Point Clouds for 3D Land Administration: Integrating Floor Plans and Nationwide Airborne LiDAR (AHN)

Citra Andinasari[1], Peter van Oosteroom[2], Edward Verbree[3]

[1] Faculty of Architecture and the Built Environment, Delft University of Technology - c.andinasari@student.tudelft.nl
[2] Faculty of Architecture and the Built Environment, Delft University of Technology - p.j.m.vanoosterom@tudelft.nl
[3] Faculty of Architecture and the Built Environment, Delft University of Technology - e.verbree@tudelft.nl

Keywords: Land Administration System, Point Cloud, Floor Plan, 3D LAS, 3D Modelling

Abstract

As urban environments become increasingly vertical, Land Administration Systems (LAS) must support complex 3D spatial representations. While Building Information Models (BIM) offer such capabilities, they are not always available. This paper investigates an alternative approach using point clouds for 3D LAS, focusing on the integration of scanned cadastral floor plans and airborne LiDAR from the Actueel Hoogtebestand Nederland (AHN). We present a semi-automated pipeline that extracts floorplan geometries, segments and enhances AHN data, and synthesizes room-level point clouds. Results from a case study in Rotterdam demonstrate the potential of this approach in the absence of BIM, supporting legal space definition and public visualization. However, challenges such as misalignment due to occlusion in AHN data and inconsistent quality in older floor plan drawings affect the accuracy and automation of the process. The synthetic point clouds include room-level attributes, enabling a seamless integration with AHN, offering a representation of real-world features such as building facades, walls, and fences, which often delineate cadastral boundaries.

1. Introduction

Rapid growth in urban areas has led to an increasing number of apartment buildings. This growth requires a LAS capable of optimally storing and visualizing the legal status of these structures, ideally with 3D representation. LAS is a system formed by land administration and land registration, which maps the land parcels and registers their Right, Restrictions, and/or Responsibilities (RRR). Building Information Modeling (BIM) has been widely used in various studies as 3D representation in digital twin (Nguyen and Adhikari, 2023; Alonso et al., 2019), demonstrating great potential to represent LAS (Mao et al., 2024; Meulmeester, 2019). However, since not all buildings have BIM data available, it raises the question of how to address this limitation.

Recent studies have used point clouds as the basis for creating digital twins. Baauw (2021) studies that the AHN point cloud is capable of fulfilling the basic requirements of a digital twin as it provides a realistic 3D visual representation and, through segmentation and classification, the semantic information can be derived, allowing direct interaction. Using point cloud, historical or previous epoch data can be easily compared for change detection and integrated with temporal attributes. However, as the point cloud from ALS can only capture the exterior building envelope, an additional method to model the walls and slabs for property boundaries needs to be explored. In the Netherlands, providing notarial deeds to the Cadastre government is obligatory, including floor plans to register the apartment rights. As the land administration system required a real-world presentation, this study attempted to visualize the 3D LAS by directly using the point cloud, enriching its semantics, and representing the 3D spatial unit derived from the floor plan.

2. Related Works

2.1 AHN Point Cloud

AHN is a Dutch national dataset acquired using ALS technology containing a point cloud, digital terrain model, and digital elevation model. Point cloud is a set of 3D data points that can be organized to capture geometric information of the entire 3D object, and also can contain attributes like semantic information (e.g. classification) and RGB color. Since its first measurement in 1996, AHN has been updated for a period of time and produced 5 (five) data series. AHN has a height accuracy of no more than 5 (five) cm with a point density between 6 and 10 points per square meter for AHN2 and AHN3 and between 10 and 14 points per square meter for AHN4. The planimetric accuracy of AHN versions 2 to 4 is roughly 5 cm random error and 8 cm systematic error (AHN, 2020). Since AHN3, the classification has been provided, such as ground, vegetation, building, and water (AHN, 2020); however, the current AHN5 still does not have a sufficient classification for building class as it only classifies the roof, not the entire building as done in AHN3 and AHN4.

2.2 Point cloud for 3D Land Administration

Visualization in the Land Administration context focuses on the representation of ownership boundaries and their related legal information. With a 3D map, the visualization is upgraded to more complex 3D structures with a sense of depth that is closer to the real world representation (Pouliot et al., 2018). A 3D parcel is the fundamental spatial unit in a LAS to which a unique and homogeneous set of rights, responsibilities, and restrictions (RRRs) is assigned. Homogeneous means that the same combination of RRRs applies uniformly to the entire 3D spatial unit. The 3D parcel is the largest spatial extent where this homogeneity holds; extending the parcel would introduce different

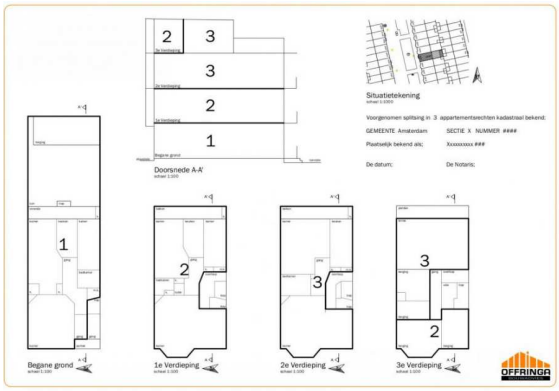


Figure 1. 2D Floor plan example in notarial deed (Meulmeester, 2019)

RRRs, while subdividing it would create neighboring parcels with identical RRRs (Oosterom et al., 2011).

Various studies have explored the use of point clouds for 3D Land Administration. Koeva et al. (2019) demonstrated the ability to automatically detect changes in building geometry over time by utilizing point clouds linked to the Land Administration Domain Model (LADM). The study showed promising results, as relevant changes for 3D Land Administration (e.g., walls and rooms) could be differentiated from temporary changes (e.g., people and furniture) and were connected to spatial subdivisions. This approach enables the Land Administration database to be updated based on the detected changes. However, the study was conducted in a university building, which does not fully represent the full range of real-world scenarios involving private units.
Another study by Bydłosz et al. (2021), in Cracow Country, Poland, proposed a 3D LAS using 3D BIM reconstructed from TLS point clouds. While point clouds offer a more realistic representation and can accommodate differences from the original architectural design, preparing a 3D model using TLS involves significant costs in terms of both time and money, which is not always possible, especially for large-scale.

2.3 Floor Plan

When registering land, the boundaries of a parcel must be clearly drawn to represent the exact division between it and neighboring properties. In the case of apartment buildings, the boundaries become more complex, as they must be represented not only on horizontal planes but also on vertical planes. Therefore, as stated in Articles 5 and 6 of the Implementation Regulation of the Land Registry Act 1994, a detailed drawing must be included when registering an apartment unit in a notarial deed. This drawing should depict the boundaries of the land, as well as a floor plan that clearly illustrates the division of private and common areas on both the ground floor and upper floors of the building (Koninkrijksrelaties, n.d.). Figure 1 illustrates that property boundaries are outlined with thick black lines (Meulmeester, 2019), which are more prominent compared to normal walls.

2.4 Parsing Floor Plan

Yin et al. (2009) describes that to decipher layout information, parsing the floor plan is required, which involves four steps: (1) Noise removal: A scanned image often contains noise and irrelevant details that need to be removed through image cleaning in order to enhance the quality of graphics recognition; (2) Text extraction: The system identifies and separates text from other graphical elements to facilitate further analysis; (3) Vectorization: To transform image pixels to the geometric primitive traditionally includes two steps. First, the raster bitmap is converted to a set of pixel chains with algorithms like parametric model fitting (HT), contouring, and skeletonization. After that, by implementing polygonal approximation or estimating curvature to determine key point segments, point chains can be segmented into sets of lines, polylines, and circular arcs; (4) Symbol recognition: After vectorization, it identifies and organizes architectural symbols or elements by using predefined constraints, thereby creating a structured representation of the building layout.
Thus, the floor plan would be preprocessed first, including cleaning the scanned image file, increasing the quality, and adjusting its scale and orientation for easier further processing. After georeferencing the raster file using QGIS, the information must be extracted for vectorization. Nottrot et al. (2023) utilized OpenCV to generate building outlines for each floor by identifying shape contours and drawing a convex hull from floor plans that contain multiple floors on a single page. The corresponding floor can be identified from the text using ACV. The resulting outline is then compared to the BAG polygons to match their scale and orientation, ensuring consistency with real-world representations.

3. Methodology

Three cadastral apartment drawings from Kadaster are used as samples for this research, drawn in different years: 1999, 2002, and 2019, located in the Rotterdam municipality. To reconstruct building point clouds from the apartment drawings for 3D LAS web visualization, five main steps are conducted. The following subsection will address the following challenges: (1) What is the suitable method to parse the cadastral floor plan? (2) How can the AHN data sets (time series, multiple versions) be integrated and pre-processed to best represent building outer envelopes? (3) What approach can be used to represent apartment spatial units and their boundaries using point cloud?

3.1 Parsing Floor Plan

Parsing the floor plan starts by preprocessing a single PDF (scan) of the cadastral apartment that contains multiple floor plans of one building. Please note there are two type of boundaries on the floorplan: the thick boundaries represent property boundaries (with a identifying label) and the thin boundaries represent the spaces inside of them. The Python package easyOCR is used to detect floor keywords and cadastral id. The OpenCV library is then applied to detect contours in the drawing near the label. If a matching contour was found above the label, it crops the image using the bounding box of the contour and exports it into a PNG image. depicts the vectorization process. Initially, each PNG image must first be converted into grayscale. Using OpenCV, it applies thresholding to create an inverted binary image, where pixel values greater than the threshold become 0 (black), while pixels with lower values (darker) become 255 (white), with the threshold determined automatically from the image histogram. Followed by morphological operations with their corresponding kernels; morphological opening, remove white pixels near edges first (erosion) then add white

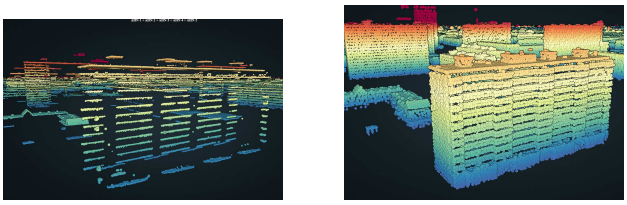
pixels around edges (dilation) to remove noise, and morphological closing, add white pixels near edges first (dilation) then remove pixels around edges (erosion) to fill small gaps to ensure full contours of walls are formed. Afterwards, it retrieves all contours and simplifies them to ensure the contour shape without redundant vertices. The tiny areas, less than 500 pixels, are removed. This minimum area, however, also relied on the resolution of input image. Before the remaining contours are converted into a polygon using Shapely, their Y coordinates are flipped to conform to the standard Cartesian coordinate system used in GIS platforms and in mathematics, and any wiggles in the remaining contours are cleared through contour approximation with a certain epsilon. The polygons are simplified again with the simplify and buffer method from Shapely and rectangularized, ensuring the symmetric shape of the polygons and removing redundant vertices. The parameters used in these Shapely functions have a similar effect to the epsilon parameter in contour approximation, controlling the balance between simplification and shape fidelity. OCR reads the room numbers in the image and assigns the room number if it is inside the room polygon. All room polygons in the same image or floor are extracted into each layer of a geopackage file per building.

3.2 Georeferencing Floor Plan

Coordinate transformation typically involves three fundamental steps: scaling, rotation, and translation (Wolf et al., 2014). Georeferencing is initialized by matching the CRS, then computing the orientation of the floor plan and parcel polygon using their respective MBR. The computation involves creating a convex hull to simplify geometry, where each pair of consecutive vertices is extracted from its exterior coordinates to form an edge. For each edge, its x and y differences (dx,dy) between the pair of vertices and their arc-tangent are calculated to determine its angle or direction towards the horizontal line (x-axis). The original polygon is virtually rotated by the negative of that angle to align that particular edge horizontally. After each rotation, the area of the bounding box is calculated, which varies depending on the angle of rotation. The angle with the smallest bounding box area is chosen, representing the most compact and efficient rectangular representation of the shape. The difference between the MBR angle of the cadastral and the floor plan reveals how much the floor plan must be rotated to align with the cadastral. After the floor plan is rotated at this angle, the bounding box of the rotated floor plan and the parcel polygon are recalculated to measure the x and y scale factor by comparing their width and height extent. Following this, translation offsets are computed by aligning the lower-left corners of the latest bounding boxes to shift the floor plan exactly at the cadastral footprint. The polygons are then georeferenced based on the cadastral polygon downloaded from PDOK.nl using minimum bounding rectangle (MBR) calculation.

3.3 Combining Multiversion AHN Point Cloud

Combining multiple versions of AHN can overcome the occlusion of the latest version of AHN data as demonstrated in Figure 2. In this example, missing sections of a building in one version are completed by corresponding parts from another version. This improvement is due to the variations in the flight path during AHN laser scanning, resulting in different parts of the building being scanned. Therefore, the AHN from multiple versions, from AHN 1 to AHN 5, would be combined together and cropped with 1-meter buffer towards the building footprint



(a) AHN5 (b) AHN 1–5

Figure 2. Comparison of AHN datasets: (a) AHN5 and (b) AHN 1–5

obtained from pdok.nl to retrieve the building points. A segmentation process should be implemented to distinguish outside walls and roofs of building points.

3.4 Ground Classification using Cloth Simulation Filter (CSF)

Ground points are filtered using the CSF through filters.csf feature in PDAL. CSF is based on simulating a simple physical process to extract ground points from LiDAR points. It inverted the original point cloud, and then a rigid grid called cloth was dropped onto the inverted surface from above. The interactions between the nodes of the cloth and the corresponding point clouds can determine the final shape of the cloth to distinguish point clouds into ground and non-ground points. There are user-defined parameters: resolution, which represents grid resolution or cell size; step or time step, which adjusts the translation of points due to gravity during each iteration; and rigidness, which determines the stiffness of the cloth, where a higher value is preferred for flat terrain while a lower value is suggested for steep slopes (Zhang et al., 2016).

3.5 Segmenting AHN Point Cloud

Segmentation is grouping several homogeneous points based on their common features. RANdom SAmple Consensus (RANSAC) is a model-fitting method that uses a mathematical representation. It defines a model parameter from a minimum sample of random points. Iteratively checking their neighboring points, then a consensus set is formed if they are a match. The non-ground points identified by the CSF filter are iteratively segmented into individual planar patches using RANSAC, implemented via the segment_plane function from Open3D, with a distance threshold of 0.3 meters and a minimum of 3 points. Normal for each point in the plane is estimated through Open3D's estimate_normal function and converted as a numpy array with Numpy's asarray to compute the angle between the normal vector and the vertical Z-axis using the inverse cosine (arccos). By calculating the angle of arccos degree, the points can be classified into Flat Roof if the angle of normal is below 25°, Sloped Roof if the angle is between 25 and 60°, and Wall if the normal is more than 60 °.

3.6 Generating Synthetic Point Cloud

To construct a synthetic building point cloud, it initially load a floor plan layer from a Geopackage input file and iterates through each room polygon, searching the boundary using polygon.exterior and calculate its boundary to estimate how many points are needed to be created based on the user-defined point density, where 20 points per square meter as point density means 0.05 m spacing between points. Following that, it iterates over the number of points and, by using linestring.interpolate on the

boundary line, it returns a point with 0.05 spacing, generating points with even spacing on the boundary line of every room polygon. Each point is then iterated and numpy.arrange(start, stop, step) is implemented to generate points along the z-axis from the floor to the ceiling with the height of the floor as start, the height of the ceiling plus the spacing as stop, and the spacing as step. The 3D coordinates along their attributes are stored as well inside this loop, creating the wall points that are generated vertically along the boundaries of the polygon from the floor to the ceiling. The height for the ground floor is calculated based on the ground point from AHN, while the height for the ceiling is computed based on the average height of the roof point in AHN divided by the total number of floors above the ground.

The bounding box for every room polygon is computed to generate a grid of points within the interior of the polygons, creating a floor and a ceiling with different z coordinates. After it iterates through all floor layers, all the points are combined into a new pandas.DataFrame along their attributes, then exported into a LAZ 1.4 point cloud file with the same CRS as the AHN file. Each point is assigned a unique ID along their detected cadasral ID which is also used as classification for the LAZ file. The entire process of this step is outlined in

3.7 Point Cloud Alignment

After the synthetic point cloud is generated, it will be aligned to AHN using ICP. ICP is a popular spatial registration-based method to align two point cloud datasets. The algorithm operates over two main steps: first, it initially finds correspondences between the target point cloud and the source point cloud by finding the nearest neighbor in Euclidean space; second, given these correspondences, it iteratively estimates the optimal rigid transformation that best aligns the source to the target by minimizing cost function (the sum of squared distance between matched pairs) until convergence or the value is less than the threshold. This algorithm is known as point-to-point ICP with equation as follow:

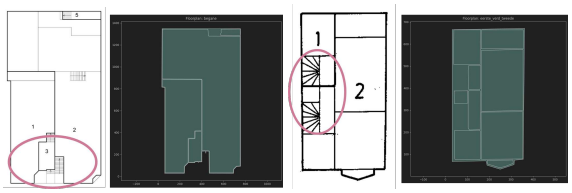
$$E(\mathbf{T}) = \sum_{(\mathbf{p},\mathbf{q})\in\mathcal{K}} ((\mathbf{p} - \mathbf{Tq}) \cdot \mathbf{n_p})^2 \quad (1)$$

The other ICP variant, point-to-plane ICP, uses the intersection of the normal point in both datasets to determine the corresponding points. To increase convergence speed, the cost function is improved by replacing point-to-point distances with point-to-plane distances, which minimizes the distance between the source point and the tangent plane of the corresponding target point (Wang and Zhao, 2017). The formula for this method:

$$E(\mathbf{T}) = \sum_{(\mathbf{p},\mathbf{q})\in\mathcal{K}} \|\mathbf{p} - \mathbf{Tq}\|^2 \quad (2)$$

After the alignment, both point cloud, AHN and synthetic floor plan point cloud will be combined into one LAS file with the same header as the latter to preserve the generated attributes. Since AHN is the envelope of the building, the kadaster ID for AHN would be the same as the outer wall in synthetic points, which is the name plus 0, as there is no cadastral number.

4. Results and Discussion



(a) Sample 2 (b) Sample 3

Figure 3. Noises in Image that needed to be cleaned manually

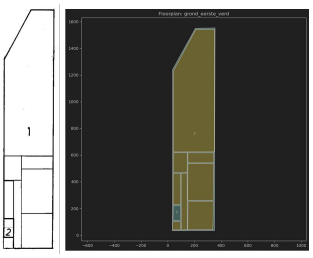
During the preprocessing of the image, OCR can read the floor label that is associated with keywords such as ”begane” and ”verdieping”; as a result, the contour block for each floor is able to be generated. The interior of the apartment can be detected and vectorized into geometric polygons. Cadastral apartment drawings depict cadastral boundaries with thicker lines and room segmentation with thinner lines. The algorithm is able to differentiate the thicker and thinner lines in the newest cadastral drawing, thus generating cadastral boundaries without room segmentation; on the contrary, for old cadastral drawings, Sample 1 and Sample 3, the polygons are generated from all room segmentation, not cadastral boundaries, as the thick lines are hard to distinguish even by eyesight. Some input images need to be cleaned manually using an image editor due to inconsistent lines or stair areas that cannot be detected during the automatic cleaning process. As Sample 1 does not require any manual cleaning, Sample 2 in Figure 3 shows that stairs and annotations in the drawing create noises, and inconsistent width boundaries lead to lines not being generated, while stairs in Sample 3 prevent room segmentation. The parameters must also be tuned for different drawing files, as each file may vary in resolution, style, and quality. The kernel size affects the result, where a higher kernel size in the open kernel removes more and larger noise. In contrast, in the close kernel, it connects larger gaps, which also influences thicker lines to be generated instead of thinner ones. A higher epsilon results in simpler contours with fewer vertices, while a lower epsilon retains more detail but may introduce jagged or overly complex geometries. This effect also applies to the parameters in simplify and buffer. The specific parameter values used are listed in Table 1 below.

	Parameter		
Process	Sample 1	Sample 2	Sample 3
open kernel	3x3	2x2	2x2
open iteration	3	2	1
close kernel	20x20	10x10	5x5
close iteration	1	2	1
epsilon	0.013	0.001	0.005
simplify	10	1	9
buffer	8	5	5

Table 1. Parameter during Vectorization

(a) Vectorization Result in Sample 1

(b) Vectorization Result in Sample 2



(c) Vectorization Result in Sample 3

Figure 4. Vectorization Result in All Samples

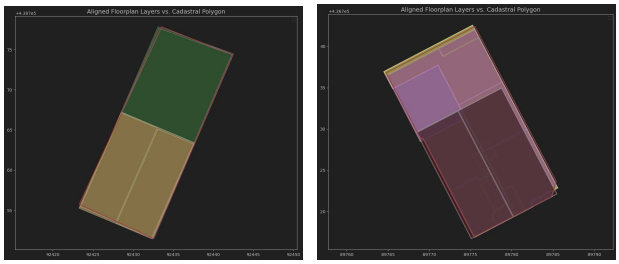
As described in previous section, the georeferencing algorithm estimates transformation parameters, including rotation, scaling, and translation. Although the georefencing algorithm uses a simple calculation, it presents an adequate result where the polygon is located in the same place as the cadastral boundary, as illustrated in Figure 5. The RMSE is also below half a meter, as shown in the Table 2. However, for certain cases, one must add additional rotations in the code input parameter, where the value of the input degree is tuned manually based on the orientation and shape of the vectorized polygon. For instance, since Sample 1 has a rectangular shape, which has rotational symmetry, it may need to be flipped.

	RMSE (cm)
Sample 1	32.18
Sample 2	18.25
Sample 3	25.71

Table 2. RMSE of Georeferencing

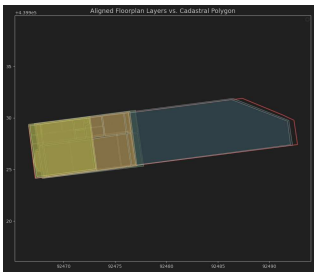
Combining multiple AHN versions can overcome occlusion in the AHN as it provides more points for the building, as can be seen in Figure 6; however, wall points are still sparse, and some parts are still missing. Another problem is that the buildings are row houses, and they were located between other units as depicted in Figure 7; therefore, the surrounding walls, particularly the shared or common wall, were impossible to acquire by LiDAR scanning.

Ground points are effectively extracted from the complete building point clouds using the CSF algorithm, configured with default parameter values suitable for moderately flat terrain. The resulting ground points are visualized as blue-colored points in Figure 8. However, during subsequent segmentation, distinguishing non-ground points, specifically separating wall and roof components (shown in red and white, respectively), remains challenging. This is particularly evident in cases involving sparse wall points and sloped roofs, such as in Sample 3 (Figure 8c). Additionally, some outliers and vegetation points



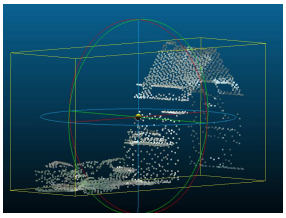
(a) All Georeferenced Floor plan in Sample 1

(b) All Georeferenced Floor plan in Sample 2

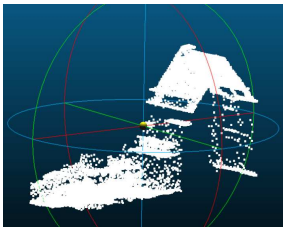


(c) All Georeferenced Floor plan in Sample 3

Figure 5. Overview of Georeferencing Results



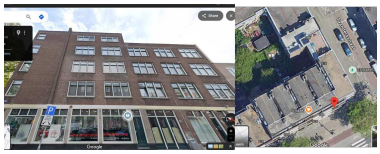
(a) AHN 5



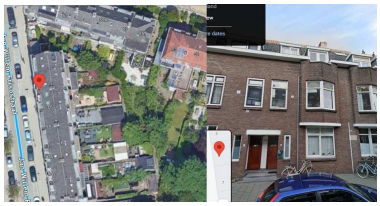
(b) Combination of All Versions of AHN

Figure 6. Comparison of AHN 5 and Combination in Sample 3

(a) Building location for Sample 1

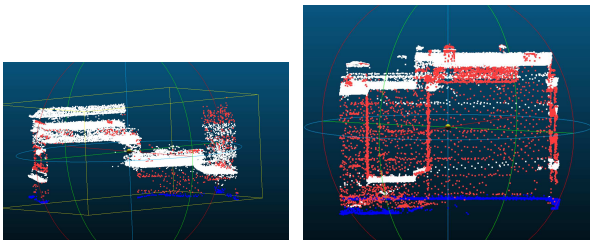


(b) Building location for Sample 2

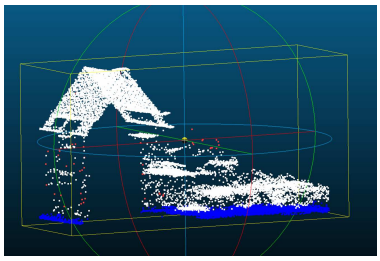


(c) Building location for Sample 3

Figure 7. Building location for All Samples



(a) Segmentation in Sample 1 (b) Segmentation in Sample 2



(c) Segmentation in Sample 3

Figure 8. Result of Segmentation in All Samples

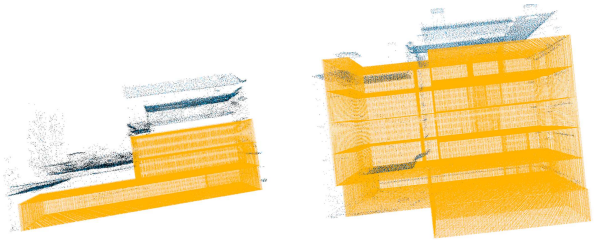
persist after classification, as seen in Sample 1 (Figure 8a), indicating limitations in the accuracy of the AHN-based classification.

The height of the ground floor fits the AHN as it uses the z value of ground points. However, the floor height does not seem to correctly conform to AHN, due to some misclassified roof points, as mentioned before in the previous step.

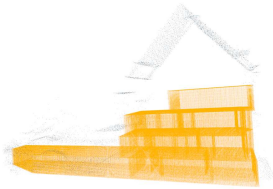
		Sample 1	Sample 2	Sample 3
Initial	Fitness	3.7e-05	6.92e-05	1.31e-04
	RMSE	1.406	1.478	1.418
	Correspondences	44	249	141
Point-to-Point ICP	Fitness	5.22e-05	8.73e-05	1.89e-04
	RMSE	1.480	1.480	1.459
	Correspondences	62	314	141
Point-to-Plane ICP	Fitness	0	7.43e-05	0
	RMSE	0	1.537	0
	Correspondences	0	267	0

Table 3. RMSE of ICP in cm

Although the floor plan polygons have been georeferenced based on the parcel polygon, and the generated point clouds from the floor plan are aligned with AHN through ICP, the highly accurate position is still hard to acquire, with an RMSE between 1.3 and 1.6 cm as can be seen in Table 3. This is due to sparse points in AHN that affect the performance of ICP. Point-to-point needs to find the corresponding point between the datasets; thus, it would be challenging if no matching points are available. Meanwhile, point-to-plane exploits normal calculation, which also becomes problematic if the surrounding neighbour points are not adequate to correctly calculate the normal for each point. For that reason, Point-to-Plane ICP only performs better for Sample 2 (Figure 9b), which has more correspondence points, while it fails completely for Sample 1 and Sample 3 (Figure 9a and 9c), which have fewer correspondence points. The algorithm will automatically use the ICP method that has lower RMSE and higher number of correspondences points. Point-to-Point ICP is preferred for Sample 1 and Sample 3, and Point-to-Plane ICP is opted for Sample 2. Although all the resulting RMSEs are slightly higher or worse than the initial for the three cases, the values for correspondences increase, depicting a greater number of matched point pairs between the source and target after alignment. This also leads to a slightly higher fitness value, which means the proportion of total source points that matched within a threshold of 2 cm.



(a) ICP Result in Sample 1 (b) ICP Result in Sample 2



(c) ICP Result in Sample 3

Figure 9. Comparison of ICP results in three samples.

After aligning the synthetic point cloud to AHN, both datasets are combined into one LAS file for each sample and uploaded into Cesium Ion. To deliver a real-world representation, integrating other datasets, including reference objects and a topography map, can offer a reference to interpret the parcel in terms of location and size (Cemellini, 2018; Kalogianni, 2016). Since the 3D parcel is represented as a point cloud, the AHN dataset can serve as a reference object, enabling seamless integration of spatial data, as illustrated in Figure 10.

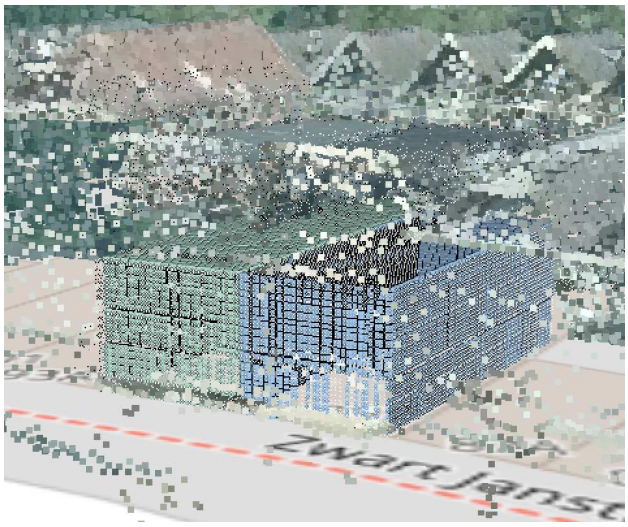
Overall, the proposed pipeline requires approximately between 44 and 97 seconds per sample from processing the floor plan to generating the synthetic building point cloud, as detailed in Table 4. However, this estimate does not account for manual interventions that may be necessary in certain cases, such as image noise cleaning, parameter tuning, and missing cadastral number assignment. Despite these exceptions, the pipeline demonstrates sufficient efficiency for large-scale or nationwide implementation, provided that some manual input is accommodated when necessary.

Process	**Sample 1**	**Sample 2**	**Sample 3**
Preprocessing	7	7	8
Vectorize	8	22	10
Georeference	1	2	1
Crop AHN	13	15	14
Segment AHN	5	6	6
Construct PC	9	31	4
Align & combine AHN	2	14	1
Total	**45**	**97**	**44**

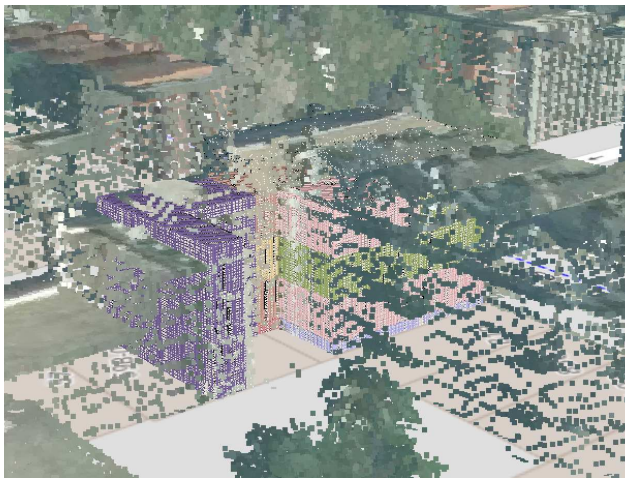
Table 4. Processing times across samples (in seconds).

5. Conclusion

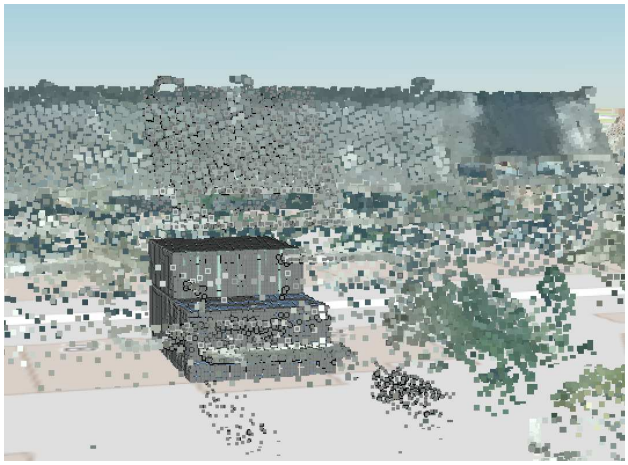
Instead of using a BIM model, this project presents an alternative using point clouds as 3D spatial units in a LAS. By com-



(a) Visualization of Sample 1



(b) Visualization of Sample 2



(c) Visualization of Sample 3

Figure 10. Visualization of all three samples.

bining cadastral drawings and a point cloud nationwide dataset, AHN, the apartment complexes with their own spatial units can be generated in this framework without additional survey or an existing BIM model. The floor plan parsing algorithm successfully detects and vectorizes the interior layouts of all three cadastral apartment samples into geometric polygons. The pipeline manages to align the vectorized polygons to the cadastral polygons with RMSEs of 18-32 cm and aligns the generated synthetic point cloud to the combined version of AHN using ICP, achieving RMSEs between 1.48 and 1.51 cm with 62-298 matching points due to the sparse AHN points on building facades. These error measurements are close to the current cadastral map in the Netherlands that has a graphic quality accuracy where the standard deviation of boundaries is 20 cm for urban areas and 40 cm for rural areas (Hagemans, 2024).

Although point clouds are widely used as input data, compared to mesh building, they can enable the seamless integration of real-world features provided from AHN such as building facades, walls, and fences, which often delineate cadastral boundaries. Another advantage is their ability to preserve geometric representation that can be directly compared to cadastral reference points measured with GNSS. Moreover, the system is capable of generating a synthetic building point cloud in under two minutes per sample, indicating the feasibility of future nationwide implementation. However, some limitations are found in this study that require further improvement, particularly some manual intervention (e.g. the removal of stairs, OCR not recognizing all labels, etc.).

6. Future Work

Future research directions include the following:

1. **Scaling and Algorithmic Robustness**: Expand to large-scale pilot areas and more diverse floor plans to improve robustness. Develop deep learning methods for floor plan parsing and enhance point cloud segmentation to reconstruct more complex and realistic building geometries.

2. **International Applicability**: Explore applicability in other countries by addressing differences in cadastral drawing formats and assessing the availability or alternatives to nationwide point cloud datasets.

3. **Accuracy Evaluation for 3D LAS**: Conduct ground-truth validation using GNSS-based cadastral reference points. Improve accuracy through occlusion correction (e.g., (Balado et al., 2019), alternative alignment methods, and integration of higher-resolution LiDAR (e.g., drone ALS, TLS, MLS).

7. Acknowledgements

We gratefully acknowledge Kadaster for providing the floor plan data. We also used ChatGPT to enhance the readability and clarity of this document.

References

AHN, 2020. Kwaliteitsbeschrijving. Publisher: AHN.

Alonso, R., Borras, M., Koppelaar, R. H. E. M., Lodigiani, A., Loscos, E., Yöntem, E., 2019. SPHERE: BIM Digital Twin Platform. *Sustainable Places 2019*, MDPI, 9.

Baauw, M., 2021. Maintaining an up to date digital twin by direct use of point cloud data. Master's thesis, Delft University of Technology.

Balado, J., Díaz-Vilariño, L., Arias, P., Lorenzo, H., 2019. Point clouds for direct pedestrian pathfinding in urban environments. *ISPRS Journal of Photogrammetry and Remote Sensing*, 148, 184–196. https://www.sciencedirect.com/science/article/pii/S0924271619300048.

Bydłosz, J., Warchoł, A., Balawejder, M., Bieda, A., 2021. Practical verification of Polish 3D cadastral model. https://doi.org/10.4233/uuid:884b0c33-0d8e-40fd-bb88-669b21798a65.

Cemellini, B., 2018. Web-based visualization of 3D cadastre. Technical report.

Hagemans, E., 2024. Development in Cadastral Surveying and Mapping in the Netherlands: About the Improved Cadastral Map of The Netherlands: Kadastrale Kaart Next. *Kart og Plan*, 117(2), 230–240. https://www.scup.com/doi/10.18261/kp.117.2.10.

Kalogianni, E., 2016. Linking the Legal with the Physical Reality of 3D Objects in the Context of Land Administration Domain Model (LADM). Master's thesis, Delft University of Technology Delft, The Netherlands.

Koeva, M., Nikoohemat, S., Elberink, S. O., Morales, J., Lemmen, C., Zevenbergen, J., 2019. Towards 3D indoor cadastre based on change detection from point clouds. *Remote Sensing*, 11(17). Publisher: MDPI AG.

Koninkrijksrelaties, M. v. B. Z. e., n.d. Uitvoeringsregeling Kadasterwet 1994. Last Modified: 2024-02-24.

Mao, P., Oosterom, P. v., Rafiee, A., 2024. A digital twin based on Land Administration.

Meulmeester, R. W. E., 2019. BIM Legal: Proposal for defining legal spaces for apart-ment rights in the Dutch cadastre using the IFC data model.

Nguyen, T. D., Adhikari, S., 2023. The Role of BIM in Integrating Digital Twin in Building Construction: A Literature Review. *Sustainability*, 15(13), 10462. https://www.mdpi.com/2071-1050/15/13/10462. Number: 13 Publisher: Multidisciplinary Digital Publishing Institute.

Nottrot, B., Folmer, E., Roy, D., Scheer, B., Merx, P., 2023. Multi-unit building address geocoding: An approach without indoor location reference data. *Transactions in GIS*, 27(1), 57–83. https://onlinelibrary.wiley.com/doi/10.1111/tgis.13017.

Oosterom, P. v., Stoter, J., Ploeger, H., Thompson, R., Karki, S., 2011. World-wide Inventory of the Status of 3D Cadastres in 2010 and Expectations for 2014. *Bridging the Gap between Cultures*.

Pouliot, J., Ellul, C., Hubert, F., Wang, C., Rajabifard, A., Kalantari, M., Shojaei, D., Atazadeh, B., Oosterom, P. V., 2018. 3D Cadastres Best Practices, Chapter 5: Visualization and New Opportunities.

Wang, F., Zhao, Z., 2017. A survey of iterative closest point algorithm. *2017 Chinese Automation Congress (CAC)*, 4395–4399.

Wolf, P. R., Dewitt, B. A., Wilkinson, B. E., 2014. Coordinate Transformations. 4th edition edn, McGraw-Hill Education, New York.

Yin, X., Wonka, P., Razdan, A., 2009. Generating 3D Building Models from Architectural Drawings: A Survey. *IEEE Computer Graphics and Applications*, 29(1), 20–30. https://ieeexplore.ieee.org/document/4736453/?arnumber=4736453. Conference Name: IEEE Computer Graphics and Applications.

Zhang, W., Qi, J., Wan, P., Wang, H., Xie, D., Wang, X., Yan, G., 2016. An Easy-to-Use Airborne LiDAR Data Filtering Method Based on Cloth Simulation. *Remote Sensing*, 8(6), 501. https://www.mdpi.com/2072-4292/8/6/501. Number: 6 Publisher: Multidisciplinary Digital Publishing Institute.