Automated detection and structuration of building and vegetation changes from LiDAR point clouds

Abderrazzaq Kharroubi [1], Zouhair Ballouch [1,2], Imane Jeddoub [1], Rafika Hajji [2], Roland Billen [1]

[1]GeoScITY, UR SPHERES, University of Liège, 4000 Liège, Belgium; (akharroubi, zouhair.ballouch, i.jeddoub, rbillen)@uliege.be
[2]College of Geomatic Sciences and Surveying Engineering, Hassan II Institute of Agronomy and Veterinary Medicine, Rabat 10101, Morocco; r.hajji@iav.ac.ma

Keywords: 3D Change Detection, Semantic Segmentation, Building, Trees, LiDAR, CityJSON.

Abstract

Urban environments are continuously changing, driven by factors such as population growth and infrastructure expansion, which necessitates regular updates to urban models. Accurate, up-to-date information on these changes is critical, particularly for national mapping agencies monitoring long-term urban development. This paper presents an automated methodology for detecting building and vegetation changes within urban environments using LiDAR point clouds, focusing on the city of Liège in Belgium. By leveraging recent aerial LiDAR data from 2022, our approach identifies, models, and integrates urban changes into a refined 3D Digital Twin model of Liège. The methodology includes preprocessing steps such as coordinate systems homogenization, noise filtering, and octree-based spatial indexing, followed by semantic and instance segmentation of point clouds using the RandLA-Net deep learning model. The change detection process focuses on four categories: appearance, disappearance, modification, and unchanged features. Achieving 100% accuracy for detecting new buildings changes, as validated within the study dataset and methodology. The modelled results are structured into a CityJSON city model. This automated approach significantly enhances urban model updates by integrating detected changes into a standardized 3D representation.

1. Introduction

Urban areas undergo continuous transformations driven by population growth, expanding infrastructure, and increasing urbanization. Keeping urban models updated is vital for efficient planning, resource management, and environmental oversight. However, traditional methods for updating these models are labor-intensive, time-consuming, and often prone to errors. This highlights the need for efficient, automated techniques to monitor and integrate urban changes. LiDAR (Light Detection and Ranging) technology has become a key tool for urban modeling, providing high-resolution, three-dimensional data that significantly improves the accuracy of urban model updates. Several studies have explored LiDAR's potential for change detection and model refinement. Extending LiDAR applications to 3D models has led to the development of semi-automated techniques focused on specific features like buildings (Tamort et al. 2024) or vegetation (Hirt et al. 2021). Although promising, these methods have limitations: they typically address only one urban feature at a time, lack full automation, and struggle with scalability. Current research still faces key challenges. First, automating change detection across varied urban features remains complex due to the heterogeneity of urban landscapes, which include different building types, sizes, and vegetation densities. Second, most existing approaches struggle to integrate multiple urban elements, such as buildings and vegetation, into a unified model. Third, there is a need for scalable, cost-effective solutions that can be applied over large areas, as required by national mapping agencies and urban planners.

A critical component of our approach is the integration of semantic information with geometric changes to achieve comprehensive 3D semantic change segmentation. Below, we illustrate the difference between binary and semantic change detection. In binary change detection (Figure 1.a), changes are identified without differentiating the types of objects involved, making it difficult to extract meaningful urban insights. In contrast, semantic change detection (Figure 1.b) provides a more nuanced analysis by distinguishing between different classes, such as buildings and trees. This distinction enables a better understanding of how urban features evolve and supports more informed decision-making.

To address the existing challenges, we present an automated workflow for detecting and modeling changes in urban buildings and vegetation using LiDAR point clouds. The research focuses on the city of Liège in Belgium, an area with a diverse urban fabric and ongoing development. We use recent aerial LiDAR data from 2022 to update the city's Digital Twin in Level of Detail (LoD) 2.2. The methodology involves several steps: preprocessing the data to remove noise and outliers, using deep learning models for semantic and instance segmentation, and detecting changes based on an object-based approach. Specifically, we employ RandLA-Net for semantic segmentation to classify the point cloud into buildings, vegetation, and other features. Changes are categorized into four types: appearance, disappearance, modification, and unchanged features. The results are structured into a CityJSON model using our proposed new change extension. Our approach provides a practical and efficient solution for urban model updates, benefiting regional and national authorities involved in urban monitoring and planning.

The main contributions of this research are: (1) a fully automated workflow for urban change detection, (2) a new CityJSON extension for change information structuration and management.

The structure of this paper is as follows: Section 2 reviews related work, highlighting the current state and limitations of existing methods. Section 3 describes our methodology, including data preprocessing, segmentation, and change detection. Section 4 presents the results and their validation. Finally, Section 5 concludes the study and discusses potential future research.

This contribution has been peer-reviewed.
https://doi.org/10.5194/isprs-archives-XLVIII-2-W8-2024-227-2024 | © Author(s) 2024. CC BY 4.0 License.

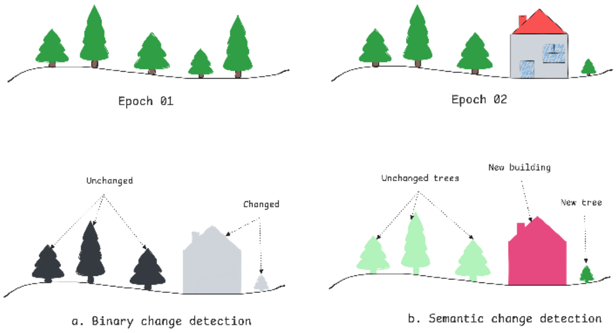


Figure 1. Difference between binary (a) and semantic change detection (b). Both projected on epoch 02.

2. Related works

3D point clouds have become a crucial data source for monitoring vegetation and buildings in urban environments. Compared to traditional image data, 3D points clouds provide richer information by capturing the geometric and structural details of objects, facilitating more accurate monitoring and analysis (Stilla and Xu 2023). However, the irregular and unstructured nature of point clouds presents unique challenges, requiring advanced methods to extract meaningful insights. Change detection approaches can be broadly categorized into two main families: derived-product methods and direct comparison techniques.

Approaches based on derived products: Derived-product methods transform raw 3D point clouds into structured representations like Digital Surface Models (DSMs) or voxel grids, making it easier to apply traditional 2D change detection algorithms. For example, DSM-based methods enable change detection in buildings by calculating height differences between multi-temporal datasets (Dini et al. 2012). Similarly, Canopy Height Models (CHMs) derived from LiDAR data are used to monitor vegetation changes, with techniques like watershed algorithms identifying individual tree crowns (Slavík et al. 2020). Voxel-based methods also regularize point clouds for easier comparison, reducing the impact of varying point densities (Harith et al. 2021). However, these derived-product methods have inherent limitations. The conversion from 3D to 2D or voxelized grids often results in information loss and interpolation errors, which can reduce the accuracy of change detection, particularly for objects with fine geometric details. Moreover, studies have shown that derived approaches struggle in areas with complex topography or where high precision is required, making direct comparison methods more suitable for such scenarios (Okyay et al. 2019).

Direct approaches using raw point clouds: Direct methods operate on raw 3D point clouds, preserving the spatial and topographical relationships of data points. Traditional techniques like Cloud-to-Cloud (C2C) and Multi-Scale Model-to-Model Cloud Comparison (M3C2) measure distances between points in multi-temporal datasets to detect changes (Lague, Brodu, and Leroux 2013; Girardeau-Montaut et al. 2005). While effective for precise geometric comparisons, C2C methods often fail to provide context about the type of changes, such as the appearance or disappearance of buildings or vegetation. Recent advancements have incorporated semantic segmentation into change detection workflows, referred to as post-classification change detection. For instance, (Awrangjeb 2015) introduced a connected component analysis method for updating building information in topographic maps, while (Dai, Zhang, and Lin 2020) developed an object-based approach that detects changes like newly constructed or demolished buildings. (Tran, Ressl, and Pfeifer 2018) extended this to vegetation monitoring, identifying changes at the tree level using a supervised classification framework.

Machine learning and deep Learning have significantly enhanced the accuracy of 3D change detection. Deep learning models, including Siamese Neural Networks and Graph Convolutional Networks (GCNs), have introduced more advanced solutions, processing raw point clouds directly and achieving high performance in change detection task (Nagy, Kovacs, and Benedek 2021). Siamese KPConv, a notable deep learning approach, directly operates on 3D point clouds to detect changes, reducing the need for voxelization and retaining geometric details (de Gélis, Lefèvre, and Corpetti 2023). However, deep learning models often require substantial computational resources and large annotated datasets, which can be a limitation in real-world applications. Self-supervised learning techniques are emerging to mitigate these data challenges, employing deep clustering and contrastive learning to improve the performance of unsupervised 3D change detection methods.

Semantic Change Detection (SCD) extends traditional methods by identifying both the regions and types of changes, such as distinguishing between a changed building and a tree. SCD approaches can be categorized into single encoder, dual encoder, and triple encoder models. Single encoder methods often suffer from class overlap, while dual encoder models like Siamese architectures provide better differentiation but may miss temporal relationships between features. Triple encoder approaches incorporate auxiliary information, enhancing accuracy but increasing computational complexity.

Despite advancements in geometric and semantic change detection, current methods often focus on one aspect, neglecting the other, reliance on labeled datasets, and limited scalability. This study introduces an unsupervised semantic change detection (SCD) pipeline that integrates both semantic and geometric analysis using 3D point cloud.

3. Methodology

This study presents a comprehensive methodology for automating the detection and modeling of building and vegetation changes using LiDAR point clouds. As summarized in Figure 2, the workflow begins with data preparation, including coordinate systems standardization, cleaning, and preprocessing of multi-temporal datasets to ensure consistency. Semantic and instance segmentation are then performed to accurately classify and isolate objects such as buildings and trees. The subsequent steps involve defining and detecting changes, quantifying these changes using relevant metrics, and generating detailed 3D models. Finally, the results are structured into the CityJSON format, facilitating seamless integration into urban digital twin applications.

3.1 Data preparation

Before the implementation phase, several data preprocessing steps were carried out. First, adaptation to the same coordinate system was performed. This step was followed by cleaning and merging the tiles covering the study area. Finally, the point clouds were prepared according to the requirements of the deep learning architecture used for classification.

3.2 Semantic and instance segmentation

Semantic segmentation was performed to accurately extract buildings and trees. To perform semantic segmentation, we used RandLA-Net as a deep learning model, which had been trained and validated on a dataset comprising urban LiDAR scans from the city of Liège, as described in our earlier work (Ballouch et al. 2024). This model achieved classification accuracy of 95% for buildings and vegetation, ensuring reliable semantic segmentation in this study. Following this, instance segmentation was applied to trees using the TreeISO algorithm (Xi and Hopkinson 2022) to isolate each tree individually and assign a unique identifier. TreeISO segments individual trees using a hierarchical, graph-based clustering approach. It first groups points into fine clusters based on spatial proximity, then aggregates these into tree segments using connectivity rules informed by elevation and distance. While designed for terrestrial laser scanning (TLS), we adapted it for aerial laser scanning (ALS) to isolate individual trees effectively from our dataset. For buildings, instance segmentation was not required, as modeling relied on PICC data, which includes footprints and unique identifiers for each building.

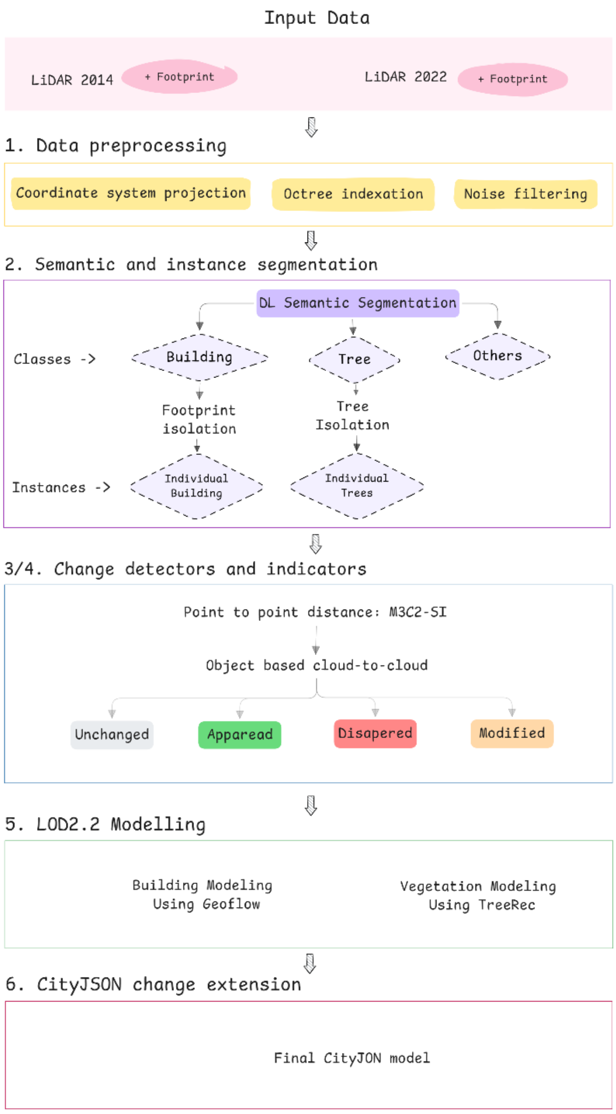


Figure 2. Overview of the Methodology Workflow

3.3 Change definition and detection

To detect changes between two epochs, we propose a method that combines point-level indicators, clustering, and object-level metrics. This approach ensures robust identification of changes while addressing common artifacts, such as façade-related noise in buildings and overlapping canopies in trees. Changes are categorized into four classes for both buildings and trees: new, lost, modified, and unchanged.

Point-level change detection begins with the calculation of distances between the two epochs using a modified version of the Multi-Scale Model-to-Model Cloud Comparison (M3C2) algorithm. To mitigate artifacts from vertical structures like building façades or dense tree canopies, M3C2 distances are projected onto a horizontal plane. This ensures that the detected changes reflect meaningful modifications in object presence or geometry rather than irrelevant vertical differences caused by variations in scan viewpoints (see Figure 10). Using these distances, objects are initially classified as new or lost based on horizontal threshold:

- **New objects** are detected in the second epoch where M3C2 distances exceed a threshold, indicating objects present in the second epoch but not in the first.
- **Lost objects** are identified from the first epoch where M3C2 distances exceed the threshold, signifying objects present in the first epoch but absent in the second.

To refine these classifications, a connected-component clustering algorithm is applied. This groups adjacent points with similar change indicators, forming spatially coherent regions of change. Clusters that fall below minimum size, typically caused by noise or minor misalignments, are removed to ensure that only meaningful changes are retained. For objects classified as unchanged, vertical changes are further analyzed. Buildings are examined for structural modifications such as height increases (e.g., additional floors) or reductions (e.g., partial demolitions), while trees are assessed for vertical growth or trimming. This vertical analysis leverages the M3C2 distances in the z-dimension to detect significant differences and our metrics for change quantification.

3.4 Object change metrics

After detecting changes, the next step involves quantifying them to characterize the modifications for both buildings and trees. These metrics provide detailed insights into the type and extent of changes.

For modified buildings, the following metrics are extracted:

- **Height Difference**: The average and maximum height differences are calculated to identify taller or shorter structures.

For trees, we detect four categories of change: new, lost, modified, and unchanged. New, lost, and unchanged trees are identified using M3C2 distance metrics to compare spatial correspondence between epochs. For modified trees, the following metrics are used to assess growth or pruning:

- **Tree Height (99th Percentile Z)**: The maximum height of the tree above ground. This metric identifies the maximum canopy height while avoiding influence from outliers (e.g., single, unconnected high points due to noise).

These metrics provide a clear and quantifiable basis for evaluating structural changes in individual trees. Since trunk positions are unavailable due to the aerial acquisition, we identify corresponding tree objects between epochs by finding the closest tree object spatially. If the nearest tree surpasses a defined maximum distance, the tree is flagged for manual user confirmation. This assumes new and lost trees have already been filtered out, leaving modified or unchanged trees for comparison. For each tree in the second epoch, the nearest tree in the first epoch is determined based only on the spatial proximity of their centroids or canopy centers, without relying on constraints like canopy dimensions or height metrics, as these can vary due to changes.

Changes are then classified using either threshold-based rules or supervised learning techniques if a labelled dataset is available. Threshold-based rules are predefined, such as classifying trees with height increases above a certain value as “heightened” or “lowered”. Alternatively, when labeled training data is available, machine learning algorithms like Random Forests can classify changes based on the extracted metrics. This approach allows for more nuanced classification while leveraging a training dataset.

3.5 Buildings and trees modelling

After the change detection step, 3D modeling of different change classes is conducted. For buildings, the footprints extracted from the previously described PICC data are used. They include attribute information for each building, along with unique identifiers. In addition to these vector data, building point clouds, derived from semantic segmentation, were used to reconstruct building geometry. Geoflow tool[1] enabled this reconstruction in CityJSON format, achieving a level of detail 2.2. For the 3D modeling of trees, the instance segmentation results were used to calculate the height of each tree above the ground, as well as specific parameters (height, width, perimeter, etc.). These extracted parameters were also reconstructed in CityJSON format using a Python script. The schema and geometry of the reconstructed models were then validated. For geometry validation, we used Val3dity[2], an open-source software dedicated to validating 3D primitives (geometries) of the model. For schema validation, we relied on the official validator[3] for CityJSON files.

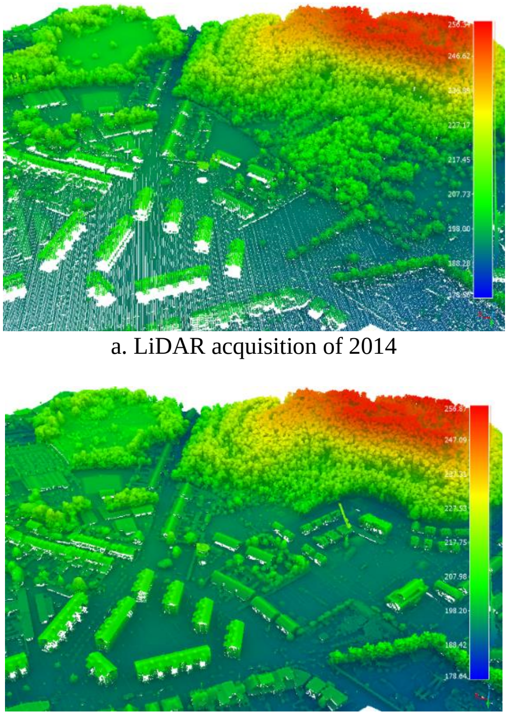
3.6 Structuration into CityJSON

This step involves the structuring of the change’s information into a standardized data model to enhance usability and interoperability. CityJSON is a lightweight and developer-friendly format for representing, storing, and extending 3D city models. Given its simplicity and its extensibility, we integrate the change information in compliance with the CityJSON standard by defining new attributes for existing city objects, namely buildings and vegetation. We define a set of attributes that report the change characteristics (e.g., the change type (refer to Table 1), the height variation, and the change uncertainty). We structure this information following the CityJSON extension specifications. The change extension is adopted upstream as a way of organizing, storing, and managing the change information in 3D city models. Following the CityJSON extension specifications 2.0.1, the change attributes are integrated as *extraAttributes* for existing city objects. Each city object is extended by the three main change attributes: change, height variation, and change uncertainty. Their values are stored respectively following the defined CityJSON extension.

4. Experiments and results

4.1 Data description

In this study, two types of data were used: LiDAR and vector. The LiDAR point clouds from 2022 have specific characteristics, including an average flight altitude (AGL) of 2400 m, a density of 6.8 points/m², and the use of Double LMSQ780 and Double VQ780II-S equipment. The planimetric accuracy is evaluated with an RMSE of ≤ 1 m, while the altimetric accuracy reaches an RMSE of ≤ 0.4 m. In contrast, the point clouds from 2014 were collected at a minimum altitude of 1015 m and a maximum altitude of 1550 m, with a density of only 0.8 pulses/m², resulting in an average spacing of 1.13 m in the direction of flight and 1.14 m transversely, while also displaying a planimetric and altimetric accuracy with a maximum RMSE of 1 m and 0.4 m, respectively. Additionally, data from the Mapping Framework Information Plan (PICC) were utilized; these vector data allowed for the extraction of footprints and attribute information for each building, complementing the analysis derived from the point clouds.



a. LiDAR acquisition of 2014

b. LiDAR acquisition of 2022

Figure 3. Epoch 1 (a) and 2 (b) of aerial point clouds over the study area colored by Height ramp.

4.2 Implementation

Semantic segmentation was performed using RandLA-Net, a deep learning model previously validated for urban classification tasks. The model classified the point cloud into four classes: buildings, vegetation, ground, and others. The overall accuracy for buildings and vegetation exceeded 95%. The experiments were conducted on a high-performance workstation equipped with an NVIDIA GeForce RTX 3090 GPU, an Intel i9-10980XE CPU, and 256 GB of RAM.

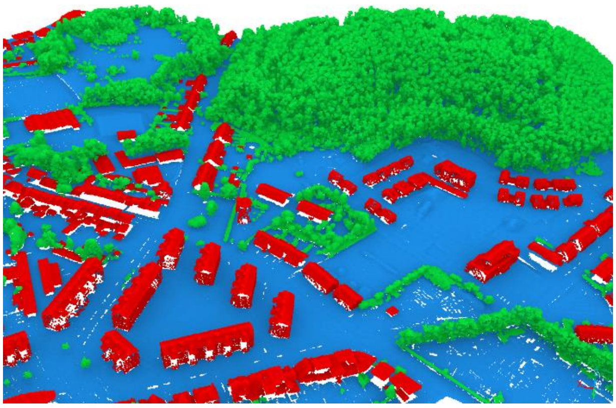


Figure 4. Results of semantic segmentation of 2022’s data

[1] https://github.com/geoflow3d/geoflow-bundle
[2] https://github.com/tudelft3d/val3dity
[3] https://validator.cityjson.org/

After semantic segmentation, vegetation classes were further processed using TreeISO for instance segmentation. However, due to the inclusion of fences and low vegetation in the vegetation class, additional filtering was applied. To isolate valid tree clusters and exclude noise or irrelevant objects (e.g., fences or isolated points), we applied a filtering criterion based on the height distribution of points within each cluster. For each segmented cluster, the 95th percentile of point heights was calculated. Clusters where the 95th percentile height exceeded 3.5 m were retained, as this threshold reflects a meaningful minimum height for trees in the study area. This percentile-based approach ensures robustness by adapting to variations in point density and cluster size, reducing sensitivity to noise and outliers compared to a fixed point count.

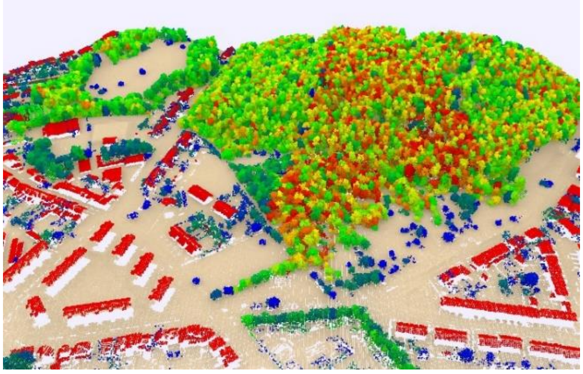


Figure 5. Vegetation clustering using TreeISO after removing small components, low vegetation, and fences (data 2014).

To detect new and demolished buildings, the building class was projected onto the XY plane, and the M3C2 algorithm was applied. For new buildings, epoch 02 served as the reference, while epoch 01 was inverted to identify demolished buildings. Height changes were calculated for individual buildings, which were isolated using PICC shapefile footprints. We applied a height threshold of 3 meters, equivalent to the average height of a floor, to detect significant vertical changes. Buildings with height increases exceeding 3 meters were classified as "heightened," while those with reductions exceeding 3 meters were classified as "lowered." However, no such vertical changes were detected in the study area, as all buildings remained within this threshold or were completely new.

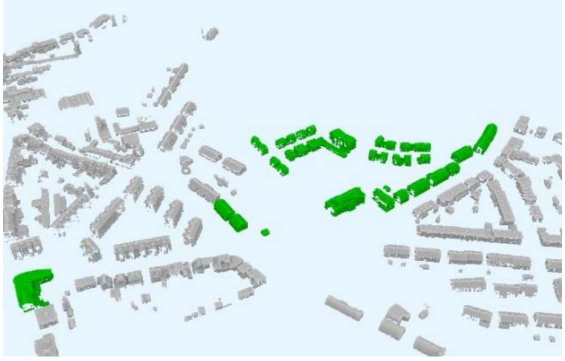


Figure 6. The new building visualized in green, and unchanged in gray.

For trees, we applied a height threshold of 1 meter to detect significant vertical changes, considering it sufficient to capture meaningful growth or pruning while avoiding noise from minor seasonal variations or data inaccuracies. Trees with height increases exceeding 1 meter were classified as "grown," while those with reductions beyond this threshold were classified as "pruned." However, challenges arose in areas with dense canopy coverage, where overlapping layers occasionally affected the clustering process, leading to misclassification. This limitation emphasizes the importance of refining segmentation techniques to handle complex vegetation structures more effectively in future studies.

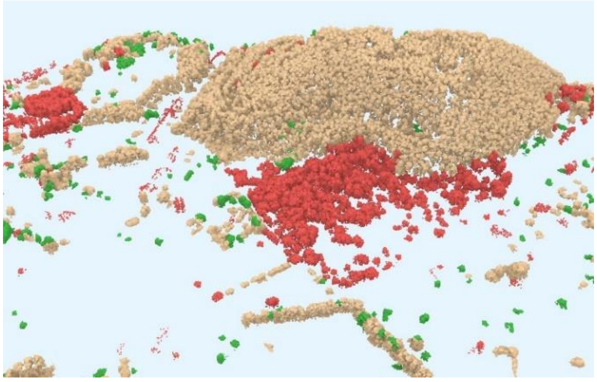


Figure 7. New vegetation visualized in green, removed in red and modified in brown.

In this work, change attributes were not included in the reconstruction process, as described in Section 3.5. We have therefore proceeded as follows: for each city model, we upgraded the city models v1.1 for buildings and v1.0 for vegetation to version v2.0 using cjio to comply with our extension, and we have added the corresponding attributes using Python code. For instance, in the LoD2.2 building model, representing all new buildings, we have assigned the change attributes to each building, reflecting the type of change (i.e., new), the height variation, and the change uncertainty. We proceeded in the same way for vegetation to report the change type.

City Objects	Building	Vegetation
Change type	Unchanged, New, Demolished, Heightened, Lowered	Unchanged, New, Lost, Growth, Trimmed

Table 1. Change type according to the city objects

The CityJSON files containing the change attributes are validated using the CityJSON schema validator (Figure 9). The aim behind the change extension is to ensure that these attributes are maintained in a standardized way and are automatically updated when a new value is available. Results are viewed below.

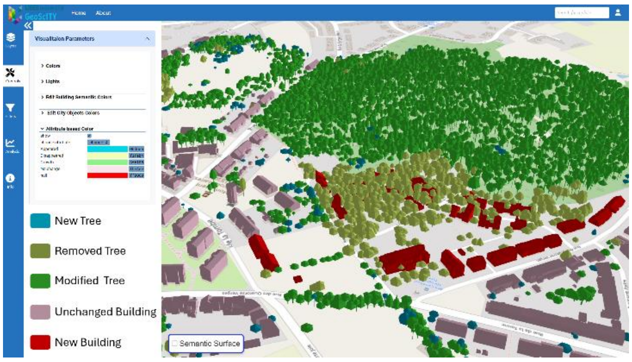
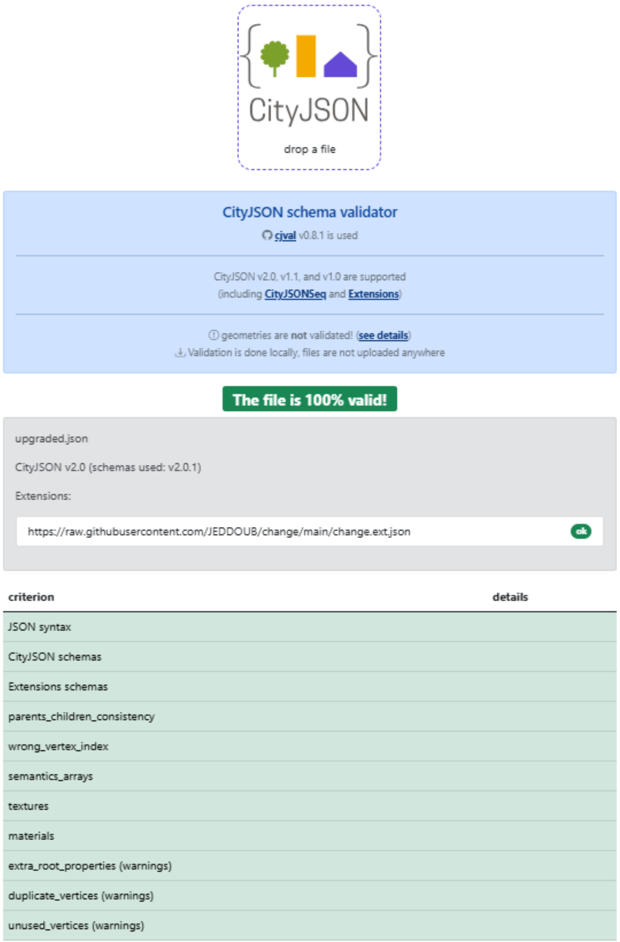


Figure 8. The results are displayed in an in-house developed tool for 3D city model structuration, management and visualization ©GeoScITY



CityJSON schema validator

cjval v0.8.1 is used

CityJSON v2.0, v1.1, and v1.0 are supported
(including CityJSONSeq and Extensions)

ⓘ geometries are not validated! (see details)
Validation is done locally, files are not uploaded anywhere

The file is 100% valid!

upgraded.json

CityJSON v2.0 (schemas used: v2.0.1)

Extensions:

https://raw.githubusercontent.com/JEDDOUB/change/main/change.ext.json ok

criterion	details
JSON syntax	
CityJSON schemas	
Extensions schemas	
parents_children_consistency	
wrong_vertex_index	
semantics_arrays	
textures	
materials	
extra_root_properties (warnings)	
duplicate_vertices (warnings)	
unused_vertices (warnings)	

Figure 9. Change extension schema validation

4.3 Discussion

The performance of the proposed methodology was qualitatively assessed by visually inspecting the results for new, lost, modified, and unchanged buildings and vegetation. Visual comparisons between the two epochs demonstrated that all new and demolished buildings were correctly identified. The use of the M3C2 algorithm, combined with the projection onto the XY plane, minimized false positives associated with vertical structures, such as façades or overlapping tree canopies. This ensured reliable detection of changes in urban environments.

While the methodology accurately detected new and lost trees, some errors were observed in the classification and clustering stages. Specifically, small clusters of vegetation were occasionally misclassified as trees, and dense vegetation often led to the over-segmentation of tree canopies. These limitations highlight the need for further refinement in the instance segmentation and clustering algorithms. Additionally, the proposed methodology relies on several threshold values (e.g., the height threshold for tree filtering and the M3C2 distance threshold for change detection). These thresholds were determined empirically and may affect the reproducibility and generalizability of the results. Future work will explore adaptive or data-driven approaches to optimize these thresholds across different datasets.

Overall, the methodology demonstrates robustness in detecting major urban changes, particularly for buildings. However, the challenges associated with vegetation segmentation and classification suggest areas for improvement in future research.

5. Conclusion

This study introduced an automated methodology for detecting and structuring changes in buildings and vegetation using multi-temporal LiDAR point clouds. By combining semantic segmentation, object-based clustering, and geometric analysis, the approach effectively identified new, lost, modified, and unchanged features, structuring them into a CityJSON format for urban digital twins. Applied to LiDAR data from Liège, Belgium, the method demonstrated high reliability in building detection and vegetation, leveraging tailored metrics like tree height. However, challenges remain in accurately clustering dense vegetation and reduce the reliance on empirically derived thresholds. Future work will focus on improving threshold adaptability, integrating additional urban features, and exploring supervised learning for enhanced scalability and robustness in large-scale urban monitoring.

Acknowledgements

This research was conducted as part of Abderrazzaq Kharroubi's PhD thesis. Abderrazzaq is an Aspirant of the Fonds de la Recherche Scientifique (FNRS), and we sincerely appreciate the financial support provided by FNRS.

References

Awrangjeb, Mohammad., 2015: Effective Generation and Update of a Building Map Database through Automatic Building Change Detection from LiDAR Point Cloud Data. *Remote Sensing* 7 (10), 14119–50. doi.org/10.3390/rs71014119.

Ballouch, Zouhair, Imane Jeddoub, Rafika Hajji, Jean Paul Kasprzyk, and Roland Billen., 2024: Towards a Digital Twin of Liege: The Core 3D Model Based on Semantic Segmentation and Automated Modeling of LiDAR Point Clouds. In *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 10, 13–20. Copernicus Publications. doi.org/10.5194/isprs-annals-X-4-W4-2024-13-2024.

Dai, Chenguang, Zhenchao Zhang, and Dong Lin., 2020: An Object-Based Bidirectional Method for Integrated Building Extraction and Change Detection between Multimodal Point Clouds. *Remote Sensing* 12 (10). doi.org/10.3390/rs12101680.

Dini, G R, K Jacobsen, F Rottensteiner, M Al Rajhi, and C Heipke., 2012: 3D BUILDING CHANGE DETECTION USING HIGH RESOLUTION STEREO IMAGES AND A GIS DATABASE.

Gélis, Iris de, Sébastien Lefèvre, and Thomas Corpetti., 2023: Siamese KPConv: 3D Multiple Change Detection from Raw Point Clouds Using Deep Learning. *ISPRS Journal of Photogrammetry and Remote Sensing* 197 (March), 274–91. doi.org/10.1016/j.isprsjprs.2023.02.001.

Girardeau-Montaut, D., M. Roux, R. Marc, and G. Thibault., 2005: Change Detection on Point Cloud Data Acquired with a Ground Laser Scanner. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives* 36.

Harith, Aljumaily, Laefer Debra F., Cuadra Dolores, and Velasco Manuel., 2021: Voxel Change: Big Data–Based Change Detection for Aerial Urban LiDAR of Unequal Densities.

This contribution has been peer-reviewed.

Journal of Surveying Engineering 147 (4), 4021023. doi.org/10.1061/(ASCE)SU.1943-5428.0000356.

Hirt, Philipp Roman, Yusheng Xu, Ludwig Hoegner, and Uwe Stilla., 2021: Change Detection of Urban Trees in MLS Point Clouds Using Occupancy Grids. *PFG - Journal of Photogrammetry, Remote Sensing and Geoinformation Science* 89 (4), 301–18. doi.org/10.1007/s41064-021-00179-4.

Lague, Dimitri, Nicolas Brodu, and Jérôme Leroux., 2013: Accurate 3D Comparison of Complex Topography with Terrestrial Laser Scanner: Application to the Rangitikei Canyon (N-Z). *ISPRS Journal of Photogrammetry and Remote Sensing* 82, 10–26. doi.org/10.1016/j.isprsjprs.2013.04.009.

Nagy, Balazs, Lorant Kovacs, and Csaba Benedek., 2021: ChangeGAN: A Deep Network for Change Detection in Coarsely Registered Point Clouds. *IEEE Robotics and Automation Letters* 6 (4), 8277–84. doi.org/10.1109/LRA.2021.3105721.

Okyay, Unal, Jennifer Telling, Craig L. Glennie, and William E. Dietrich., 2019: Airborne Lidar Change Detection: An Overview of Earth Sciences Applications. *Earth-Science Reviews* 198 (August): 102929. doi.org/10.1016/j.earscirev.2019.102929.

Slavík, Martin, Karel Kuželka, Roman Modlinger, Ivana Tomášková, and Peter Surový., 2020: Uav Laser Scans Allow Detection of Morphological Changes in Tree Canopy. *Remote Sensing* 12 (22), 1–15. doi.org/10.3390/rs12223829.

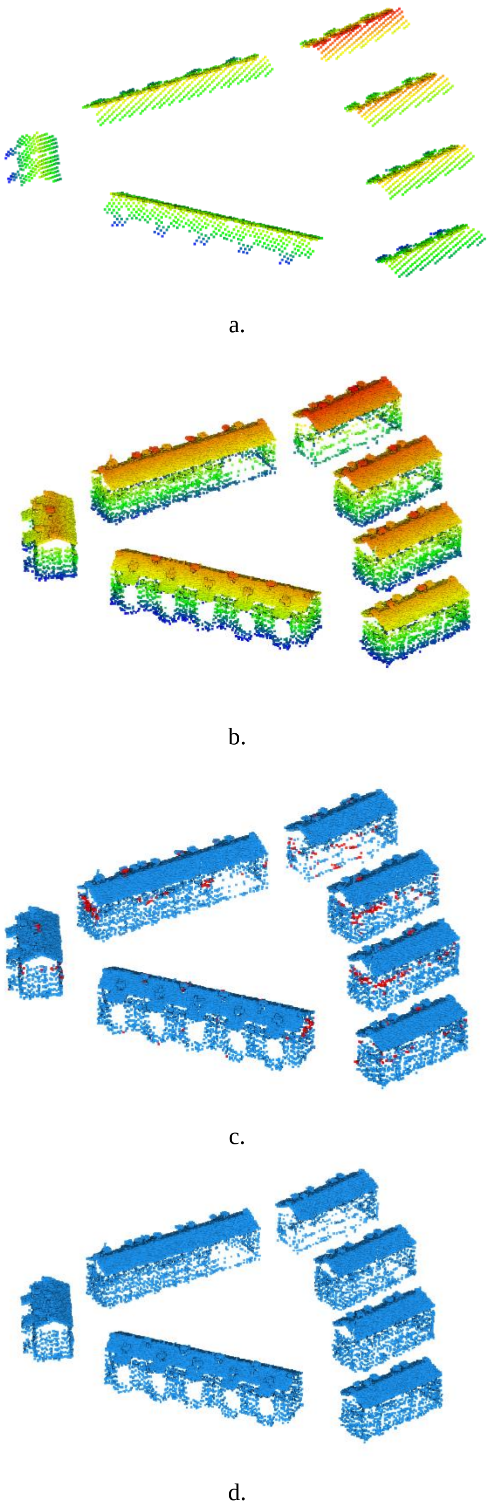
Stilla, Uwe, and Yusheng Xu., 2023: Change Detection of Urban Objects Using 3D Point Clouds: A Review. *ISPRS Journal of Photogrammetry and Remote Sensing*. Elsevier B.V. doi.org/10.1016/j.isprsjprs.2023.01.010.

Tamort, A., A. Kharroubi, R. Hajji, and R. Billen., 2024: 3D CHANGE DETECTION FOR SEMI-AUTOMATIC UPDATE OF BUILDINGS IN 3D CITY MODELS. In *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives*, 48, 349–55. International Society for Photogrammetry and Remote Sensing. doi.org/10.5194/isprs-archives-XLVIII-4-W9-2024-349-2024.

Tran, Thi Huong Giang, Camillo Ressl, and Norbert Pfeifer., 2018: Integrated Change Detection and Classification in Urban Areas Based on Airborne Laser Scanning Point Clouds. *Sensors (Switzerland)* 18 (2). doi.org/10.3390/s18020448.

Xi, Zhouxin, and Chris Hopkinson., 2022: 3D Graph-Based Individual-Tree Isolation (Treeiso) from Terrestrial Laser Scanning Point Clouds. *Remote Sensing* 14 (23). doi.org/10.3390/rs14236116.

Appendix



a.

b.

c.

d.

Figure 10. Visual Comparison of LiDAR point clouds and change detection Results: (a) Point Cloud from 2014, (b) Point Cloud from 2022, (c) Significant Changes Detected Using M3C2 (Red Indicating Façade Noise as relevant change), and (d) Noise-Free Changes Using Projected M3C2.