points inside the cell at two epochs (Epoch I and Epoch II). We study both single rigid deformation including displacement (with different types of parameters like 'min', 'max', 'median', 'mean', 'centre') and angle variation, and mixed rigid deformation using the ICP (Iterative Closest Point) algorithm.
- *Merge*: This is designing a grouping algorithm to merge adjacent cells with consistent deformation status into a uniform partial area, i.e., forming a sub-deformation. Finally, all sub-deformation partial areas can compose a complete deformation map. It is worth noting that merging can smooth the deformation in sub-area boundaries, which can bring discontinuities in some scenarios. Nevertheless, the merging process can generate higher-level abstraction about deformation description.

Multiple-aspect Evaluation. To evaluate our hybrid deformation modelling and our cell-based deformation computation, we studied on three different types of point cloud datasets, from simulated data using programming, to data captured in the practical experiments, and finally to the real-life deformation data from LiDAR.

- *Landslide Simulation*: The first main experimental evaluation is based on a landslide simulation using a computer program. We simulated a slope slid into two sub slopes, and tested different models like displacement, angle variation, as well as rigid transformation.

We compared cell-based hybrid deformation modelling with global modelling, and observed better performance from our cell-based approach.

- *Practical Experiment*: We additionally performed a practical experiment, using the "Sandbox" and the Konica Minolta VIVID 3D laser scanner from UNIL (University of Lausanne). We designed two scenarios: one is focusing on the movement of an object (like a stone) on the slope, and the other one is the movement of the slope itself (like erosion).
- *Real-life Cases*: More importantly, we tested our method using two real-life deformation cases in Switzerland. One is the landslide in Flamatt, for which we compared the results with the sources provided by the surveying company (BBHN-SA). The other real-life case is the erosion on a slope under a big dam in Valais. We compared our cell-based approach with 3DReshaper, a commercial software for point cloud processing.

Extension of the Cell-based Approach. To improve the capability of our cell-based approach for deformation monitoring, we designed a couple of important extensions. For the split step, we designed "dynamic cells" to automatically build different cell sizes and "cell correspondence" to find the best matching cells between two epochs. To enhance the detection step, we proposed a method of "adaptive model selection" for each cell. In addition, we considered "deformation evolution" over a long period with multiple epoch point cloud datasets.

- *Designing Dynamical Cells*: Previously, the cell-based deformation monitoring approach utilises uniform and static cells, which means that we build meta-deformation for each cell with the same size. It is hard to find a very suitable fixed cell size: if it is too small, there will be a lot of cells and building meta-deformation for so many cells is time-consuming and not efficient; if it is too big, the meta-deformation of each cell may be not accurate. Therefore, we provided a solution that can design dynamic cells, based on the top-down split strategy according to model errors.
- *Finding Cell Correspondence*: In some cases, if the deformation is very large (e.g., the horizontal displacement is bigger than a cell size), cells with the same ID at two epochs do not have the same content and have no direct correspondence (see Figure 6.3). Therefore, we designed a method to find the cell correspondence. In this thesis, we applied a matching method to find such cell correspondence based on local search optimisation. Particularly, we tested our method in the experimental rockfall scenario, to find the cells that contain the same rock at different epochs.
- *Movement tracking with data-driven cell*: We focused on "regular 3D cell" and "direct cell correspondence", where deformation is restricted to an exact multiple of the cell size (e.g., inside one cell or deformation from one cell to neighbouring cells). However, real-life cases are much more complex. For example, the cell cannot be always regular. Deformation has its own boundary. In addition, deformation could happen from one cell to another cell with overlapping part. Therefore, the exact multiple of the cell size is not always adequate. To solve this problem, we introduce the "meta-cell" concept into the "movement tracking"

extension and utilise the meta-cell to form "a data-driven cell with arbitrary boundary". Furthermore, the meta-cell can assist in finding a better cell correspondence. Now, the moving distance of a searching cell can not only be an exact multiple of the cell size, but also be non-integer-multiple. We applied this algorithm on a rockfall scenario with our experimental datasets, i.e., tracking the moving parts like a rock with a series of epochs.

- *Building Adaptive Modelling*: In the early stage of this thesis, each cell was built with the same model, e.g., finding point cloud centre to calculate the displacement using a uniform linear regression model for each cell. However, an advanced method should be able to pick the most suitable model for each cell automatically. In such a case, we adaptively select a suitable model to build regression on points in each cell. The algorithm is able to choose between 1st-order linear model and 2nd-order polynomial model according to the characteristics of point clouds in the two corresponding cells.

- *Computing Deformation Evolution*: Finally, we studied the deformation evolution based on multiple epochs of point cloud datasets captured by LiDAR. With regards to our erosion case study in Valais, there are four epochs of point cloud datasets. We identified that the main deformation happened between the first and the second epochs. There was only some slight deformation between Epoch 2, Epoch 3, and Epoch 4.

In summary, the major novel contributions of this thesis is to provide hybrid rich deformation modelling as well as designing a cell-based deformation computation approach. The hybrid model can capture heterogeneous deformation characteristics and generate an informative deformation map for representing the deformation of a large area. The cell-based computation process can automatically and incrementally generate the deformation map from raw 3D point cloud data sets. This novel model and computing approach can be applied to many real-life deformation scenarios using LiDAR data. More importantly, the cell-model can support various deformation types, and real-life irregular cells can also be applied.

7.2 Limitations and Outlook

As an interdisciplinary study, this thesis proposed a cell-based approach on deformation monitoring in the environmental scenarios. This work combined a couple of computer science methods (e.g., cell-based conceptual modelling, splitting and merging, point cloud regression, ICP modelling) with traditional geodetic engineering for deformation analysis (e.g., landslides, erosion, subsidence) using LiDAR datasets. From a high-level and methodology perspective, this thesis has a valuable contribution to this deformation monitoring field, in terms of providing an interdisciplinary approach, i.e., our cell-based deformation monitoring via 3D point clouds. We also provided many supporting detailed algorithmic solutions, for example: different types of cell splitting (uniform cells vs. dynamic cells), different types of meta-deformation detection (e.g., rigid deformation like displacement, angle variation, and ICP), adaptive modelling, and deformation evolution. Nevertheless, we identify following limitations and possible future works as the outlook of this thesis.

Supporting Rich Meta-Deformation Modelling. For current meta-deformation detection for cells, the thesis primarily focused on rigid deformation like displacement, angle variation, and ICP. In future, this work could consider more advanced models in traditional geodetic surveying techniques. For example, we can apply non-rigid models (like scaling, shearing, strain) to build meta-deformation based on point clouds in a cell. This could be quite challenging as these traditional non-rigid models are largely based on a small set of data observations, but not for point clouds directly. In addition, advanced graphic-based model (like mesh) can be applied for each cell to build meta-deformation.

Enhanced Method in Finding Cell-Correspondence. In this thesis, both splitting the object space into cells and merging the cells into a deformation map are based on some hierarchical methods, either top-down or bottom-up. To enhance our cell strategy, we designed extensions using "dynamic cells" and building "cell correspondence". However, for real-life deformation datasets, it is even more difficult to find the correct cell correspondence for a complex scenario (e.g., different cells mix up between two epochs, $cell_1^{(I)} \rightarrow cell_2^{(II)}, cell_2^{(I)} \rightarrow cell_3^{(II)}, cell_3^{(I)} \rightarrow cell_1^{(II)}$). The current solution to find cell correspondence is based on a local optimisation, i.e., finding $cell_i^{(I)}$'s corresponding cell at Epoch II does not consider other cells at Epoch I. As an outlook, it is useful to design a global optimisation solution to find best correspondences for all cells between two epochs.

Post-processing & Statistical-tests of Deformation Map. In the current merge step that generates a deformation map from the meta-deformation of the cells, we have already designed a post-processing sub-step that deals with the outliers, where the meta deformation of one cell is significantly different from that of neighbouring cells. For example, the vegetation on the ground such as bush or grass could bring such outliers. In this thesis, the intuitive method consists in replacing the meta-deformation of the discrepant cell with the value of its neighbours. In real-life, it is non-trivial to judge whether a discrepant point is an outlier or a distinguishing valuable point. Future work should put more efforts on dealing with noisy data in the point clouds. In addition, statistical tests can be applied to validate the results of the final deformation map.

Long-term Deformation Monitoring and Forecasting. In current experiments on deformation evolution, we only tested a displacement model for detecting deformation evolution for our Valais scenario with four epochs of point cloud datasets. Future work could analyse long terms deformation evolution using advanced meta-deformation models with datasets for more epochs. In addition, a trend model can be established, which can provide valuable feedbacks for future data capture in the field. For example, if the deformation is huge, we can advise field engineers to capture data more frequently; on the contrary, if the deformation is quite small, we can spread the data collection. In such case, we can build adaptive scheduling for the field investigation, which will save cost for field data capture using LiDAR. In addition, the trend model can be used to forecast future deformation.

Bibliography

[AFP$^+$06] L. Alba, L. Fregonese, F. Prandi, M. Scaioni, and P. Valgoi. Structural Monitoring of a Large Dam by Terrestrial Laser Scanning. In *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, volume 36, 2006.

[AGB$^+$99] F. Amelung, D. L. Galloway, J. W. Bell, H. A. Zebker, and R. J. Laczniak. Sensing the Ups and Downs of Las Vegas: INSAR Reveals Structural Control of Land Subsidence and Aquifer-System Deformation. *Geology*, 27(6):483–486, 1999.

[AJOV09] A. Abellán, M. Jaboyedoff, T. Oppikofer, and J. M. Vilaplana. Detection of Millimetric Deformation Using a Terrestrial Laser Scanner: Experiment and Application to a Rockfall Event. *Natural Hazards and Earth System Science*, 9(2):365–372, 2009.

[Akc07] D. Akca. *Least Squares 3D Surface Matching.* PhD thesis, ETH, Zurich, Switzerland, 2007.

[AMGC02] M. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking. *IEEE Transactions on Signal Processing*, 50(2):174 –188, 2002.

[AOA$^+$06] M. Acar, M. T. Özlüdemir, O. Akyilmaz, R. N. Çelik, and T. Ayan. Deformation Analysis with Total Least Squares. *Natural Hazards and Earth System Science*, 6(4):663–669, 2006.

[ARS06] A. Adam, E. Rivlin, and I. Shimshoni. Robust Fragments-based Tracking Using the Integral Histogram. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 798–805, Washington, DC, USA, 2006.

[AZGK09] M. H. Aly, H. A. Zebker, J. R. Giardino, and A. G. Klein. Permanent Scatterer Investigation of Land Subsidence in Greater Cairo, Egypt. *Geophysical Journal International*, 178(3):1238–1245, 2009.

[BC86] T. J. Broida and R. Chellappa. Estimation of Object Motion Parameters from Noisy Images. *IEEE Transactions on Pattern Analysis and Machine*

Intelligence, 8(1):90–99, 1986.

[BCFZ08] P. Baldi, N. Cenni, M. Fabris, and A. Zanutta. Kinematics of a Landslide Derived from Archival Photogrammetry and GPS Data. *Geomorphology*, 102(3-4):435–444, 2008.

[BCK08] J. Bond, A. Chrzanowski, and D. Kim. Bringing GPS into Harsh Environments for Fully Automated Deformation. *GPS Solutions*, 12(1):1–11, 2008.

[Ben75] J. L. Bentley. Multidimensional Binary Search Trees Used for Associative Searching. *Communications of the ACM*, 18(9):509–517, 1975.

[Bha10] B. Bhatta. *Global Navigation Satellite Systems: Insights into GPS, GLONASS, Galileo, Compass, and Others*. BS Publications, 2010.

[BHK98] J. A. Behr, K. W. Hudnut, and N. E. King. Monitoring Structural Deformation at Pacoima Dam, California Using Continuous GPS. In *Proceedings of ION GPS*, volume 11, pages 59–68, 1998.

[BHM01] W. Boehler, G. Heinz, and A. Marbs. The Potential of Non-contact Close Range Laser Scanners for Cultural Heritage Recording. In *Proceeding of 18th International Symposium of CIPA (Committee for Documentation of Cultural Heritage), Potsdam, Germany*, pages 330–336, 2001.

[Bir98] S. Birchfield. Elliptical Head Tracking Using Intensity Gradients and Color Histograms. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, CVPR 98, pages 232–237, Washington, DC, USA, 1998.

[BK99] D. Beymer and K. Konolige. Real-time Tracking of Multiple People Using Continuous Detection. In *IEEE International Conference on Computer Vision, Frame-Rate Workshop*, 1999.

[BKSS90] N. Beckmann, H.-P. Kriegel, R. Schneider, and B. Seeger. The R*-tree: an Efficient and Robust Access Method for Points and Rectangles. In *Proceedings of the 1990 ACM International Conference on Management of Data (SIGMOD)*, pages 322–331, New York, USA, 1990.

[BL07] K. H. Bae and D. D. Lichti. On-site Self-calibration Using Planar Features for Terrestrial Laser Scanners. In *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, volume 36, pages 14–19, 2007.

[BM92] P. J. Besl and N. D. McKay. A Method for Registration of 3-D Shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, 1992.

[BP07] A. Bugeau and P. Perez. Detection and Segmentation of Moving Objects in Highly Dynamic Scenes. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2007.

[BR02] F. Bernardini and H. Rushmeier. The 3D model Acquisition Pipeline. *Com-*

puter Graphics Forum, 21(2):149–172, 2002.

[Bro92] L. G. Brown. A Survey of Image Registration Techniques. *ACM Computing Surveys*, 24(4):325–376, 1992.

[BS11] L. Barazzetti and M. Scaioni. Photogrammetric Tools for Deformation Measurements. In *XX Congresso Associazione Italiana di Meccanica Teorica e Applicata (AIMETA)*, page 10, 2011.

[BVT92] C. Bradley, G. W. Vickers, and J. Tlusty. Automated Rapid Prototyping Utilizing Laser Scanning and Free-form Machining. *CIRP Annals - Manufacturing Technology*, 41(1):437–440, 1992.

[BWL+09] X. Bai, X. Wang, L. Latecki, W. Liu, and Z. Tu. Active Skeleton for Non-rigid Object Detection. In *the 12th IEEE International Conference on Computer Vision (ICCV)*, pages 575–582, 2009.

[Cas88] W. F. Caspary. *Concepts of Network and Deformation Analysis.* University of New South Wales (UNSW), Sydeney, Australia, 1988.

[CBS00] F. Chen, G. M. Brown, and M. Song. Overview of Three-dimensional Shape Measurement Using Optical Methods. *Optical Engineering*, 39(1):10–22, 2000.

[CC88] M. A. R. Cooper and P. A. Cross. Statistical Concepts and their Application in Photogrammetry and Surveying. *The Photogrammetric Record*, 12(71):637–663, 1988.

[CC08] J. Chen and B. Chen. Architectural Modeling from Sparsely Scanned Range Data. *International Journal of Computer Vision*, 78(2-3):223–236, 2008.

[CCB+05] M. Crosetto, B. Crippa, E. Biescas, O. Monserrat, and M. Agudo. State of the Art of Land Deformation Monitoring Using Differential SAR Interferometry. In *ISPRS Hannover Workshop on High Resolution Earth Imaging for Geospatial Information*, pages 17–20, 2005.

[CCDMW+08] E. Cabral-Cano, T. H. Dixon, F. Miralles-Wilhelm, O. Díaz-Molina, O. Sánchez-Zamora, and R. E. Carande. Space Geodetic Imaging of Rapid Ground Subsidence in Mexico City. *Geological Society of America Bulletin*, 120:1556–1566, 2008.

[CCE+04] P. Canuti, N. Casagli, L. Ermini, R. Fanti, and P. Farina. Landslide Activity as a Geoindicator in Italy: Significance and New Perspectives from Remote Sensing. *Environmental Geology*, 45(7):907–919, 2004.

[CCFM08] U. Castellani, M. Cristani, S. Fantoni, and V. Murino. Sparse Points Matching by Combining 3D Mesh Saliency with Statistical Descriptors. *Computer Graphics Forum*, 27(2):643–652, 2008.

[CCRS86] A. Chrzanowski, Y. Chen, P. Romero, and J. M. Secord. Integration of Geodetic and Geotechnical Deformation Surveys in the Geosciences. *Tectonophysics*, 130(1-4):369–383, 1986.

[CCS90] Y. Chen, A. Chrzanowski, and J. Secord. A Strategy for the Analysis of the

Stability of Reference Points in Deformation Surveys. *CISM Journal*, 44(2):39–46, 1990.

[CDBA03] B. Casson, C. Delacourt, D. Baratoux, and P. Allemand. Seventeen Years of the "La Clapiére" Landslide Evolution Analysed from Ortho-rectified Aerial Photographs. *Engineering Geology*, 68(1-2):123–139, 2003.

[CEO01] CEOS Disaster Management Support Group. Landslides. Technical report, Final Report of the CEOS Disaster Management Support Group, 2001.

[CF65] N. L. Carter and M. Friedman. Dynamic Analysis of Deformed Quartz and Calcite from the Dry Creek Ridge Anticline, Montana. *American Journal of Science*, 263(9):747–785, 1965.

[CFN$^+$03] C. Colesanti, A. Ferretti, F. Novali, C. Prati, and F. Rocca. SAR Monitoring of Progressive and Seasonal Ground Deformation Using the Permanent Scatterers Technique. *IEEE Transactions on Geoscience and Remote Sensing*, 41(7):1685–1701, 2003.

[CFP09] L. Cascini, G. Fornaro, and D. Peduto. Analysis at Medium Scale of Low-Resolution DInSAR Data in Slow-Moving Landslide-affected Areas. *ISPRS Journal of Photogrammetry and Remote Sensing*, 64(6):598–611, 2009.

[Che83] Y. Q. Chen. Analysis of Deformation Surveys - a Generalized Method. Technical report, No. 94., Department of Surveying Engineering, University of New Brunswick, Fredericton, NB, Canada, 1983.

[Chr81] A. Chrzanowski. A Comparison of Different Approaches into the Analysis of Deformation Measurements. In *Proceedings of FIG-XVI Congress, Paper No. 602.3*, 1981.

[Chr86] A. Chrzanowski. Geotechnical and Other Non-Geodetic Method in Deformation Measurement. In *Proceedings of the Deformation Measurement Workshop. October 31st-November 1st, Editors: Yehuda Bock, Massachusetts Institute of Technology, Cambridge, MA.*, pages 112–152, 1986.

[CKS97] V. Caselles, R. Kimmel, and G. Sapiro. Geodesic Active Contours. *International Journal of Computer Vision*, 22:61–79, 1997.

[CLG11] J. Chow, D. Lichti, and C. Glennie. Point-based versus Plane-based Self-calibration of Static Terrestrial Laser Scanners. In *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, ISPRS Workshop Laser Scanning 2011*, volume XXXVIII-5/W12, 2011.

[CLT10] J. C. K. Chow, D. D. Lichti, and W. F. Teskey. Self-calibration of the Trimble (Mensi) GS200 Terrestrial Laser Scanner. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, XXXVI, 2010.

[CMK96] C. Carnec, D. Massonnet, and C. King. Two Examples of the Use of SAR Interferometry on Displacement Fields of Small Spatial Extent. *Geophysical Research Letters*, 23(24):3579–3582, 1996.

[CMM02] D. Comaniciu, P. Meer, and S. Member. Mean Shift: a Robust Approach toward Feature Space Analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24:603–619, 2002.

[COCC⁺12] F. Cigna, B. Osmanoglu, E. Cabral-Cano, T. H. Dixon, J. A. Ávila Olivera, V. H. Garduño Monroy, C. DeMets, and S. Wdowinski. Monitoring Land Subsidence and its Induced Geological Hazard with Synthetic Aperture RADAR Interferometry: a Case Study in Morelia, Mexico. *Remote Sensing of Environment*, 117:146–161, 2012.

[Coo84] M. A. R. Cooper. Deformation Measurement by Photogrammetry. *Photogrammetric Record*, 11(63):291–301, 1984.

[CRH09] M. Choudhury, C. Rizos, and B. Harvey. A Survey of Techniques and Algorithms in Deformation Monitoring Applications and the Use of the Locata Technology for Such Applications. In *22nd International Technical Meeting of the Satellite Division of the U.S. Institution of Navigation, Savannah, Georgia, USA, 22-25 September*, pages 668–678, 2009.

[CRM03] D. Comaniciu, V. Ramesh, and P. Meer. Kernel-based Object Tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(5):564–577, 2003.

[CSR10] M. D. Calabro, D. A. Schmidt, and J. J. Roering. An Examination of Seasonal Deformation at the Portuguese Bend Landslide, Southern California, Using RADAR Interferometry. *Journal of Geophysical Research*, 115(F2):1–10, 2010.

[CTA97] P. Checchin, L. Trassoudaine, and J. Alizon. Segmentation of Range Images into Planar Regions. In *International Conference on Recent Advances in 3-D Digital Imaging and Modeling*, pages 156–163, 1997.

[CTL08] J. Chow, W. Teskey, and D. D. Lichti. Self-calibration and Evaluation of the Trimble GX Terrestrial Laser Scanner. In *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, volume Commission I, WG I/3., 2008.

[CW04] C. Colesanti and J. Wasowski. Satellite SAR Interferometry for Wide-area Slope Hazard Detection and Site-Specific Monitoring of Slow Landslides. In *Proceedings 9th International Symposium on Landslides, Rio de Janeiro, Lacerda WA et al. (ed) Landslides, Evaluation & Stabilization.*, pages 117–125, 2004.

[CWC04] K. Cheng, C. Wei, and S. Chang. Locating Landslides Using Multitemporal Satellite Images. *Advances in Space Research*, 33:296–301, 2004.

[DACV04] C. Delacourt, P. Alleman, B. Casson, and H. Vadon. Velocity Field of the “La Clapiere” Landslide Measured by the Correlation of Aerial and Quickbird Satellite Images. *Geophysical Research Letters*, 31, 2004.

[Dav79] R. W. Davidge. *Mechanical Behavior of Ceramics.* Cambridge Solid State

Science Series, 1979.

[DGK12] M. Debella-Gilo and A. Kaab. Measurement of Surface Displacement and Deformation of Mass Movements Using Least Squares Matching of Repeat High Resolution Satellite and Aerial Images. *Remote Sensing*, 4(1):43–67, 2012.

[DK97] V. Der Kooij. Land Subsidence Measurements at the Belridge Oil Fields from ERS InSAR Data. In *European Space Agency Special Publication ESA SP-414, Proceeding of third ERS Symposium*, pages 1853–1858, 1997.

[DL83] A. Dermanis and E. Livieratos. Applications of Deformation Analysis in Geodesy and Geodynamics. *Reviews of Geophysics*, 21(1):41–50, 1983.

[DM00] D. DeCarlo and D. Metaxas. Optical Flow Constraints on Deformable Models with Applications to Face Tracking. *International Journal of Computer Vision*, 38:99–127, 2000.

[DNPM08] P. Dorninger, C. Nothegger, N. Pfeifer, and G. Molnár. On-the-job Detection and Correction of Systematic Cyclic Distance Measurement Errors of Terrestrial Laser Scanners. *Journal of Applied Geodesy*, 2(4):191–204, 2008.

[DPHW05] Y. Ding, X. Ping, M. Hu, and D. Wang. Range Image Segmentation Based on Randomized Hough Transform. *Pattern Recognition Letters*, 26(13):2033–2041, 2005.

[DR08] J. Donovan and A. W. Raza. A Change Detection Method for Slope Monitoring and Identification of Potential Rockfall Using Three-dimensional Imaging. In *the 42nd U.S. Rock Mechanics Symposium (USRMS)*, 2008.

[Ede01] H. Edelsbrunner. *Geometry and Topology for Mesh Generation.* Cambridge Monographs on Applied and Computational Mathematics. Cambridge University Press, 2001.

[EEA04] S. Erol, B. Erol, and T. Ayan. A General Review of the Deformation Monitoring Techniques and A Case Study: Analyzing Deformations Using GPS/Leveling. In *Procedings of XXth ISPRS Congress Geo-Imagery Bridging Continents*, pages 12–23, 2004.

[EhBPG04] S. F. El-hakim, J. A. Beraldin, M. Picard, and G. Godin. Detailed 3D Reconstruction of Large-scale Heritage Sites with Integrated Techniques. *IEEE Computer Graphics and Applications*, 24(3):21–29, 2004.

[EKB10] R. Engelkemeir, S. D. Khan, and K. Burke. Surface Deformation in Houston, Texas Using GPS. *Tectonophysics*, 490(1-2):47–54, July 2010.

[ER02] A. El-Rabbany. *Introduction to GPS: the Global Positioning System.* Artech House, 2002.

[ERH10] E. Erten, A. Reigber, and O. Hellwich. Generation of Three-dimensional Deformation Maps from INSAR Data Using Spectral Diversity Techniques. *ISPRS Journal of Photogrammetry and Remote Sensing*, 65(4):388–394, 2010.

[FAD96] B. Fruneau, J. Achache, and C. Delacourt. Observation and Modelling of the

Saint-Étienne-de-Tinée Landslide Using SAR Interferometry. *Tectonophysics*, 265:181–190, 1996.

[FB81] M. A. Fischler and R. C. Bolles. Random Sample Consensus: a Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[FBG98] E. J. Fielding, R. G. Blom, and R. M. Goldstein. Rapid Subsidence over Oil Fields Measured by SAR Interferometry. *Geophysical Research Letters*, 25(17):3215–3218, 1998.

[FCB$^+$08] R. Fell, J. Corominas, C. Bonnard, L. Cascini, E. Leroi, and W. Z. Savage. Guidelines for Landslide Susceptibility, Hazard and Risk Zoning for Land Use Planning. *Engineering Geology*, 102(3-4):85–98, 2008.

[Fie04] H. L. Field. *Landscape Surveying*. Cengage Learning, 2004.

[FM04] C. Frühlich and M. Mettenleiter. Terrestrial Laser Scanning - New Perspectives in 3D Surveying. In *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, volume XXXVI - 8/W2, 2004.

[FMN87] T. J. Fan, G. Medioni, and R. Nevatia. Segmented Descriptions of 3-D Surfaces. *IEEE Journal of Robotics and Automation*, 3(6):527–538, 1987.

[FR00] C. S. Fraser and B. Riedel. Monitoring the Thermal Deformation of Steel Beams via Vision Metrology. *ISPRS Journal of Photogrammetry and Remote Sensing*, 55(4):268–276, 2000.

[FSC$^+$07] J. C. Fernandez, A. Singhania, J. Caceres, K. C. Slatton, M. Starek, and R. Kumar. An Overview of LIDAR Point Cloud Processing Software. Technical report, No. Rep-2007-12-001, Geosensing Engineering and Mapping (GEM), University of Florida, 2007.

[GA05] A. Gruen and D. Akca. Least Squares 3D Surface and Curve Matching. *ISPRS Journal of Photogrammetry and Remote Sensing*, 59(3):151–174, 2005.

[GCR00] J. A. Gili, J. Corominas, and J. Rius. Using Global Positioning System Techniques in Landslide Monitoring. *Engineering Geology*, 55(3):167–192, 2000.

[GG01] M. Greenspan and G. Godin. A Nearest Neighbor Method for Efficient ICP. In *Proceedings of the third International Conference on 3-D Digital Imaging and Modeling*, pages 161–168, 2001.

[GGE09] S. Gleason and D. Gebre-Egziabher. *GNSS Applications and Methods*. GNSS Technology and Applications. Artech House, 2009.

[Gho93] S. Ghosal. Segmentation of Range Images: an Orthogonal Moment-based Integrated Approach. *IEEE Robotics and Automation*, 9(4), 1993.

[GKHMA06] G. Guler, H. Kilic, G. Hosbas, and K. O. M. Asce. Evaluation of the Movements of the Dam Embankments by means of Geodetic and Geotechnical Methods. *Journal of Surveying Engineering*, pages 31–39, 2006.

[GLFS04] S. Gordon, D. Lichti, J. Franke, and M. Stewart. Measurement of Structural Deformation Using Terrestrial Laser Scanners. In *1st FIG International Symposium on Engineering Surveys for Construction Works and Structural Engineering*, 2004.

[GLP06] R. V. Gosliga, R. Lindenbergh, and N. Pfeifer. Deformation Analysis of a Bored Tunnel by means of Terrestrial Laser Scanning. In *International Archives of Photogrammetry, Remote Sensing and Spatial Information*, volume 36, 2006.

[GMA+09] F. Guzzetti, M. Manunta, F. Ardizzone, A. Pepe, M. Cardinali, G. Zeni, P. Reichenbach, and R. Lanari. Analysis of Ground Deformation Detected Using the SBAS-DInSAR Technique in Umbria, Central Italy. *Pure and Applied Geophysics*, 166(8-9):1425–1459, 2009.

[GMV06] P. Guha, A. Mukerjee, and K. S. Venkatesh. Spatio-temporal Discovery: Appearance + Behavior = Agent. In *Proceedings of the 5th Indian conference on Computer Vision, Graphics and Image Processing (ICVGIP)*, pages 516–527, 2006.

[GRG04] F. Gielsdorf, A. Rietdorf, and L. Gruendig. A Concept for the Calibration of Terrestrial Laser Scanners. In *FIG Working Week*, 2004.

[GS04] A. Giussani and M. Scaioni. Applications of TLS to Support Landslides Study: Survey Planning, Operational Issues and Data Processing. In *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, volume 36, pages 318–323, 2004.

[GS08] V. Gikas and M. Sakellariou. Horizontal Deflection Analysis of a Large Earthen Dam by means of Geodetic and Geotechnical Methods. In *13th FIG Symposium on Deformation Measurement and Analysis/4th IAC Symposium on Geodesy for Geotechnical and Structural Engineering, May 12-15*, 2008.

[GSC+06] N. Glenn, D. Streutker, D. Chadwick, G. Thackray, and S. Dorsch. Analysis of LiDAR-derived Topographic Information for Characterizing and Differentiating Landslide Morphology and Activity. *Geomorphology*, 73(1-2):131–148, 2006.

[GSS93] N. Gordon, D. Salmond, and A. Smith. Novel Approach to Nonlinear/Non-Gaussian Bayesian State Estimation. *IEEE Proceedings on Radar and Signal Processing*, 140(2):107 –113, 1993.

[GVEHG04] A. Guarnieri, A. Vettore, S. El-Hakim, and L. Gonzo. Digital Photogrammetry and Laser Scanning in Cultural Heritage Survey. In *XXth Congress of the International Society for Photogrammetry and Remote Sensing*, 2004.

[GVG96] R. Gens and J. L. Van Genderen. SAR Interferometry - Issues, Techniques, Applications. *International Journal of Remote Sensing*, 17(10):1803–1835, 1996.

[GW08] C. D. Ghilani and P. R. Wolf. *Elementary Surveying: an Introduction to Geomatics*. Pearson Education, 2008.

[GY03] M. Greenspan and M. Yurick. Approximate K-d Tree Search for Efficient ICP.

In *Proceedings of Fourth International Conference on 3-D Digital Imaging and Modeling (3DIM)*, pages 442–448, 2003.

[GZZ+12] J. Gong, Q. Zhu, R. Zhong, Y. Zhang, and X. Xie. An Efficient Point Cloud Management Method Based on a 3D R-tree. *Photogrammetric Engineering & Remote Sensing*, 78(4):373–381, 2012.

[HBH09] N. Hilker, A. Badoux, and C. Hegg. The Swiss Flood and Landslide Damage Database 1972-2007. *Natural Hazards and Earth System Science*, 9(3):913–925, 2009.

[HC91] S. Huang and B. Chen. Integration of Landsat and Terrain Information for Landslide Study. In *Proceedings of 8th Thematic Conference on Geological Remote Sensing*, pages 743–754, 1991.

[Hei04] A. Hein. *Processing of SAR Data: Fundamentals, Signal Processing, Interferometry.* Signals and Communication Technology. Springer, 2004.

[Her03] J. Hervas. Monitoring Landslides from Optical Remotely Sensed Imagery: the Case History of Tessina Landslide, Italy. *Geomorphology*, 54(1-2):63–75, 2003.

[HHA+06] W. Haeberli, B. Hallet, L. Arenson, R. Elconin, O. Humlum, and A. Ka. Permafrost Creep and Rock Glacier Dynamics. *Permafrost and Periglacial Processes*, 214(June):189–214, 2006.

[HM01] J. Huang and C.-h. Menq. Automatic Data Segmentation for Geometric Feature Extraction from Unorganized 3-D Coordinate Points. *IEEE Transactions on Robotics and Automation*, 17(3):268–279, 2001.

[HNR93] D. Huttenlocher, J. Noh, and W. Rucklidge. Tracking Non-rigid Objects in Complex Scenes. In *Proceedings of fourth International Conference on Computer Vision*, pages 93–101, 1993.

[HR01] S. Halim and S. Ranjit. Deformation Analysis of a Geodetic Monitoring Network. *Geomatica*, 55(3), 2001.

[HS81] B. K. Horn and B. G. Schunck. Determining Optical Flow. *Artificial Intelligence*, 17(1-3):185 – 203, 1981.

[HS88] C. Harris and M. Stephens. A Combined Corner and Edge Detection. In *Proceedings of the fourth Alvey Vision Conference*, pages 147–151, 1988.

[Hud98] J. A. Hudnut K. W. Behr. Continuous GPS Monitoring of Structural Deformation at Pacoima Dam, California. *Seismological Research Letters*, 69; NU:299–308, 1998.

[HW86] T. Heliker Griggs and Wright. Volcano Monitoring at the U.S. Geological Survey's Hawaiian Volcano Observatory. *Earthquake Information Bulletin*, 18, 1986.

[IDB97] S. Intille, J. Davis, and A. Bobick. Real-time Closed-world Tracking. In *Proceedings 1997 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 697–703, 1997.

[IVGT06] C. Ioannidis, A. Valani, A. Georgopoulos, and E. Tsiligiris. 3D Model Generation for Deformation Analysis Using Laser Scanning Data of a Cooling Tower. In *3rd IAG/12th FIG Symposium*, 2006.

[JK11] E. Jayabalan and A. Krishnan. Object Detection and Tracking in Videos Using Snake and Optical Flow Approach. In *Computer Networks and Information Technologies*, volume 142 of *Communications in Computer and Information Science*, pages 299–301. 2011.

[JOA$^+$10] M. Jaboyedoff, T. Oppikofer, A. Abellán, M.-H. Derron, A. Loye, R. Metzger, and A. Pedrazzini. Use of LIDAR in Landslide Investigations: a Review. *Natural Hazards*, 61(1):1–24, 2010.

[JT80] C. Jackins and S. Tanimoto. Oc-trees and their Use in Representing Three-dimensional Objects. *Computer Graphics and Image Processing*, 14(3):249–270, 1980.

[KÖ0] A. Kääb. Photogrammetry for Early Recognition of High Mountain Hazards: New Techniques and Applications. *Physics and Chemistry of the Earth, Part B: Hydrology, Oceans and Atmosphere*, 25(9):765–770, 2000.

[Kaa02] A. Kaab. Monitoring High-mountain Terrain Deformation from Repeated Air- and Spaceborne Optical Data: Examples Using Digital Aerial Imagery and ASTER data. *ISPRS Journal of Photogrammetry and Remote Sensing*, 57:39–52, 2002.

[KAB10] Y. Kalkan, R. M. Alkan, and S. Bilgi. Deformation Monitoring Studies at Ataturk Dam. In *FIG Congress*, 2010.

[KC09] S. Krinidis and V. Chatzis. Fuzzy Energy-based Active Contours. *IEEE Transactions on Image Processing*, 18(12):2747–2755, 2009.

[KCM04] J. Kang, I. Cohen, and G. Medioni. Object Reacquisition Using Invariant Appearance Model. In *Proceedings of 17th International Conference on the Pattern Recognition (ICPR)*, pages 759–762, Washington, DC, USA, 2004.

[KF88] H. Kahmen and W. Faig. *Surveying.* Walter de Gruyter, 1988.

[KHG97] A. Kaab, W. Haeberli, and G. H. Gudmundsson. Analysing the Creep of Mountain Permafrost Using High Precision Aerial Photogrammetry: 25 Years of Monitoring Gruben Rock Glacier, Swiss Alps. *Permafrost and Periglacial Processes*, 8:409–426, 1997.

[KHK07] K. Kraus, I. Harley, and S. Kyle. *Photogrammetry: Geometry from Images and Laser Scans.* De Gruyter Textbook. 2007.

[KKOF04] R. Kadobayashi, N. Kochi, H. Otani, and R. Furukawa. Comparison and Evaluation of Laser Scanning and Photogrammetry and their Combined Use for Digital Recording of Cultural Heritage. In *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, volume 35, pages 401–406, 2004.

[KM95] T. Kersten and H. G. Maas. Photogrammetric 3-D Point Determination for Dam Monitoring. *Optical 3-D Measurement III*, pages 161–168, 1995.

[KM98] R. Klees and D. Massonnet. Deformation Measurements Using SAR Interferometry: Potential and Limitations. *Geologie en Mijnbouw*, 77(2):161–176, 1998.

[KP93] G. E. Karras and E. Petsa. DEM Matching and Detection of Deformation in Close-Range Photogrammetry without Control. *Photogrammetric Engineering & Remote Sensing*, 59(9):1419–1424, 1993.

[KTL$^+$04] D. Koller, M. Turitzin, M. Levoy, M. Tarini, G. Croccia, P. Cignoni, and R. Scopigno. Protected Interactive 3D Graphics via Remote Rendering. *ACM Transaction on Graphics*, 23(3):695–703, 2004.

[Kua96] S. Kuang. *Geodetic Network Analysis and Optimal Design: Concepts and Applications.* Ann Arbor Press, 1996.

[KWT88] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active Contour Models. *International Journal of Computer Vision*, 1(4):321–331, 1988.

[KY00] H. Kimura and Y. Yamaguchi. Detection of Landslide Areas Using Satellite RADAR Interferometry. *Photogrammetric Engineering & Remote Sensing*, 66(3):337–344, 2000.

[KYS08] T. Kurahashi, Y. Yajima, and Y. Sasaki. Landslide Disasters and Hazard Maps along National Highways in Japan. In *the 2nd East Asia Landslides Symposium*, 2008.

[LB09] T. Lovas and A. Berenyi. Laser Scanning in Deformation Measurements: Acquiring 3D Spatial Data of Bridges. *GIM International*, 23(3):17–21, 2009.

[LCPH02] M. A. Lefsky, W. B. Cohen, G. G. Parker, and D. J. Harding. LIDAR Remote Sensing for Ecosystem Studies. *BioScience*, 52(1):19–30, 2002.

[LCZD01] B. Li, R. Chellappa, Q. Zheng, and S. Der. Model-based Temporal Object Verification Using Video. *IEEE Transactions on Image Processing*, 10(6):897–908, 2001.

[LDHH11] M. J. Lato, M. S. Diederichs, D. J. Hutchinson, and R. Harrap. Evaluating Roadside Rockmasses for Rockfall Hazards Using LIDAR Data: Optimizing Data Collection and Processing Protocols. *Natural Hazards*, 60(3):831–864, 2011.

[Lem08] M. Lemmens. Leica Geosystems, HDS: fifth User Conference and HQ, San Ramon, USA. *GIM International*, 22(2):37–41, 2008.

[Lem10] M. Lemmens. Terrestrial Laser Scanners. *GIM International, August*, 2010.

[Lem11] M. Lemmens. Geo-information. *Forum American Bar Association*, pages 101–121, 2011.

[Lev99] M. Levoy. The Digital Michelangelo Project. In *Proceedings of Second International Conference on 3-D Digital Imaging and Modeling*, pages 2–11, 1999.

[LF05] V. Lepetit and P. Fua. *Monocular Model-Based 3D Tracking of Rigid Objects.* Foundations and Trends in Computer Graphics and Vision Series. 2005.

[LHN09] Y. Li, C. Huang, and R. Nevatia. Learning to Associate: HybridBoosted Multi-target Tracker for Crowded Scene. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2953–2960, 2009.

[LHRB05] O. Lateltin, C. Haemmig, H. Raetzo, and C. Bonnard. Landslide Risk Management in Switzerland. *Landslides*, 2(4):313–320, 2005.

[Lic07] Lichti D. Error Modelling, Calibration and Analysis of an AM-CW Terrestrial Laser Scanner System. *ISPRS Journal of Photogrammetry and Remote Sensing*, 61(5):307–324, 2007.

[Lic10] D. Lichti. Terrestrial Laser Scanner Self-calibration: Correlation Sources and their Mitigation. *ISPRS Journal of Photogrammetry and Remote Sensing*, 65:93–102, 2010.

[Lin01] L. Linsen. Point Cloud Representation. Technical report, Faculty of Computer Science, University of Karlsruhe, Germany, 2001.

[LIP⁺10] F. Leberl, A. Irschara, T. Pock, P. Meixner, M. Gruber, S. Scholz, and A. Wiechert. Point Clouds : LIDAR versus 3D Vision. *Photogrammetric Engineering & Remote Sensing*, 76(10):1123–1134, 2010.

[LLHY02] P. S. Lin, J. Y. Lin, H. C. Hung, and M. D. Yang. Assessing Debris Flow Hazard in a Watershed in Taiwan. *Engineering Geology*, 66:295–313, 2002.

[LLN11] Q. Liu, H. Li, and K. N. Ngan. Automatic Body Segmentation with Graph Cut and Self-adaptive Initialization Level Set (SAILS). *Journal Visual Communication and Image Representation*, 22(5):367–377, 2011.

[LN03] S. C. Lee and R. Nevatia. Interactive 3D Building Modeling Using a Hierarchical Representation. In *Proceedings of IEEE Workshop on Higher-Level Knowledge in 3D Modelling and Motion in conjunction with 9th International Conference on Computer Vision (ICCV)*, pages 58–65, 2003.

[Low04] D. G. Lowe. Distinctive Image Features from Scale-invariant Keypoints. *International Journal of Computer Vision*, 60(2):91–110, November 2004.

[LPR05] R. Lindenbergh, N. Pfeifer, and T. Rabbani. Accuracy Analysis of the Leica Hds3000 and Feasibility of Tunnel Deformation Monitoring. In *International Society Photogrammetry and Remote Sensing, WG III/3, III/4, V/3 workshop Laser scanning 2005*, pages 24–29, 2005.

[LSP08] H. Li, R. W. Sumner, and M. Pauly. Global Correspondence Optimization for Non-rigid Registration of Depth Scans. *Computer Graphics Forum*, 27(5), 2008.

[LTW⁺03] K. Lim, P. Treitz, M. Wulder, B. St-Onge, and M. Flood. LiDAR Remote Sensing of Forest Structure. *Progress in Physical Geography*, 27(1):88–106, March 2003.

[Lut02] J. A. Lutes. *Automated Dam Displacement Monitoring Using a Robotic Total Station.* Technical report, Department of Geodesy and Geomatics Engineering, University of New Brunswick. 2002.

[LX08] Y. Liu and Y. Xiong. Automatic Segmentation of Unorganized Noisy Point Clouds Based on the Gaussian Map. *Computer-Aided Design*, 40(5):576–594, 2008.

[Maa98] H.-G. Maas. Photogrammetric Techniques for Deformation Measurements on Reservoir Walls. In *Proceedings of the IAG Symposium on Geodesy for Geotechnical and Structural Engineering*, pages 319–324, 1998.

[MAH+99] J. E. Means, S. A. Acker, D. J. Harding, J. B. Blair, M. A. Lefsky, W. B. Cohen, M. E. Harmon, and W. A. McKee. Use of Large-footprint Scanning Airborne LIDAR to Estimate Forest Stand Characteristics in the Western Cascades of Oregon. *Remote Sensing of Environment*, 67(3):298–308, 1999.

[MBC+03] P. Mora, P. Baldi, G. Casula, M. Fabris, M. Ghirotti, E. Mazzini, and A. Pesci. Monitoring Landslide Movements Using GPS and Digital Photogrammetry. *Engineering Geology*, 68:103–121, 2003.

[MC08] O. Monserrat and M. Crosetto. Deformation Measurement Using Terrestrial Laser Scanning Data and Least Squares 3D Surface Matching. *ISPRS Journal of Photogrammetry and Remote Sensing*, 63(1):142–154, 2008.

[MCXL11] D. Mao, Y. Cao, J. Xu, and K. Li. Object Tracking Integrating Template Matching and Mean Shift Algorithm. In *International Conference on Multimedia Technology*, pages 3583–3586, 2011.

[MDH+02] S. Mazzotti, H. Dragert, R. D. Hyndman, M. M. Miller, and J. A. Henton. GPS Deformation in a Region of High Crustal Seismicity: N. Cascadia forearc. *Earth and Planetary Science Letters*, 198(1-2):41–48, 2002.

[MG03] P. J. Miranda and H. D. Granados. Fast Hazard Evaluation Employing Digital Photogrammetry: Popocatépetl glaciers, Mexico. *Geofisica Internacional*, 42(2):275–283, 2003.

[MH06] H.-g. Maas and U. Hampel. Photogrammetric Techniques in Civil Engineering Material Testing and Structure Monitoring. *Photogrammetric Engineering & Remote Sensing*, 72(1):39, 2006.

[MHG05] G. Metternicht, L. Hurni, and R. Gogu. Remote Sensing of Landslides: an Analysis of the Potential Contribution to Geo-spatial Systems for Hazard Assessment in Mountainous Environments. *Remote Sensing of Environment*, 98(2-3):284–303, 2005.

[MMB+08] P. E. Miller, J. P. Mills, S. L. Barr, M. Lim, D. Barber, G. Parkin, B. Clarke, S. Glendinning, and J. Hall. Terrestrial Laser Scanning for Assessing the Risk of Slope Instability along Transport Corridors. In *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, volume 37, pages 495–500, 2008.

[MMB11] P. E. Miller, J. P. Mills, and S. L. Barr. A Remote Sensing Approach for Landslide Hazard Assessment on Engineered Slopes. *Remote Sensing*, pages 1–9, 2011.

[MMC02] J.-P. Malet, O. Maquaire, and E. Calais. The Use of Global Positioning System Techniques for the Continuous Monitoring of Landslides: Application to the Super-Sauze Earthflow (Alpes-de-Haute-Provence, France). *Geomorphology*, 43(1-2):33–54, 2002.

[MMP99] J. L. Moss, W. J. McGuire, and D. Page. Ground Deformation Monitoring of a Potential Landslide at La Palma, Canary Islands. *Journal of Volcanology and Geothermal Research*, 94(1-4):251–265, 1999.

[MNT05] Y. Manolopoulos, A. Nanopoulos, and Y. Theodoridis. *Rtrees: Theory and Applications*. Springer, 2005.

[MPR$^+$09] G. Molnár, N. Pfeifer, C. Ressl, P. Dorninger, and C. Nothegger. Range Calibration of Terrestrial Laser Scanners with Piecewise Linear Functions. *Photogrammetrie, Fernerkundung, Geoinformation*, 1:9–21, 2009.

[MS05] K. Mikolajczyk and C. Schmid. A Performance Evaluation of Local Descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10):1615–1630, 2005.

[MSVW96] F. Mantovani, R. Soeters, and C. J. Van Westen. Remote Sensing Techniques for Landslide Studies and Hazard Zonation in Europe. *Geomorphology*, 15:213–225, 1996.

[Mur08] G. Murphy. Determining Stand Value and Log Product Yields Using Terrestrial LIDAR and Optimal Bucking: a Case Study. *Journal of Forestry*, 106(6):317–324, 2008.

[MVM09] F. P. Miller, A. F. Vandome, and J. McBrewster. *Glonass*. VDM Publishing House Ltd., 2009.

[Nak00] S.-I. Nakamura. GPS Measurement of Wind-induced Suspension Bridge Girder Displacements. *Journal of Structural Engineering*, 126(12):1413–1419, 2000.

[Nat81] National Research Council (U.S.). Panel on Crustal Movement Measurements. *Geodetic Monitoring of Tectonic Deformation: toward a Strategy*. National Academies, 1981.

[NDWB$^+$10] T. Nuttens, A. De Wulf, L. Bral, B. De Wit, L. Carlier, M. De Ryck, D. Constales, and H. De Backer. High Resolution Terrestrial Laser Scanning for Tunnel Deformation Measurements. In *FIG Congress: Facing the Challenges - Building the Capacity, Sydney, Australia*, 2010.

[NLH07] A. Nuchter, K. Lingemann, and J. Hertzberg. Cached Kd Tree Search for ICP Algorithms. In *3Dim'07: Sixth International Conference on 3-D Digital Imaging and Modeling*, number X, pages 419–426, 2007.

[NLSH12] J. Nurmi, S. Lohan, S. Sand, and H. Hurskainen. *Galileo Positioning Technology*. Signals and Communication Technology. 2012.

[NMRK98] R. Nagarajan, A. Mukherjee, A. Roy, and M. Khire. Temporal Remote Sensing Data and GIS Application in Landslide Hazard Zonation of Part of Western Ghat, India. *International Journal of Remote Sensing*, 19(4):573–585, 1998.

[NMV$^+$03] F. Nilforoushan, F. Masson, P. Vernant, C. Vigny, J. Martinod, M. Abbassi, H. Nankali, D. Hatzfeld, R. Bayer, F. Tavakoli, A. Ashtiani, E. Doerflinger, M. Daignieres, P. Collard, and J. Chery. GPS Network Monitors the Arabia-Eurasia Collision Deformation in Iran. *Journal of Geodesy*, 77(7-8):411–422, 2003.

[NPS06] A. Nickitopoulou, K. Protopsalti, and S. Stiros. Monitoring Dynamic and Quasi-Static Deformations of Large Flexible Engineering Structures with GPS: Accuracy, Limitations and Promises. *Engineering Structures*, 28(10):1471–1482, 2006.

[NSC06] P. Nillius, J. Sullivan, and S. Carlsson. Multi-target Tracking - Linking Identities Using Bayesian Network Inference. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 2187–2194, Washington, DC, USA, 2006.

[NT01] J. Neira and J. Tardos. Data Association in Stochastic Mapping Using the Joint Compatibility Test. *IEEE Transactions on Robotics and Automation*, 17(6):890–897, 2001.

[NZWJ09] X. Ning, X. Zhang, Y. Wang, and M. Jaeger. Segmentation of Architecture Shape Information from 3D Point Cloud. In *Proceedings of the 8th International Conference on Virtual Reality Continuum and its Applications in Industry*, volume 1, pages 127–132, 2009.

[OKCH10] M. J. Olsen, F. Kuester, B. J. Chang, and T. C. Hutchinson. Terrestrial Laser Scanning-based Structural Damage Assessment. *Journal of Computing in Civil Engineering*, 24(03):264–272, 2010.

[Pag04] M. Paganini. The Use of Space-borne Sensors for the Monitoring of Slope Instability: Present Potentialities and Future Opportunities. In *J.R. Centre (Ed.), Workshop on Risk Mitigation of Slope Instability, Institute for the Protection and Security of the Citizen*, 2004.

[PD00] N. Paragios and R. Deriche. Geodesic Active Contours and Level Sets for the Detection and Tracking of Moving Objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(3):266–280, 2000.

[PHVG02] P. Perez, C. Hue, J. Vermaak, and M. Gangnet. Color-based Probabilistic Tracking. In *Proceedings of the 7th European Conference on Computer Vision (ECCV)*, pages 661–675, 2002.

[PKA$^+$07] G. Pavlidis, A. Koutsoudis, F. Arnaoutoglou, V. Tsioukas, and C. Chamzas.

Methods for 3D Digitization of Cultural Heritage. *Journal of Cultural Heritage*, 8:93–98, 2007.

[Pop09] S. C. Popescu. *LIDAR: Remote Sensing of Terrestrial Environments.* Taylor and Francis, illustrate edition, 2009.

[PP83] H. Papo and A. Perelmuter. Reparametrization of Deformation Analysis. *Manuscripta Geodaetica*, 8:41–58, 1983.

[PP09] A. Prokop and H. Panholzer. Assessing the Capability of Terrestrial Laser Scanning for Monitoring Slow Moving Landslides. *Natural Hazards and Earth System Science*, 9(6):1921–1928, 2009.

[R9̈6] J. M. Rüeger. *Electronic Distance Measurement: an Introduction.* Springer, 4 edition, 1996.

[Rei79] D. Reid. An Algorithm for Tracking Multiple Targets. *IEEE Transactions on Automatic Control*, 24(6):843–854, 1979.

[Rem03] F. Remondino. From Point Cloud to Surface: the Modeling and Visualization Problem. In *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, International Workshop On Visualization and Animation of Reality-Based 3D Models*, volume 34, page 5, 2003.

[Rem06] F. Remondino. Image-Based 3D Modelling: a Review. *Photogrammetric Record*, 21(September):269–291, 2006.

[Res10] Y. Reshetyuk. A Unified Approach to Self-calibration of Terrestrial Laser Scanners. *ISPRS Journal of Photogrammetry and Remote Sensing*, 65(5):445–456, 2010.

[Riz02] V. Rizzo. GPS Monitoring and New Data on Slope Movements in the Maratea Valley (Potenza, Basilicata). *Physics and Chemistry of the Earth, Parts A/B/C*, 27(36):1535–1544, 2002.

[RKR$^+$08] C. Ruwwe, B. Keck, O. Rusch, U. Zolzer, and X. Loison. Image Registration by means of 3D Octree Correlation. In *IEEE 10th Workshop on Multimedia Signal Processing*, pages 515–519, 2008.

[Rob79] K. D. Robertson. The Use and Calibration of Distance Measuring Equipment for Precise Mensuration of Dams (Revised). Technical report, U.S. Army Engineer Topographic Laboratories, Fort Belvoir, Va, 1979.

[Rot04] H. Rott. Requirements and Applications of Satellite Techniques for Monitoring Slope Instability in Alpine Areas. In *Workshop on Risk Mitigation of Slope Instability, Jrc-Institute for the Protection and Security of the Citizen*, 2004.

[RRSJACBG02] J. J. Ramirez-Ruiz, H. Santiago-Jimenez, E. Alatorre-Chaez, and M. Breton-Gonzalez. EDM Deformation Monitoring of the 1997-2000 Activity at Volcan De Colima. *Journal of Volcanology and Geothermal Research*, 117(1-2):61–67, 2002.

[RS91] K. Rangarajan and M. Shah. Establishing Motion Correspondence. *CVGIP:*

Image Understanding, 54(1):56–73, 1991.

[RS05] J. Ray and K. Senior. Geodetic Techniques for Time and Frequency Comparisons Using GPS Phase and Code Measurements. *Metrologia*, 42:215–232, August 2005.

[RVDHV06] T. Rabbani, F. A. Van Den Heuvel, and G. Vosselmann. Segmentation of Point Clouds Using Smoothness Constraint. In *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, volume 36, pages 248–253, 2006.

[Sam10] A. Sampath. Segmentation and Reconstruction of Polyhedral Building Roofs from Aerial LIDAR Point Clouds. *IEEE Transactions on Geoscience and Remote Sensing*, 48(3):1554–1567, 2010.

[Sar07] S. Sarangi. *Surface Reconstruction from Unorganized Point Cloud Data Using Incremental Delaunay Triangulation.* State University of New York at Buffalo, 2007.

[Sav03] P. D. Savvaidis. Existing Landslide Monitoring Systems and Techniques. *School of Rural and Surveying Engineering, the Aristotle University of Thessaloniki, From Stars to Earth and Culture*, pages 242–258, 2003.

[SBT93] G. Stephen, J. Brownjohn, and C. Taylor. Measurements of Static and Dynamic Displacement from Visual Monitoring of the Humber Bridge. *Engineering Structures*, 15(3):197–208, 1993.

[SBW02] H. Schweitzer, J. W. Bell, and F. Wu. Very Fast Template Matching. In *Proceedings of the 7th European Conference on Computer Vision (ECCV)*, pages 358–372, 2002.

[Sch06] D. Schneider. Terrestrial Laser Scanning for Area Based Deformation Analysis of Towers and Water Dams. In *Proceedings of 3rd IAG/12th FIG Symposium*, 2006.

[Sch09] D. Schneider. Calibration of a Riegl LMS-Z420I based on a Multi-station Adjustment and a Geometric Model with Additional Parameters. *The International Archives of the Photogrammetry, Remote Sensing And Spatial Information Sciences, Part 3/W8*, pages 177–182, 2009.

[SD01] A. D. Sappa and M. Devy. Fast Range Image Segmentation by an Edge Detection Strategy. In *Proceedings of third International Conference on 3-D Digital Imaging and Modeling*, pages 292–299, 2001.

[SDA03] C. Squarzoni, C. Delacourt, and P. Allemand. Nine Years of Spatial and Temporal Evolution of the La Valette Landslide Observed by SAR Interferometry. *Engineering Geology*, 68:53–66, 2003.

[SDA05] C. Squarzoni, C. Delacourt, and P. Allemand. Differential Single-Frequency GPS Monitoring of the La Valette Landslide (French Alps). *Engineering Geology*, 79(3-4):215–229, 2005.

[SED07] M. Strand, F. Erb, and R. Dillmann. Range Image Registration Using an Octree Based Matching Strategy. In *International Conference on Mechatronics and Automation, 2007*, pages 1622–1627, 2007.

[See01] A. A. Seemkooei. Strategy for Designing Geodetic Network with High Reliability and Geometrical Strength. *Journal of Surveying Engineering*, 127(3):104–117, 2001.

[SF90] M. R. Shortis and C. S. Fraser. A Review of Close Range Optical 3D Measurement. In *Procedings of 16th National Surveying Conference*, page 10, 1990.

[SFTA08] A. Suga, K. Fukuda, T. Takiguchi, and Y. Ariki. Object Recognition and Segmentation Using SIFT and Graph Cuts. In *19th International Conference on Pattern Recognition*, pages 1–4, December 2008.

[She08] Y. Sheng. Quantifying the Size of a LIDAR Footprint: a Set of Generalized Equations. *IEEE Geoscience and Remote Sensing Letters*, 5(3):419–422, 2008.

[SHF$^+$06] H. P. Sato, H. Hasegawa, S. Fujiwara, M. Tobita, M. Koarai, H. Une, and J. Iwahashi. Interpretation of Landslide Distribution Triggered by the 2005 Northern Pakistan Earthquake Using Spot 5 Imagery. *Landslides*, 4(2):113–122, 2006.

[SI04] T. Schulz and H. Ingensand. Terrestrial Laser Scanning - Investigations and Applications for High Precision Scanning. In *FIG Working Week*, 2004.

[Sim96] D. A. Simon. *Fast and Accurate Shape-based Registration.* PhD thesis, Carnegie Mellon University, 1996.

[Sin02] V. Singhroy. Landslide Hazards: Ceos, the Use of Earth Observing Satellites for Hazard Support: Assessments and Scenarios. Technical report, Final Report of the Ceos Disaster Management Support Group, NOAA, 2002.

[SJ87] I. K. Sethi and R. Jain. Finding Trajectories of Feature Points in a Monocular Image Sequence. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(1):56–73, 1987.

[SKMG04] D. Serby, E. Koller-Meier, and L. V. Gool. Probabilistic Object Tracking Using Multiple Features. In *Proceedings of 17th International Conference on Pattern Recognition (ICPR)*, volume 2, pages 184–187, Washington DC, USA, 2004.

[SKW$^+$08] T. Strozzi, A. Kouraev, A. Wiesmann, U. WegmÜLler, A. Sharov, and C. Werner. Estimation of Arctic Glacier Motion with Satellite L-band SAR Data. *Remote Sensing of Environment*, 112(3):636–645, 2008.

[SM86] M. J. Sevilla and M. D. Martin. Geodetic Network Design for Crustal Deformation Studies in the Caldera of Teide Area. *Tectonophysics*, 130(1-4):235–248, 1986.

[Smi02] L. C. Smith. Emerging Applications of Interferometric Synthetic Aperture Radar (INSAR) in Geomorphology and Hydrology. *Annals of the Association of American Geographers*, 92(3):385–398, 2002.

[SPT$^+$03] E. Stylianidis, P. Patias, V. Tsioukas, L. Sechidis, and C. Georgiadis. A Digital Close-range Photogrammetric Technique for Monitoring Slope Displacements. In *Proceedings of 11th FIG Symposium on Deformation Measurements*, 2003.

[SRPR08] S. Sutor, R. Rohr, G. Pujolle, and R. Reda. Efficient Mean Shift Clustering Using Exponential Integral Kernels. 2008.

[SS01] H. Setan and R. Singh. Deformation Analysis of a Geodetic Monitoring Network. *Geomatica*, 55(3):333–346, 2001.

[SSV07] J. Sankaranarayanan, H. Samet, and A. Varshney. A Fast All Nearest Neighbor Algorithm for Applications Involving Large Point-Clouds. *Computers & Graphics*, 31(2):157–174, 2007.

[ST94] J. Shi and C. Tomasi. Good Features to Track. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 593–600, June 1994.

[Sta03] R. Staiger. Terrestrial Laser Scanning Technology, Systems and Applications. In *2nd FIG Regional Conference*, 2003.

[Ste06] H. Sternberg. Deformation Measurements at Historical Buildings with Terrestrial Laser Scanners. In *International Archives of Photogrammetry and Remote Sensing*, volume xxxvi, 2006.

[STH80] C. C. Slama, C. Theurer, and S. W. Henriksen, editors. *Manual of Photogrammetry.* 1980.

[SVW96] R. Soeters and C. J. Van Westen. Slope Instability Recognition, Analysis and Zonation. In *Turner, K., Schuster, R.L. (Eds.), Landslides Investigation and Mitigation, Transportation Research Board, Special Report 247, National Research Counci*, pages 129–177. 1996.

[SWK07] R. Schnabel, R. Wahl, and R. Klein. Efficient RANSAC for Point-Cloud Shape Detection. *Computer Graphics Forum*, 26(2):214–226, 2007.

[SWKZ04] T. Schafer, T. Weber, P. Kyrinovic, and M. Zamencnikova. Deformation Measurement Using Terrestrial Laser Scanning at the Hydropower Station of Gabcikovo. In *INGEO and FIG Regional Central and Eastern European Conference on Engineering Surveying*, 2004.

[Sze96] R. Szeliski. Matching 3-D Anatomical Surfaces with Non-Rigid Deformations Using Octree-Splines. *International Journal of Computer Vision*, 18(2):171–186, 1996.

[Tan09] P. Tang. *Extraction of Surveying Goals from Point Clouds Obtained from Laser Scanners to Support Bridge Inspection.* PhD thesis, Carnegie Mellon University, 2009.

[TGZG07] G. Teza, A. Galgaro, N. Zaltron, and R. Genevois. Terrestrial Laser Scanner to Detect Landslide Displacement Fields: a New Approach. *International Journal of Remote Sensing*, 28(16):3425–3446, 2007.

[TLP06] M. Tsakiri, D. Lichti, and N. Pfeifer. Terrestrial Laser Scanning for Deformation Monitoring. In *Proceedings of 3rd IAG/12th FIG Symposium*, 2006.

[TOD+08] J. Travelletti, T. Oppikofer, C. Delacourt, J. P. Malet, and M. Jaboyedoff. Monitoring Landslide Displacements during a Controlled Rain Experiment Using a Long-Range Terrestrial Laser Scanning (TLS). In *International Archives of Photogrammetry and Remote Sensing*, volume 37, pages 485–490, 2008.

[Tor01] W. Torge. *Geodesy.* Walter de Gruyter, 3, illustrated edition, 2001.

[TPG08] G. Teza, A. Pesci, and A. Galgaro. Grid_strain and Grid_strain3: Software Packages for Strain Field Computation in 2D and 3D Environments. *Computers & Geosciences*, 34(9):1142–1153, 2008.

[TPGG08] G. Teza, A. Pesci, R. Genevois, and A. Galgaro. Characterization of Landslide Ground Surface Kinematics from Terrestrial Laser Scanning and Strain Field Computation. *Geomorphology*, 97(3-4):424–437, 2008.

[Tri12] Trimble. Trimble S8 Datasheets, 2012.

[U.S02] U.S. Army Corps of Engineers. Engineering and Design-Structural Deformation Surveying. Technical report, Department of the Army, U.S. Army Corps of Engineers, 2002.

[Var78] D. J. Varnes. Slope Movement Types and Processes. In *Landslides Analysis and Control, Specific Report, Transportation Research Board Special Report*, volume 176, pages 11–33, 1978.

[Ver02] M. Vermeer. Review of the GPS Deformation Monitoring Studies. *Radiation and Nuclear Safety Authority*, 2002.

[VLL09] B. Vendeville, A. Lacoste, and L. Loncke. Role of Fluid Overpressure and River Incision on Landslides: Field Work and Experimental Modeling. In *American Geophysical Union, Fall Meeting*, 2009.

[VRB01] C. J. Veenman, M. Reinders, and E. Backer. Resolving Motion Correspondence for Densely Moving Points. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23:54–72, 2001.

[Wan02] J. Wang. Pseudolite Applications in Positioning and Navigation: Progress and Problems. *Journal of Global Positioning Systems*, 1:48–56, 2002.

[Wan11] M. Wang. *Segmentation of LIDAR Point Clouds.* PhD thesis, National Cheng Kung University, 2011.

[WB94] M. Wani and B. G. Batchelor. Edge-region-based Segmentation of Range Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(3):314–319, 1994.

[WGM12] J. Wu, P.-Y. Gillieron, and B. Merminod. Cell-based Automatic Deformation Computation by Analyzing Terrestrial LIDAR Point Clouds. *Photogrammetric Engineering & Remote Sensing*, 78(4):317–329, 2012.

[WH01] W. M. Welsch and O. Heunecke. Models and Terminology for the Analysis of

Geodetic Monitoring Observations. Technical report, FIG Working Group 6.1, 2001.

[WHBD08] W. Wagner, M. Hollaus, C. Briese, and V. Ducic. 3D Vegetation Mapping Using Small-footprint Full-waveform Airborne Laser Scanners. *International Journal of Remote Sensing*, 29(5):1433–1452, 2008.

[WKW$^+$02] H. Woo, E. Kang, S. Wang, K. Wan, and H. Lee. A New Segmentation Method for Point Cloud Data. *International Journal of Machine Tools and Manufacture*, 42(2):167–178, 2002.

[WP97] W. S. Whyte and R. E. Paul. *Basic Surveying.* Elsevier Science & Technology, 4, illustrated edition, 1997.

[WS09a] J. Wang and J. Shan. Segmentation of LIDAR Point Clouds for Building Extraction. In *Procedings of American Society Photogrammetry, Remote Sensing, Annual Conference*, pages 9–13, 2009.

[WS09b] G. Wieczorek and J. B. Snyder. Monitoring Slope Movements. *Geological Monitoring: Boulder, Colorado, Geological Society of America*, pages 245–271, 2009.

[Yas07] N. Yastikli. Documentation of Cultural Heritage Using Digital Photogrammetry and Laser Scanning. *Journal of Cultural Heritage*, 8:423–427, 2007.

[YD96] Y. Yacoob and L. Davis. Recognizing Human Facial Expressions from Long Image Sequences Using Optical Flow. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(6):636–642, 1996.

[YJS06] A. Yilmaz, O. Javed, and M. Shah. Object Tracking: a Survey. *ACM Computing Surveys*, 38(4), 2006.

[YL89] N. Yokoya and M. D. Levine. Range Image Segmentation based on Differential Geometry: a Hybrid Approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(6):643–649, 1989.

[YL99] M. Yang and E. Lee. Segmentation of Measured Point Data Using a Parametric Quadric Surface Approximation. *Computer Aided Design*, 31(7):449–457, 1999.

[YLS04] A. Yilmaz, X. Li, and M. Shah. Contour-based Object Tracking with Occlusion Handling in Video Acquired Using Mobile Cameras. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(11):1531–1536, 2004.

[Zar91] J. Zarzycki. *Glasses and the Vitreous State.* Cambridge Solid State Science Series, Eds. Clarke, D.R., Et Al. 1991.

[ZBB$^+$06] A. Zanutta, P. Baldi, G. Bitelli, M. Cardinali, and A. Carrara. Qualitative and Quantitative Photogrammetric Techniques for Multi-temporal Landslide Analysis. *Annals of Geophysics*, 49:1121–1134, 2006.

[ZCF07] C. Zhang, M. P. Crane, and C. S. Fraser. Terrain Deformation Modeling by Photogrammetric Exploitation of High Resolution Satellite Imagery. In *Proceeding of ASPRS Annual Conference*, 2007.

[ZEC+03] D. A. Zimble, D. L. Evans, G. C. Carlson, R. C. Parker, S. C. Grado, and P. D. Gerard. Characterizing Vertical Forest Structure Using Small-footprint Airborne LIDAR. *Remote Sensing of Environment*, 87:171 – 182, 2003.

[Zen12] D. Zeng. *Advances in Computer Science and Engineering.* Advances in Intelligent and Soft Computing. Springer-Verlag Gmbh, 2012.

[Zha94] Z. Zhang. Iterative Point Matching for Registration of Free-form Curves and Surfaces. *International Journal of Computer Vision*, 13(2), 1994.

[ZLLX02] C. Zhou, C. Lee, J. Li, and Z. Xu. On the Spatial Relationship between Landslides and Causative Factors on Lantau Island, Hong Kong. *Geomorphology*, 43:197–207, 2002.

[ZZ97] D. Zhao and X. Zhang. Range-data-based Object Surface Segmentation via Edges and Critical Points. *IEEE Transactions on Image Processing*, 6(6):826–830, 1997.

Curriculum Vitae

Jing WU
EPFL ENAC IIE TOPO
Lausanne
CH-1015
Switzerland

`http://people.epfl.ch/jing.wu`
eMail: jing.wu@epfl.ch
wujing346@gmail.com
Nationality: Chinese

RESEARCH INTERESTS

- Environmental Monitoring, Deformation Monitoring, Natural Hazard Assessment
- Terrestrial LiDAR, GNSS, Photogrammetry, Remote Sensing
- Point Cloud Processing, Geographic/Spatial Data Analysis

EDUCATIONAL BACKGROUND

- *Ph.D. in Computer Science* (2008.07–2012.06)
  Swiss Federal Institute of Technology - Lausanne (EPFL), Switzerland
  Thesis: Cell-based Deformation Monitoring via 3D Point Clouds

- *M.Sc. in Geomatics Engineering* (2005.09–2008.06)
  College of Geomatics (CG)
  Shandong University of Science and Technology (SDUST), Qingdao, China
  Thesis: Performance Evaluation of 3D Laser Scanner and Its Applications

- *B.Sc. in Geomatics Engineering* (2001.09–2005.06)
  College of Geomatics (CG)
  Shandong University of Science and Technology (SDUST), Qingdao, China
  Thesis: Study on Ground Subsidence using InSAR Images

PROFESSIONAL EXPERIENCES

- *Junior Researcher at EPFL, Switzerland* (2008.07–2012.06)
  Geodetic Engineering Laboratory (TOPO)
  Topics: "Deformation modelling", "Cell-based deformation computation via point clouds"

BIBLIOGRAPHY

- *Research Assistant at Wuhan University, China* (2006.07–2007.04)
  School of Geodesy and Geomatics
  Topic: “Point clouds collection by terrestrial laser scanner”, “Point cloud analysis”

- *Research Assistant at SDUST, China* (2005.09–2008.06)
  Lab of Geodetic Measurement
  Topic: “3D laser scanner calibration, Data acquisition and reconstruction”

TEACHING EXPERIENCES

- *Geodetic Engineering, by Prof. Bertrand Merminod* (2009, 2010, 2012)
  First Teaching Assistant, 4 credits, 2012 Spring semester
  First Teaching Assistant, 4 credits, 2010 Spring semester
  First Teaching Assistant, 4 credits, 2009 Spring semester

- *Quantitative Methods I, by Prof. Bertrand Merminod* (2008, 2009, 2010, 2011)
  Teaching Assistant, 4 credits, 2011 Autumn Semester
  First Teaching Assistant, 5 credits, 2010 Autumn Semester
  First Teaching Assistant, 5 credits, 2009 Autumn Semester
  Teaching Assistant, 4 credits, 2008 Autumn Semester

- *Survey and Mapping of the Built Environment, by Pierre-Yves Gilliéron* (2009, 2010)
  First Teaching Assistant, 4 credits, 2010 Spring semester
  First Teaching Assistant, 4 credits, 2009 Spring semester

MAIN PUBLICATIONS

(1) Journal Publications

Jing Wu, Pierre-yves Gilliéron, Bertrand Merminod. Cell-based Automatic Deformation Computation from Terrestrial LIDAR Point Clouds, *Photogrammetric Engineering & Remote Sensing (PE&RS)*, 78(4):317-329, 2012.

(2) Conference Publications

Jing Wu, Pierre-yves Gilliéron, Bertrand Merminod. A Split/Merge Approach for Landslide Monitoring via Terrestrial Laser Scanning, *the 3rd International Conference of Earth Observation of Global Changes (EOGC)*, Munich, Germany, April 13-15, 2011.

Jing Wu, Pierre-yves Gilliéron, Bertrand Merminod. An Integrated Methodology for Change Detection via Terrestrial Laserscanning: Case Study of the Landslide in Flamatt, *the 8th Swiss Geoscience Meeting*, Fribourg Switzerland, November 19-20, 2010.

Jing Wu, Bertrand Merminod. A Split/Merge Approach for Change Detection via Terrestrial LIDAR, *Workshop on Remote Sensing Methods for Change Detection and Process Modeling*, Cologne, Germany, November 18-19, 2010.

(3) Previous Journal Publications

Jing Wu, Fengxiang Jin, Jian Wang. Three Dimension Modelling of Buildings on 3D Laser Scanner Data, *Engineering of Surveying and Mapping*, 16(5):57-60, 2007 (in Chinese).

HONOURS

- *Best Master Thesis Award* (2008)
  Granted by Shandong University of Science and Technology
- *Excellent Graduate Student* (2005)
  Granted by Shandong Province Education Commission
- *Excellent Student Scholarship* (2002, 2003, 2004)
  Granted by Shandong University of Science and Technology
- *The Second Prize of the 3rd Symposium of Science & Technique* (2003)
  Granted by College of Geomatics, SDUST