A hierarchical approach for near real-time 3D surface change analysis of permanent laser scanning point clouds

Ronald TABERNIG[1,2,*], **William ALBERT**[1], **Hannah WEISER**[1,2], and **Bernhard HÖFLE**[1,2]

[1] *Heidelberg University, Institute of Geography, 3DGeo Research Group, Heidelberg, Germany, (ronald.tabernig@uni-heidelberg.de)*

[2] *Interdisciplinary Center of Scientific Computing (IWR), Heidelberg University, Heidelberg, Germany*

[*] *corresponding author*

Abstract

Modern permanently installed laser scanning systems (PLS) allow capturing point clouds in short intervals (e.g., sub-hourly), bringing us closer to the early detection of small surface changes that may precede larger events. Predicting potential hazards necessitates near real-time surface change computation. This requires reliable and efficient methods that can be operated directly on laser scanners in the future. We propose a method that combines low-resolution (meters) change detection with high-resolution (centimeters) change analysis. First, utilizing the Mahalanobis distance, a change detection approach identifies significant intra-voxel changes, filtering out temporary changes (e.g., tree movements due to wind) to retain only persistent, relevant changes (e.g., rock movements). Second, sub-point clouds of areas exhibiting significant change are extracted and subjected to point cloud-based surface change analysis. Hierarchical analysis of point clouds with fine-tuned key parameters results in a data volume reduction of over 95% and a miss rate of less than 6%, both relative to a manually annotated reference point cloud. Furthermore, a computation time decrease of 97% is achieved relative to an M3C2-only run. Our approach is based on the hierarchical detection and analysis of areas exhibiting surface change. This method is particularly efficient when these areas are considerably smaller than the monitored area, allowing processing within seconds and much faster than data acquisition. A further advantage is that this methodology is implemented using the open-source Python libraries *py4dgeo* and *VAPC*, which enables straightforward integration into your own PLS monitoring workflows, allowing processing much faster than data acquisition (e.g. within seconds).

Keywords: Hierarchical analysis, Permanent laser scanning, Change detection, Change analysis

Received: 3rd December 2024. Revised: 14th March 2025. Accepted: 14th March 2025.

1 Introduction

Deformation monitoring through repeated LiDAR acquisitions is a valuable technique for change detection in outdoor settings. The analysis of surface deformations, such as gravitational mass movements (Jaboyedoff et al., 2012), can be done due to densely sampled surfaces at resolutions ranging from sub-centimeter to decimeter scales, depending on the distance to the target. Modern laser scanning systems (Kromer et al., 2017; Czerwonka-Schröder et al., 2022) are furthermore capable of acquiring data with high temporal resolution, e.g., at sub-hourly intervals. This enables the detection of small surface changes that may precede larger events and could hence trigger early warning systems for natural hazards. Research on these 4D (3D+time) datasets demonstrates that pre-failure events in rockfall monitoring settings can be detected through the temporal domain (Williams et al., 2019).

Much research has been conducted on the detailed analysis of bi-temporal point clouds (Qin et al., 2016). Current methods for 4D analysis are still evolving and are inherently computationally expensive and time-consuming (Anders et al., 2020). Permanent laser scanning (PLS) processing pipelines typically analyze data retrospectively or rely on a stable internet connection to transfer data to high-end hardware for analysis (Czerwonka-Schröder et al., 2022). As continuous data acquisition becomes increasingly feasible, rapid on-device computations will be required to detect surface change within billions of points in a short period of time. This is especially important for monitoring critical infrastructure, where timely data analysis can substantially impact decision-making. To the best of our knowledge, there is currently no method that combines near real-time change detection with detailed change analysis (Wellhausen et al., 2017; Gehrung et al., 2019; Fahle et al., 2023).

We address this gap with a two-stage hierarchical approach that extends state-of-the-art PLS processing pipelines. The initial stage of this framework is the data minimization phase, which reduces the point cloud data to areas where persistent surface change (e.g., rockfall) may have occurred by coarse change detection. This is accomplished using the open-source *VAPC* library (Tabernig et al., 2024), which enables voxel-based identification of notable discrepancies between bi-temporal point clouds. In the second step, the change analysis step, we perform point cloud-based change analysis in regions established during the data minimization phase. This step provides detailed information on the actual change magnitudes and directions. For this, in this study, we use the M3C2 algorithm (Lague et al., 2013) implemented in the *py4dgeo* library (py4dgeo Development Core Team, 2023).

Decision making in the context of natural hazard monitoring relies on low computation times and high completeness, which typically represents a trade-off. We aim to find a Pareto-optimal solution that reduces data volume and computation time while minimizing the number of missed points that contain information about actual surface change. To assess the reliability of the proposed workflow, we employ both data from a real rockfall event and synthetic LiDAR data (Weiser and Höfle, 2024) generated by simulating a rockfall.

2 Data Description

To enable the detection of persistent change, we calibrate our method using virtual laser scanning (VLS) point clouds of simulated rockfall events to determine the most effective parameters. We develop four simulation scenarios to evaluate the method's performance in various contexts. Parameter combinations resulting in the smallest miss rates (false negative rate/FNR) at the simulated data are then applied to the real data.

2.1 Real data

The test site is located in Trier, Germany, and it is monitored hourly using a permanently installed laser scanner (RIEGL VZ-2000i) (Czerwonka-Schröder et al., 2022). Scans are acquired with a resolution of 15 mdeg and a pulse repetition rate of 50 kHz. On the morning of August 26, 2024, a rockfall (ca. 150 m³) occurred. A rockfall net successfully stopped rock fragments that would otherwise have reached the road or railway. In this study, we use data from the epoch before the rockfall and the epoch following it, with a one-hour interval between the point clouds. To facilitate the analysis, we focus exclusively on bi-temporal data and illustrate the impact this approach has on point cloud time series.

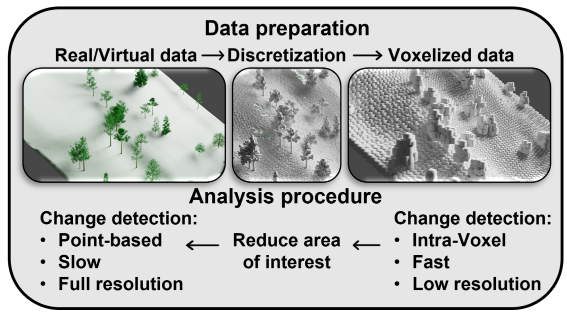

Figure 1. Illustration of the proposed hierarchical analysis for point clouds. Top row (left to right) shows data preparation steps, including voxelization of point clouds. Bottom row (right to left) depicts data analysis steps: intra-voxel change detection, 3D masking, and point cloud-based change analysis on segmented regions.

2.2 Simulated PLS data

To reliably calibrate our change detection method, we generate simulated rockfall events through a three-stage process: creation of a digital twin (DT), introduction of dynamics, and VLS (Weiser and Höfle, 2024).

Creation of rockfall digital twin: We construct a DT of the study site to replicate the real-world environment. This involves generating a high-resolution 3D mesh using the Poisson reconstruction algorithm (Kazhdan et al., 2006) in CloudCompare (Girardeau-Montaut, 2024), adding volume to the surface features using Blender (Blender Online Community, 2024), and integrating vegetation movement and rockfall events. Vegetation movement is implemented in the scene by adding wind-affected trees. The purpose is to calibrate the change detection algorithm to detect changes that are relevant to us (rockfall) and filter out those that are not (wind-affected trees). Seven tree models, representing the structures observed at the actual study site, are generated using Blender's *Sapling Tree Gen* add-on (Weber and Penn, 1995). Wind-induced movements are simulated by applying Sapling Tree Gen's built-in wind animation to introduce sufficient movement to challenge the change detection algorithm. The temporal variability is covered by replacing each tree model representation with different wind-induced states for each simulation run. For the rockfall simulation, the extruded mesh is subjected to fracturing using Blender's Cell Fracture add-on (Barton et al., 2024). Selected areas are intentionally over-fractured to generate small rock fragments, simulating rockfall debris. To ensure that even smaller events are detected, we cover rock fragment volumes of a range of sizes smaller than those from the event at the study site (Tab. 1). The simulation, based on Blender's built-in physics, achieves run-out lengths consistent with those observed at the real study site.

VLS of rockfall digital twin: To generate realistic point cloud data from the dynamic DT, we simulate a PLS setup replicating the real-world configuration using HELIOS++ (Winiwarter et al., 2022). The virtual scanner is configured to emulate the RIEGL VZ-2000i specifications, resulting in identical vertical and horizontal resolutions as the actual scanner deployed at the study site. Scanning is conducted before and after the simulated rockfall and vegetation dynamics. Through this process, we generate five distinct point clouds: one representing the unchanged base scene, three capturing scenarios (S1, S2, and S3) with both surface changes and wind dynamics (Tab. 1), and one scenario reflecting wind dynamics without any rockfall associated surface changes. By replicating the key components and dynamics of the real study site, the DT ensures the method's reliability and effectiveness when applied to real-world data.

3 Methods

3.1 Hierarchical change analysis

Hierarchical analysis is essential to preliminarily identify areas with potential surface changes. It adds fast and robust change detection before performing a comprehensive surface change analysis

Table 1. Volume of rockfall fragments (m³) across simulated scenarios (S), representing a range of smaller volumes compared to the real event.

Metric	S1	S2	S3
Fragments	9	1	30
Mean (m³)	0.879	3.006	1.772
Std Dev (m³)	1.421	0.000	1.958
Min (m³)	0.029	3.006	0.021
Max (m³)	4.536	3.006	8.091
Sum (m³)	7.908	3.006	53.167

only on unstable areas. Our workflow consists of three main steps: 1) Two point clouds are voxelized and in each voxel the point distribution is described statistically. By this, significant discrepancies between the two datasets are identified. 2) Areas exhibiting notable voxel differences are determined through statistical testing and are extracted from the original point clouds using a three-dimensional mask. 3) Surface change analysis is conducted on the extracted clusters to assess the detailed changes between the point clouds (Fig. 1). This hierarchical approach enables time-efficient analysis of massive point clouds, which is critical for time-sensitive applications such as natural hazards monitoring.

3.1.1 Detection of persistent change using voxelized point clouds

Our approach detects persistent changes (e.g., rockfall) while filtering out temporary variations (e.g., wind-induced tree movement) by analyzing the spatial distribution of points within individual voxels. Voxels containing vegetation exhibit a wide, irregular, and dynamically varying distribution of points, whereas those representing stable rock surfaces display narrower distributions. Similar to Fahle et al. (2023), we quantify this variability using the Mahalanobis distance (Mahalanobis, 1936), which measures a point's deviation from the local voxel distribution in units of standard deviation. Consequently, a point at an equivalent Euclidean distance from the voxel center yields fewer standard deviations in vegetated voxels than in rock-face voxels due to the broader spread of points.

We begin this step by voxelizing each point cloud using a user-defined voxel size, which must be large enough to enhance computation speed but small enough to accurately capture the essential structural details of the point cloud. Voxels occupied in only one of the two epochs are considered to have undergone significant change. For meaningful hypothesis testing, we do not consider voxels that are occupied by less than 30 points. Voxels occupied in both epochs ($t1$ and $t2$) are further investigated. For each voxel pair, we compute the voxels' centers of gravity (μ_{t1} and μ_{t2}) and covariance matrices (S_{t1} and S_{t2}). Using these parameters, we compute the Mahalanobis distance (D) (Mahalanobis, 1936). We compute this distance twice: once from the distribution of the voxel at $t1$ to μ_{t2}, resulting in D_1, and once from the distribution of the voxel at $t2$ to μ_{t1}, resulting in D_2. To maximize the likelihood of detecting significant differences, we proceed with the larger Mahalanobis distance (D_M) for further analysis.

$$D_1(\mu_1, \mu_2) = \sqrt{(\mu_1 - \mu_2)^T S_2^{-1} (\mu_1 - \mu_2)} \quad (1)$$

$$D_2(\mu_2, \mu_1) = \sqrt{(\mu_2 - \mu_1)^T S_1^{-1} (\mu_2 - \mu_1)} \quad (2)$$

$$D_M = \max(D_1, D_2) \quad (3)$$

The computed Mahalanobis distance (D_M) is compared against the chi-squared distribution to determine the p-value for each voxel:

$$p\text{-value} = 1 - \text{CDF}(\chi^2, D_M^2) \quad (4)$$

Where CDF is the cumulative distribution function. In order to detect outliers, a significance level (α) is determined through optimization. If the p-value is below α, we consider the difference between the compared voxels to be statistically significant. In addition to the voxel size, α is the second input parameter of our method.

To facilitate voxel-based operations, we developed the open-source Python library *VAPC* (Tabernig et al., 2024), which enables users to perform hierarchical change analysis efficiently, and to plug in their own deformation analysis method in a hierarchical workflow. Furthermore, *VAPC* enables quick comparison of own methods with baseline state-of-the-art methods such as M3C2 and variants thereof.

3.1.2 Masking unstable areas

Voxels exhibiting significant differences are identified and merged. These merged voxels serve as three-dimensional masks to extract regions showing changes in both point clouds. To ensure that we extract sufficient points for consecutive change analysis, we take into account the maximum normal radius used in the M3C2 computation. Specifically, points within the mask's point cloud are voxelized using a voxel size equal to half of the maximum normal radius we want to use in the M3C2. Subsequently, the mask is expanded by incorporating all 26 neighboring voxels of each voxel, regardless of whether they are detected as significantly changed. This expansion process guarantees that normals are adequately computed at the edges of the point cloud segments.

3.1.3 Change analysis using extracted point clouds

The change analysis step can include any custom change analysis method. We choose to use the M3C2 algorithm (Lague et al., 2013) as a consistent benchmark because it is a well-established method suitable for the process we are monitoring. More specifically, we analyze the extracted point clouds for surface changes using the *py4dgeo* library (py4dgeo Development Core Team, 2023). The same M3C2 settings are used for all computations, which allows a fair comparison of computation times. We use a single normal radius of 1 m, a cylinder radius of 1 m, and a maximum distance of 20 m.

Our hierarchical analysis method requires two parameters: voxel size and the designated significance level α to identify significant surface changes in the voxel space. These parameters strongly influence the result and must be selected based on extensive testing. To better understand their impact and relation, we conduct a thorough Pareto analysis (Sect. 3.2).

3.2 Parameter selection using Pareto Analysis

Pareto analysis is a method for determining tradeoffs between competing objectives in situations where a single solution is not attainable. This involves evaluating each potential solution against all others. A solution is considered non-dominated if no other solution exists that matches or exceeds its performance in every objective while offering an improvement in at least one. This approach identifies a set of balanced trade-off options, where any improvement in one objective would lead to a compromise in another (Deb, 2001). The collection of these non-dominated solutions is referred to as the Pareto frontier, representing the boundary of tradeoffs in the objective space. Seppelt et al. (2013) successfully applied Pareto analysis to identify trade-offs between ecosystem services, land use, and biodiversity for land management. Inspired by this approach, our study addresses the trade-off between two critical requirements in rockfall monitoring: rapid data analysis and low FNR. Rapid data analysis supports timely decision making, and a low FNR is critical to avoid missing important events. For example, our voxel-based filtering step aims to reduce data volume and thus computation time as much as possible. If the filtering step is too strict, too many points will be removed, increasing the FNR. Conversely, if the filtering step is too mild, a lower FNR can be maintained, but the reduction in computation time will be less. To objectively handle this trade-off, we apply Pareto analysis considering two explicit criteria: (1) the False Negative Rate (FNR), and (2) the relative computation time (RCT), directly correlated with data volume reduction (DVR). Specifically, our approach explores different combinations of voxel sizes and alpha parameters. We find the Pareto frontier by evaluating the result of each distinct combination regarding these two objectives. The resulting Pareto frontier defines the parameter combinations for which no other combination yields simultaneously lower FNR and lower RCT. The Pareto-frontier provides a transparent set of solutions, from which practitioners can choose based on specific operational priorities, either prioritizing accuracy (low FNR) or computational efficiency (low RCT). Thus, the final selection of parameters remains a subjective decision, guided by the specific demands of the rockfall monitoring task at hand.

3.3 Evaluation

We use a two-step approach to determine parameter combinations for the detection of persistent changes in our study site. These steps are performed once at the beginning and the selected parameters can then be applied to a time series of arbitrary length. First, we employ a grid search on simulated scenarios to identify parameter combinations that meet the minimum requirements. Second, we evaluate these parameter combinations using the manually annotated real point cloud.

For the first step, we compute the FNR for 180 combinations per simulated scenario. Voxel sizes ranging from 1 m to 10 m with a 1 m step size, from 10 m to 20 m with a 2 m step size, and significance levels between 0.1 and 0.99999 are selected. For each parameter combination, we calculate the mean FNR across all scenarios. Combinations with an FNR exceeding 0.2 are excluded, as these values are inadequate for further analysis.

To evaluate the approach, a point cloud containing a rockfall event from the study site is manually classified as either "change" or "no change", serving as the reference for quantifying the FNR, DVR, and computation time. Based on the results on this dataset, we obtain the Pareto-efficient solutions, from which we select the parameter combination we qualitatively present the results in (Section 4.2). Regarding computation time, we are interested in the relative computation time (RCT), defined as the processing time for the full hierarchical analysis (voxel-based detection of persistent change + change analysis on masked subset) relative to the time required to execute full-resolution change analysis (M3C2) on the entire area. Finally, the effectiveness of the Pareto frontier is compared to the non-hierarchical results.

4 Results

4.1 Pareto frontier analysis

Solutions on the Pareto frontier, which represent balanced trade-offs, indicate that data volume can be reduced by more than 90% while maintaining an

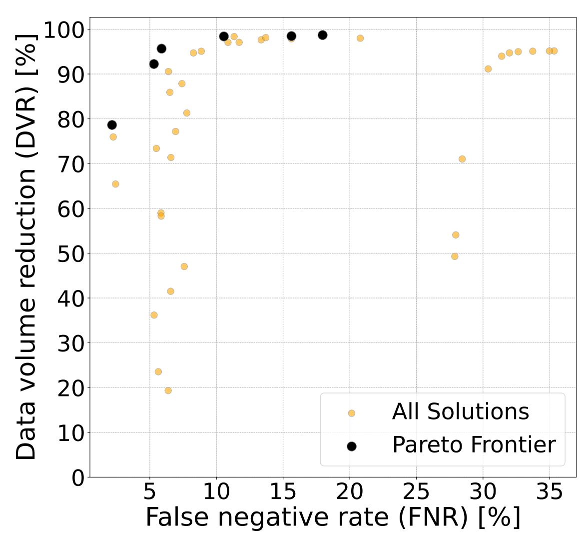

Figure 2. Solutions of all parameter combinations (orange points) and the Pareto-efficient solutions (black points), which represent the balanced trade-offs between false negative rate (FNR), data volume reduction (DVR), and computation time (not shown on plot). FNR and DVR are relative to the manually annotated reference point cloud.

Table 2. Overview of the Pareto-efficient solutions. Three solutions with either low false negative rates (FNR) or low relative computation times (RCT) are highlighted in bold to emphasize possible trade-off solutions. Computation times are normalized to the maximum value, which is the time required using only M3C2 on the full point cloud. The table presents the voxel size (v), significance level (α), FNR, data volume reduction (DVR), RCT for the Voxel step (RCT Voxel), RCT for the M3C2 step (RCT M3C2), and RCT for the full hierarchical analysis (RCT Combined).

v [m]	α	FNR [%]	DVR [%]	RCT Voxel [%]	RCT M3C2 [%]	RCT Combined [%]
3	0.90000	18.0	98.7	2.4	0.2	2.6
3	0.95000	15.6	98.5	2.4	0.2	2.6
4	**0.99000**	**5.3**	**92.2**	**2.5**	**1.0**	**3.5**
6	0.99000	10.6	98.4	2.4	0.3	2.7
6	**0.99900**	**5.9**	**95.7**	**2.4**	**0.6**	**3.0**
10	**0.99999**	**2.2**	**78.6**	**2.7**	**9.3**	**12.0**

FNR below 10% (Fig. 2). Based on these results, further analysis of Pareto efficient solutions shows that the RCT for the voxel step remains nearly constant at 2.5%, while the RCT for the M3C2 step varies depending on the input parameters voxel size and α (Tab. 2). The solution that achieves the lowest FNR has the highest RCT (for M3C2 and combined), while the solutions with the lowest RCT have the highest FNR. For a given voxel size, a larger α leads to an increased M3C2-based computation time. This effect is evident for voxel sizes of 3 m and 6 m, where the method detects more false positives with larger alpha. As a consequence, the overall computation time increases and the FNR decreases, producing more conservative results. These findings demonstrate the complex interplay between voxel size and α in determining overall computation time and FNR. The combination of a voxel size of 6 m and an α of 0.99900 provides a reasonable trade-off, achieving an FNR below 6% and reducing computation time by 97%. We select this parameter combination for our visual analysis.

4.2 Reduction of detected changes

Fig. 3 clearly demonstrates the reduction in data volume when using the hierarchical approach. Using M3C2 without hierarchical analysis enables the detection of the rockfall but also identifies numerous changes in vegetation as significant. In contrast, hierarchical analysis initially reduces the area of interest to regions where changes of interest have occurred, including some additional voxels. Subsequently, the M3C2 computation primarily detects significant changes in the rockfall area. We observe that hierarchical analysis effectively detects relevant changes while also handling vegetated areas, which would otherwise require additional filtering during preprocessing (Kromer et al., 2017).

5 Discussion

We demonstrate that combining voxel-based change detection with point-based change analysis enables rapid detection of surface changes while efficiently filtering irrelevant changes, such as non-destructive vegetation dynamics. Specifically, rockfalls can be effectively extracted from point clouds.

Compared to previous studies such as Fahle et al. (2023), our approach employs higher α necessary for detecting subtle surface changes. This difference arises from two factors: 1) While Fahle et al. (2023) utilized mobile laser scanning, we employed a permanent laser scanner. This allowed us to consistently capture the study site from the same viewpoint with identical scan settings, ensuring that the distribution of points remains highly similar between epochs. 2) Our study site includes vegetation, and therefore requires higher sensitivity in change detection. This increased sensitivity is particularly advantageous for applications where even small surface changes are highly relevant (like for rockfall monitoring).

In practical applications, such as long-term monitoring with PLS systems, our hierarchical analysis method significantly enhances computational efficiency. Let's assume a PLS scenario with hourly scans over a month (720 epochs). Even if events of similar extent to the rockfall in our study occurred between each epoch, our method would reduce the data volume to less than that of 31 epochs

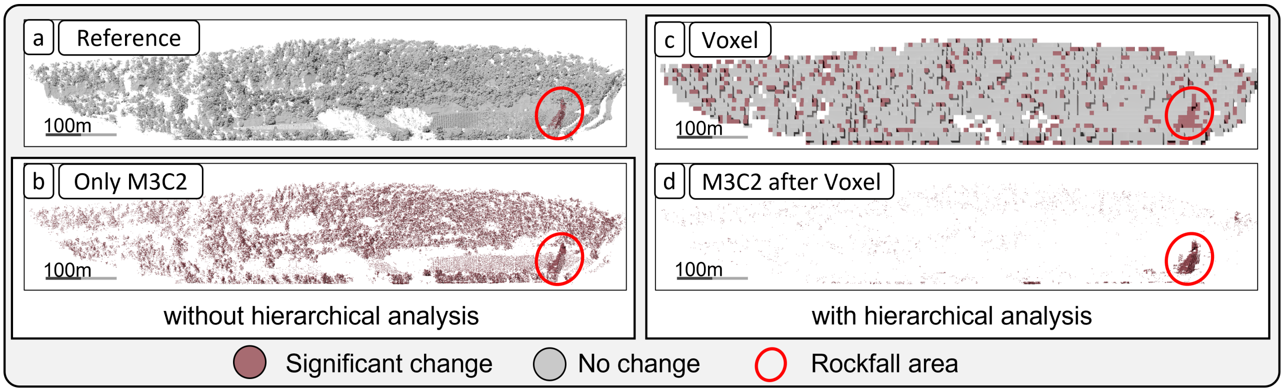

Figure 3. Comparison of (a) the manually labeled reference point cloud, (b) the M3C2 algorithm applied to the full point cloud, (c) the intermediate hierarchical analysis showing voxels with significant change, and (d) the M3C2 algorithm applied to areas with significant voxel change. Panel (a) is colored based on manual labels, while panels (b) and (d) are colored according to the significance of the M3C2 distance. Panel (c) highlights significant changes between voxels. The red ellipse indicates the area where the rockfall has occurred. Date of rockfall: 2024.08.26.

(4.3%). This reduction facilitates more detailed and frequent change analyses and could allow the integration of multiple complementary methods like M3C2 and F2S3 (Gojcic et al., 2020), which would otherwise be computationally infeasible in a near real-time context. Consequently, this approach supports more timely and informed decision-making in fields such as natural hazard monitoring.

One of the main limitations of our study is that the final parameters we identified are specific to the current study site and may require adjustment for different environments or sensor configurations. To address this issue, we suggest employing digital twins tailored to the requirements of each study site, as demonstrated in this study. The comparison of computation times does not include the time required to initialize the monitoring system and determine the final parameter combination. This initialization duration varies with the level of automation and the number of parameters evaluated, making it an important factor during system setup.

Our change detection method has low hardware requirements because keeping working memory requirements low is straightforward due to the chosen fixed voxel frame approach. On-device processing could significantly reduce data storage requirements and enable near real-time transmission of surface change information to end-users or decision makers. This capability facilitates establishing baselines for updating scan schedules, thereby enhancing the responsiveness and efficiency of monitoring systems.

6 Conclusion

This study presents a powerful hierarchical analysis framework for point cloud data that optimizes computational performance and data management by combining low-resolution change detection with high-resolution change analysis. The approach combines 1) identification of change areas by detecting statistical intra-voxel changes and 2) point based change analysis (M3C2) in the reduced areas. Our method brings us closer to near real-time on-device processing of PLS data compared to the current state-of-the-art, which relies on retrospective data analysis. Future studies should investigate various algorithm combinations and the transition to on-device processing. All presented methods are released open-source (`https://github.com/3dgeo-heidelberg`). Thus, they can be combined easily with your own deformation method to create efficient hierarchical workflows for permanent laser scanning.

Acknowledgements

This work is funded by the BMBF AIMon 5.0: Project no. 02WDG1696 and by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – no. 496418931 (VirtuaLearn3D) and no. 528521476 (Fostering a community-driven and sustainable HELIOS++ scientific software). The Federal Ministry of Education and Research

(BMBF) is funding the AIMon 5.0 project within the funding measure 'Digital GreenTech – Environmental Engineering meets Digitalisation' as part of the 'Research for Sustainability (FONA) Strategy'.

References

Anders, K., Winiwarter, L., Lindenbergh, R., Williams, J. G., Vos, S. E., and Höfle, B. (2020). 4D objects-by-change: Spatiotemporal segmentation of geomorphic surface change from LiDAR time series. *ISPRS J Photogramm Remote Sens*, 159:352–363.

Barton, C., phymec, and Sharybin, S. (2024). Cell Fracture. Blender Add-on.

Blender Online Community (2024). *Blender - a 3D modelling and rendering package*. Blender Foundation, Stichting Blender Foundation, Amsterdam.

Czerwonka-Schröder, D., Anders, K., Winiwarter, L., and Wujanz, D. (2022). Permanent terrestrial lidar monitoring in mining, natural hazard prevention and infrastructure protection – chances, risks, and challenges: A case study of a rockfall in tyrol, austria. In *Proceedings of the 5th Joint International Symposium on Deformation Monitoring—JISDM 2022*.

Deb, K. (2001). *Multiobjective Optimization Using Evolutionary Algorithms. Wiley, New York*.

Fahle, L., Petruska, A., Walton, G., Brune, J., and Holley, E. (2023). Development and testing of octree-based intra-voxel statistical inference to enable real-time geotechnical monitoring of large-scale underground spaces with mobile laser scanning data. *Remote Sensing*, 15:1764.

Gehrung, J., Hebel, M., Arens, M., and Stilla, U. (2019). A fast voxel-based indicator for change detection using low resolution octrees. *ISPRS Annals*, IV-2/W5:357–364.

Girardeau-Montaut, D. (2024). Cloudcompare (version 2.13.1) [gpl software]. retrieved from http://www.cloudcompare.org/.

Gojcic, Z., Zhou, C., and Wieser, A. (2020). F2S3: Robustified determination of 3D displacement vector fields using deep learning. *J Appl Geod*, 14(2):177–189.

Jaboyedoff, M., Oppikofer, T., Abellán, A., Derron, M.-H., Loye, A., Metzger, R., and Pedrazzini, A. (2012). Use of LIDAR in landslide investigations: A review. *Natural Hazards*, 61:5–28.

Kazhdan, M., Bolitho, M., and Hoppe, H. (2006). Poisson Surface Reconstruction. In Sheffer, A. and Polthier, K., editors, *Symposium on Geometry Processing*. The Eurographics Association.

Kromer, R. A., Abellán, A., Hutchinson, D. J., Lato, M., Edwards, T., and Jaboyedoff, M. (2017). A 4D Filtering and Calibration Technique for Small-Scale Point Cloud Change Detection with a Terrestrial Laser Scanner. *Remote Sensing*, 7(10):13029–13052.

Lague, D., Brodu, N., and Leroux, J. (2013). Accurate 3D comparison of complex topography with terrestrial laser scanner: Application to the Rangitikei canyon (N-Z). *ISPRS J Photogramm Remote Sens*, 82:10–26.

Mahalanobis, P. (1936). On the generalised distance in statistics. *Proceedings of the National Institute of Sciences of India*.

py4dgeo Development Core Team (2023). py4dgeo: library for change analysis in 4D point clouds. https://github.com/3dgeo-heidelberg/py4dgeo.

Qin, R., Tian, J., and Reinartz, P. (2016). 3d change detection – approaches and applications. *ISPRS J Photogramm Remote Sens*, 122:41–56.

Seppelt, R., Lautenbach, S., and Volk, M. (2013). Identifying trade-offs between ecosystem services, land use, and biodiversity: a plea for combining scenario analysis and optimization on different spatial scales. *Current Opinion in Environmental Sustainability*, 5(5):458–463.

Tabernig, R., Albert, W., Weiser, H., and Höfle, B. (2024). VAPC - Voxel Analysis for Point Clouds. https://github.com/3dgeo-heidelberg/vapc.

Weber, J. and Penn, J. (1995). Creation and Rendering of Realistic Trees. In *Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '95, page 119–128, New York, NY, USA. Association for Computing Machinery.

Weiser, H. and Höfle, B. (2024). Virtual laser scanning of dynamic scenes (vls-4d): Current approaches and future perspectives in remote sensing. *eartharxiv*.

Wellhausen, L., Dubé, R., Gawel, A., Siegwart, R., and Cadena, C. (2017). Reliable real-time change detection and mapping for 3d lidars. In *2017 IEEE International Symposium on Safety, Security and Rescue Robotics (SSRR)*, pages 81–87.

Williams, J. G., Rosser, N. J., Hardy, R. J., and Brain, M. J. (2019). The Importance of Monitoring Interval for Rockfall Magnitude-Frequency Estimation. *J Geophys Res Earth Surf*, 124(12):2841–2853.

Winiwarter, L., Esmorís Pena, A. M., Weiser, H., Anders, K., Martínez Sánchez, J., Searle, M., and Höfle, B. (2022). Virtual laser scanning with helios++: A novel take on ray tracing-based simulation of topographic full-waveform 3d laser scanning. *Remote Sens Environ*, 269.