# An unsupervised, open-source workflow for 2D and 3D building mapping from airborne LiDAR data

Hunsoo Song and Jinha Jung, *Member, IEEE*

***Abstract*—This study introduces an automated, open-source workflow for large-scale 2D and 3D building mapping using airborne LiDAR data. Uniquely, our workflow operates entirely unsupervised, eliminating the need for any training procedures. We have integrated a specially tailored digital terrain model generation algorithm into our workflow to prevent errors in complex urban landscapes, especially around highways and overpasses. Through fine rasterization of LiDAR point clouds, we've enhanced building-tree differentiation. Additionally, we've reduced errors near water bodies and augmented computational efficiency by introducing a new planarity calculation. Our workflow offers a practical and scalable solution for the mass production of rasterized 2D and 3D building maps from raw airborne LiDAR data. Our method's robustness has been rigorously validated across a diverse dataset in comparison with deep learning-based and hand-digitized products. Through these extensive comparisons, we provide a valuable analysis of building maps generated via different methodologies. We anticipate that our highly scalable building mapping workflow will facilitate the production of reliable 2D and 3D building maps, fostering advances in large-scale urban analysis. The source code for our workflow is publicly accessible at: https://github.com/hunsoosong/airborne-lidar-building-mapping.**

***Index Terms*—Building mapping, airborne laser scanning, unsupervised, 3D building, open-source**

## I. Introduction

### A. Current State of Large-area 2D Building Mapping

Buildings are key structures in which numerous human activities unfold. They offer invaluable insights into human practices and the subsequent environmental impacts [1]. Building maps, especially those derived from remote sensing imagery, are crucial to numerous fields, including disaster management, urban ecology, smart city planning, population estimation, and humanitarian aid [2]–[7]. However, the inherent uncertainties and errors in building maps can mislead studies reliant on them. Thus, the remote sensing community is dedicated to enhancing the quality of such maps.

Publicly available building maps typically provide only 2D information due to dependence on optical imagery. For instance, OpenStreetMap (OSM) provides extensive spatial coverage but lacks 3D information and suffers from inconsistencies in completeness, accuracy, and data vintage [8], [9]. Authoritative maps, often seen as ground-truth [10], [11], guarantee accuracy but typically lack 3D information and are limited in spatial coverage due to high production costs.

Since the rise of deep learning in building extraction, highlighted by the SpaceNet challenge [12], deep learning-based methods have dominated recent literature [13]–[15]. Yet, deep model performance can vary based on training conditions and target areas [16], [17]. Thus, creating a universally dependable model remains elusive, despite advances in domain adaptation [17], [18] and sample refinement [19].

One significant milestone in large-area building mapping was the release of Microsoft Building Footprints [20]. This project produced the largest machine-generated building footprints, utilizing the EfficientNet model [21] trained on millions of building labels and corresponding satellite images. While this initiative marked a major advancement by offering the first continental-scale open-building maps, the quality can be inconsistent because it used diverse optical images with varying conditions for the mapping, similar to the challenges faced by OpenStreetMap [22]. Given that the state-of-the-art accuracy of deep learning-based mapping with optical imagery doesn't yet match authoritative maps—even with millions of labels—it underscores the need for further innovations to bridge the accuracy gap.

### B. Towards Large-area 3D Building Mapping

Beyond the concern of building map accuracy, it's essential to recognize that buildings are inherently 3D entities. As cities become denser and buildings taller, a 3D perspective is crucial for understanding urban developments and their environmental impacts [4], [23]. These maps enhance research in various areas such as urban climate studies [24], disaster management [25], population estimation [26], [27], and the assessment of other forms of land classification [28]. It can also foster new research domains in various smart city applications [29]–[33].

Airborne laser scanning (ALS) stands out as the prime technique for obtaining comprehensive 3D building details. Even with the associated costs, ALS stands as a preferable choice over optical or SAR sensors [23], [34]–[36], which often fall short in delivering precise and detailed 3D data. When in-depth 3D building data is essential and LiDAR promises more accurate 2D building delineations, ALS emerges as the preferred choice for both 2D and 3D building mapping. This is particularly due to the scalability and accuracy challenges observed with optical image-based deep learning approaches [12], [16]. Hence, if a robust workflow for 2D and 3D building mapping from ALS is established, LiDAR-based building mapping could be more appealing than other methods in terms of both accuracy and cost.

In recent decades, many building mapping algorithms using airborne LiDAR data have been developed. These include unsupervised and traditional machine-learning methods propelled

The authors are with the Lyles School of Civil Engineering, Purdue University, West Lafayette, IN 47907 USA (e-mail: hunsoo@purdue.edu; jinha@purdue.edu). *(Corresponding author: Jinha Jung.)*

by competitions, notably by the ISPRS community [37], [38]. While these methods have demonstrated effectiveness with smaller datasets, their scalability is uncertain due to the limited diversity and spatial extent of their study areas, raising transferability concerns [17], [39]. Also, the implementation details of many such algorithms are not publicly shared, hindering widespread application and further development. Furthermore, post the rise of deep learning, their performances have been observed to lag behind in general [40], [41]. Meanwhile, there's also a rising demand in extracting detailed building structures, like roofs and facades, for precise "3D modeling" [42]–[53]. While these methods yield refined 3D building models fit for standards like CityGML [54] or BIM [55], the intricate detail focus often compromises the scalability vital for large-area building mapping projects.

This paper presents an end-to-end, open-source workflow specifically designed for the mass production of rasterized 2D and 3D building maps. Utilizing discrete point cloud data from standard topography ALS, our workflow produces reliable 2D and 3D building maps without the need for repetitive parameter tuning and surpasses the accuracy of Microsoft Building Footprints. Operating unsupervised, it leverages the inherent physical properties of buildings: ground-standing objects with laser-impermeable and smooth surfaces. This physical-property-driven approach ensures consistent, reliable results and minimizes unexpected errors—a crucial aspect for large-scale mapping. Through rigorous validation with a vast and varied dataset, we confirmed that our approach is not only computationally but also operationally scalable, well-positioned to facilitate large-area 3D building mapping.

The main contributions can be summarized as follows.

- We present an open-source workflow that can facilitate the mass production of rasterized 2D and 3D building maps for large-area using airborne LiDAR data.
- Our physical-property-driven workflow produces building maps with greater accuracy than Microsoft Building Footprints, ensuring robust and consistent performance across varied landscapes without parameter tuning.
- Our study provides a comprehensive comparison of building maps derived from different methods: LiDAR-based unsupervised, deep learning with multiple sources (+ Microsoft Building Footprints), and hand-digitization.

The paper is structured as follows: Section II details the proposed workflow. Section III presents experimental results and in-depth error analysis. Section IV discusses computational complexity, parameter impact, and workflow limitations. Section V concludes the study.

## II. Methodology

### A. Overview

The proposed workflow generates rasterized 2D and 3D building maps from the raw ALS point clouds. The workflow operates in a simple but robust rule-based approach by exploiting the physical properties of buildings, which are ground-standing and have laser-impermeable, relatively smooth surfaces. The workflow has been optimized through iterative refinements. Figure 1 summarizes the workflow and its optimization procedures.

### B. The proposed workflow

*1) From ALS point clouds to Finely Rasterized DSM:* Our building mapping workflow starts from the raw point cloud sourced from a typical ALS system. These point clouds depict 3D coordinates captured by airborne LiDAR sensors. Given that these data points only represent a fraction of the earth's surface, there's a natural limitation to their representation.

Data points can be sparse due to factors like flight configuration and the laser's interaction with ground objects. To manage this inconsistency, point clouds are usually converted to a gridded format known as the digital surface model (DSM). However, during this transformation, certain areas, or grids, might not register a LiDAR point. A common method to avoid these gaps is to use a coarser ground sampling distance (GSD) ensuring most grids have a LiDAR point [40], [56]–[58]. Yet, generating a DSM with a coarse grid can result in data loss and blur distinctions between buildings and trees.

Our approach employs "fine rasterization" for DSM creation. This method projects point clouds onto finely gridded surfaces and fills in gaps later. Even if this means more gaps initially, it helps prevent data loss from overlapping LiDAR points that happens more when we rasterize DSM with a coarse grid for. Also, when multiple points land on a single grid, we opt for the lowest elevation point. This optimizes the distinction between buildings and trees, especially when lasers penetrate through tree canopies.

The nuances of DSM, as influenced by the rasterization technique, are exemplified in Figure 2. Figure 2(a) is an RGB image sourced from Google Earth. Figure 2(b) displays the LiDAR point occupancy, highlighting grids occupied by LiDAR points in white, with unoccupied grids in black. These black grids indicate the DSM's void areas before interpolation. Figure 2(c-d) contrast the DSMs produced by coarse and fine rasterizations. The former takes the highest elevation when multiple points overlap, while the latter chooses the lowest. Each DSM has a GSD of 2-meter and 0.5-meter, respectively. The images clearly depict the challenge of coarse rasterization in differentiating between buildings and trees. Conversely, fine rasterization retains building shapes more accurately and depicts trees as distinct, scattered points, emphasizing the penetration difference between buildings and trees.

*2) From Finely Rasterized DSM to DTM* and NDHM:* Our workflow characterizes a building as an object that stands on the ground. This means that buildings are relatively tall and display a distinct height difference from the adjacent ground at their boundaries. Simply calculating the relative height above the nearby ground and applying a mask based on a certain height elevation can effectively identify building candidates. This identification is facilitated by creating a digital terrain model (DTM). When this DTM is subtracted from the DSM, the result is a normalized digital height model (NDHM), representing height above ground.

Our workflow adopts a DTM generation method, "DTM*", from [59], chosen for its computational efficiency and ability to preserve detailed object boundaries. DTM* classifies an object as a region fully enclosed by steep slopes. The process initiates with the creation of a break-line map

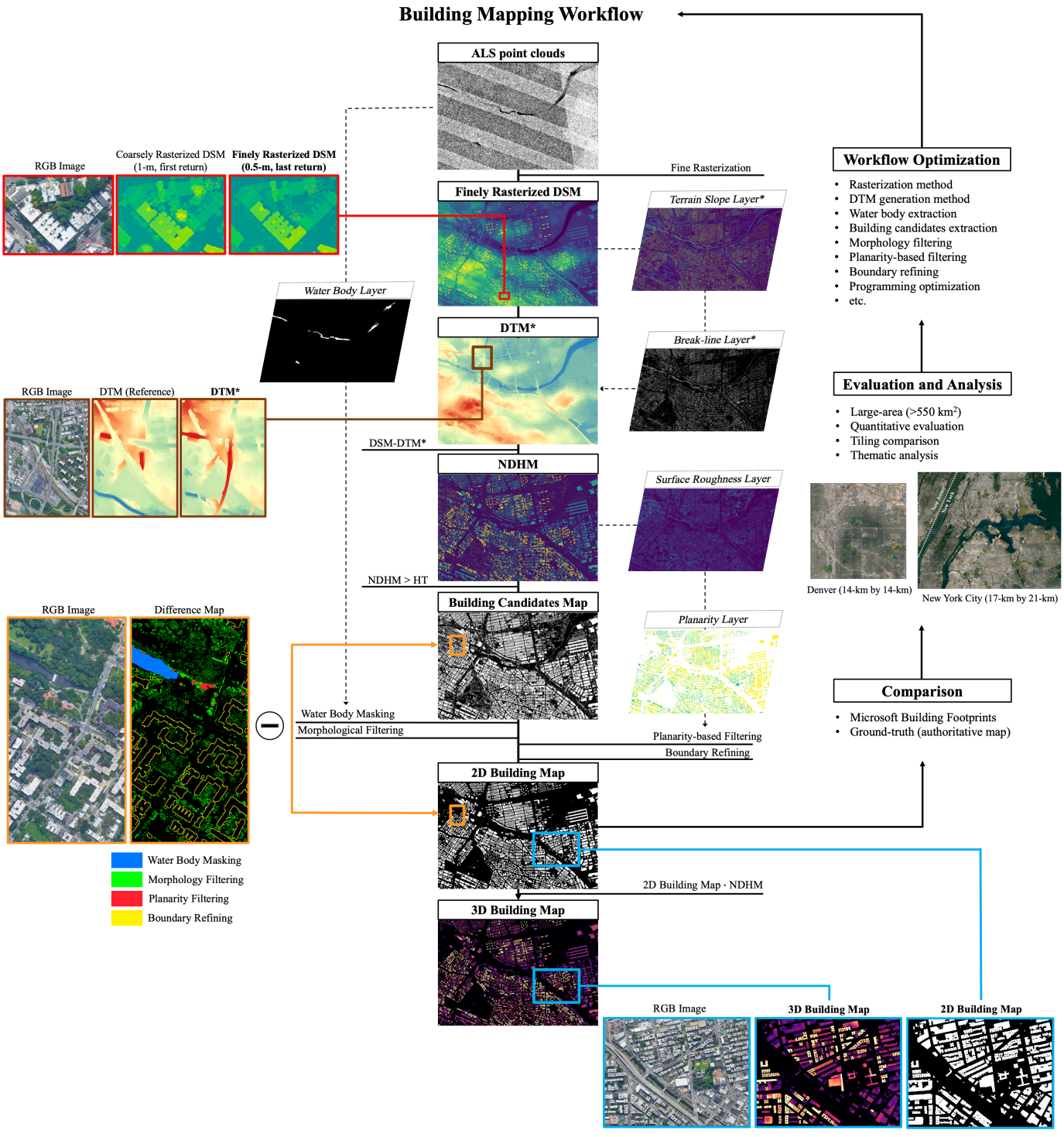



Fig. 1: Overview of the methodology

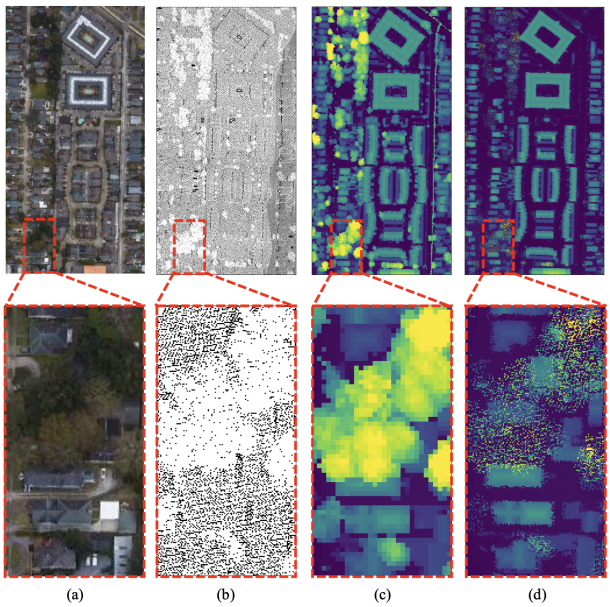



Fig. 2: Comparison of two DSM rasterization methods: (a) RGB image, (b) LiDAR occupancy map, (c) coarsely rasterized DSM, (d) finely rasterized DSM.

that delineates steep slopes. Following this, a connected component algorithm is employed, filtering regions fully surrounded by steep slopes as objects while ensuring all ground sections are interconnected smoothly. This unique object-wise filtering performed by DTM*, which outlines the boundaries of ground-standing objects using the same DSM, enhances computational efficiency and helps to avert common errors often associated with large buildings and their boundary definitions. Another crucial reason for adopting DTM* is that it considers bridges and overpasses as ground, unlike typical DTM generation methods [60]. Figure 3 showcases a comparison between DTM* and a reference DTM ("DTM(R)") obtained from the U.S. Geological Survey. DTM* uniquely identifies overpasses as ground, setting them apart from buildings in the subsequent NDHM and Building Candidate Maps, a distinction not made by the reference DTM. This distinction is vital, as, without it, bridges and overpasses might be mistakenly identified as buildings in subsequent workflow stages.

*3) From NDHM to Building Candidates Map, 2D and 3D Building Maps:* Building Candidates Map is generated by applying a certain Height Threshold (HT) to the NDHM. Building Candidates Map will represent all objects that are relatively taller than the nearby ground as a binary format, and the objects might be buildings, non-building entities, or noise. To distill actual buildings from these candidates, our workflow executes four sequential operation: (1) water body masking, (2) morphological filtering, (3) planarity-based filtering, (4) boundary refining.

The first operation is the water body masking. This is essential to filter out noise from water bodies that might mistakenly be identified as buildings. This method employs a concept similar to the studies in [61], [62], leveraging the lower point density over water surfaces. However, in urban settings, tall structures can similarly reduce point density due to occlusion. Recognizing this, we've incorporated specific adjustments. Our methodology classifies surface water based on local LiDAR point density within a 9 by 9 window on a 0.5-m resolution DSM. If the center pixel's density falls 2 sigma below the average, it's marked as water. We've enhanced this process with two further rules: firstly, we eliminate smaller water bodies (those under 1,000 $m^2$) from masking, since they're likely obscured by tall buildings. Secondly, we enforce a 5-meter water buffer on larger water bodies, minimizing the risk of misidentifying buildings due to adjacent water noise.

The second operation utilizes morphological filtering, capitalizing on size differences between buildings and relatively smaller entities such as trees. Due to LiDAR's ability to penetrate trees but not solid building roofs, trees appear as noise in the Building Candidates Map, a contrast emphasized with fine rasterization, as shown in Figure 2. Therefore, we apply erosion to the Building Candidates Map to remove small objects, followed by dilation to restore any remaining eroded pixels. Since mapping is binary, balancing omission and commission errors is key. The filter's kernel size (K1) affects this balance: a larger K1 reduces commission errors but can erase small buildings, while a smaller kernel does the opposite. We discuss K1's implications in Section IV-B.

The third operation is planarity-based filtering, leveraging the relatively consistent height feature of buildings in NDHM. While previous methods have employed co-occurrence matrix-based [63], [64], eigenvalue-based [65], [66], and Entrophy-based [40] strategies, we've used a more efficient approach to determine local height variations. Firstly, our workflow rounds the NDHM to integer values and counts the unique integers within a square kernel (K2) on the NDHM, creating a surface roughness layer as illustrated in Figure 1. Each pixel in this layer shows the number of unique integer values in a K2-sized window. A pixel is marked "planar" if its roughness falls below the Roughness Threshold (RT). For instance, with K2 set at 5 and RT at 4, if a 5 by 5 window on the rounded NDHM has fewer than 4 unique integers, the central pixel is designated planar. The algorithm then determines the "planarity" for each building candidate, which is the proportion of planar pixels in the candidate. The planarity layer in Figure 1 displays these values. Assuming that building roofs have a planar surface compared to non-building objects, we employ a planarity-based filtering algorithm to exclude objects with planarity values less than a specified ratio, termed the Dense Tree (DT) value. This value is employed as a descriptor to differentiate dense trees from building candidates. A more detailed description of DT is provided in Section IV-B.

The fourth operation is boundary refining. We apply a dilation kernel of size K3 to refine LiDAR's underestimated building boundaries. This deformation and underestimation are detailed in Section IV-B. Although intricate boundary refining methods [33], [67], [68] might delineate building boundaries more precisely, often at the expense of requiring shape constraints, we prioritize efficiency and scalability by opting for the simpler dilation approach.

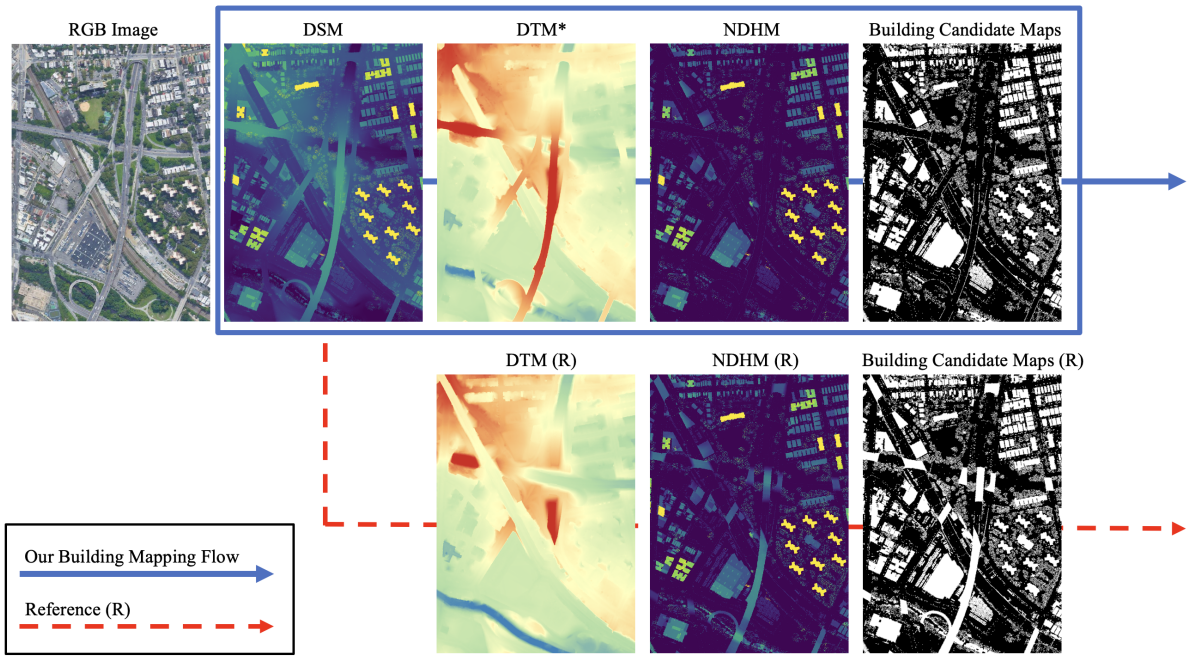



Fig. 3: Comparison of DTM* and a reference DTM and their respective results. DTM* can prevent overpasses from becoming building candidates.

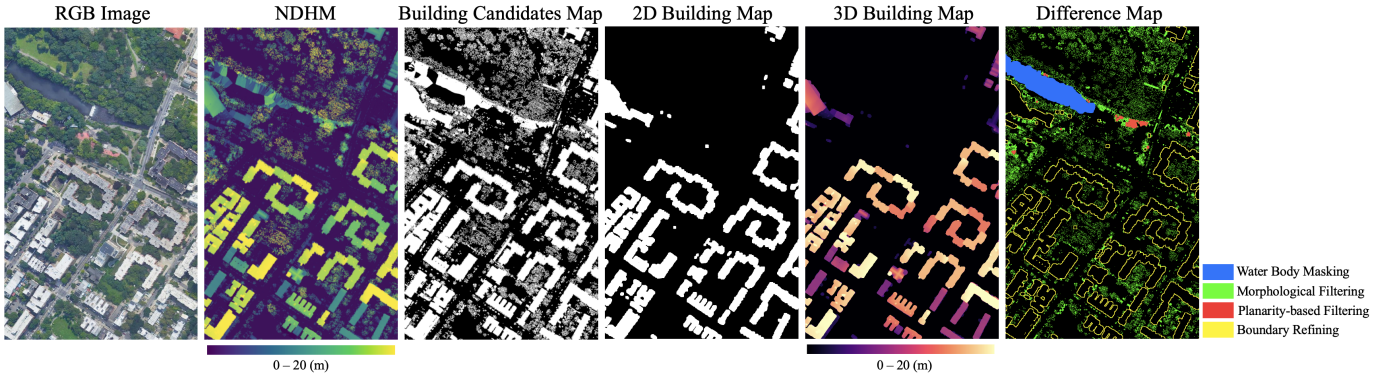



Fig. 4: From NDHM to Building Candidates Map, 2D and 3D Building Maps. The color in Difference Map indicates which operation caused the difference between Building Candidates Map and 2D Building Map.

Finally, the workflow generates a 3D building map by extracting building pixels of the 2D building map from the NDHM. Figure 4 shows a part of the workflow from NDHM to 2D and 3D Building Maps. "Difference Map" represents the difference between Building Candidates Map and the 2D Building Map. The color in Difference Map indicates which operation caused that difference.

By default, our workflow generates building maps of 0.5-meter resolution as default, and its parameters have been optimized through extensive experiments. HT was set as 1.5 meters. The kernel size (K1) of morphological filters was set as 7 (a 7 by 7 pixels window). A 5 by 5 pixels window was used for K2. RT and DT were set as 4 and 0.1, respectively. K3 was set as a 5 by 5 pixels window by default. While the optimal parameter combination may differ based on the specific dataset, these default settings have proven to be robust.

## III. Experimental Results

### A. Optimization and Evaluation

We tested the proposed workflow on datasets from Denver, Colorado (Figure 5), and New York City (NYC), New York (Figure 6), covering an area exceeding 550 $km^2$. These datasets included authoritative building maps and Microsoft Building Footprints for comparison. The LiDAR data were sourced from the U.S. Geological Survey's 3DEP program.

The vastness of our study area made the conventional evaluation - providing averaged metrics across the entire area - less effective. Thus, in addition to the conventional evaluation metrics, we introduced a tiling comparison method for more comprehensive evaluation. This method segmented each city into tiles of 0.5 km by 0.5 km, yielding 784 tiles for Denver and 1428 for NYC. For each tile, we calculated the IoU and

Fig. 5: Study area of Denver (14-km by 14-km)

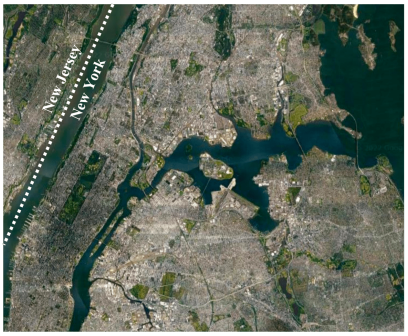



Fig. 6: Study area of New York City (NYC) (17-km by 21-km, New Jersey is excluded)

ranked them, identifying and emphasizing key disparities to enhance comparative analysis.

Three maps were compared: LiDAR building map (result of our workflow), Microsoft Building Footprints, and the authoritative ground-truth. Please note the time discrepancies across different maps. The details are summarized in Table I.

TABLE I: Summary of the Dataset

| **Parameter** | **Denver** | **NYC** |
|---|---|---|
| Area | 196 $km^2$ | 357 $km^2$ |
| LiDAR Sensor | Leica TerrainMapper | Leica ALS70 |
| Data Collection | May - Sep 2020 | Mar - Apr 2014 |
| Point Density | 4-pt/$m^2$ | 5-pt/$m^2$ |
| Ground-truth Vintage | 2018 | Feb 2022 |
| MS Footprints Vintage | 2018-2019 | Mostly 2019 |
| No. of Tiles | 784 | 1428 |

### *B. Results of the Denver dataset*

*1) Conventional quantitative results:* As shown in Table II, our workflow outperformed Microsoft Building Footprints in all metrics. Intersection over Union (IoU) and recall were particularly higher than that of Microsoft Building Footprints. Considering the LiDAR building map was generated in a fully unsupervised way with a single default parameter set, the result shows that our workflow can produce a building map more accurately than Microsoft's deep learning-based method as long as a decent quality of ALS data is available.

*2) Tiling comparison:* From Denver's 784 tiles, we extracted 5 tiles that ranked within the top 10% showing notable map differences. Figure 7 presents RGB images alongside their 3D building map, LiDAR building map (produced by our workflow), Microsoft Building Footprints, and ground-truth. The RGB imagery is sourced from National Agriculture Imagery Program (NAIP)'s orthoimagery taken in 2015. Given the 2015 vintage of the RGB imagery, it is anticipated that buildings evident in both the RGB image and LiDAR building map would also feature in Microsoft Building Footprints, taking into account their respective vintages (see Table I). Rankings presented above the RGB images offers insight into the degree of difference of the tile among across all 784 tiles of the Denver dataset. IoU values, which compare both the LiDAR Building Map and Microsoft Building Footprints to the ground truth, are also presented.

TABLE II: Conventional quantitative results of the Denver dataset

| | IoU | Precision | Recall | F1-score |
|---|---|---|---|---|
| Our workflow | **81.8** | **91.2** | **88.8** | **90.0** |
| Microsoft's | 77.3 | 90.4 | 84.2 | 87.2 |

Distinctive performance disparities were evident with large and uniquely-shaped buildings. Microsoft's footprints had difficulties with vast structures such as shopping malls and warehouses, as well as with buildings having unique contours, like sports complexes. These challenges could arise from a lack of representative training data for these less common building types, or the constrained input dimensions typically used by deep learning models for semantic segmentation. For instance, there's a possibility that a model's input only captures the central portion of a large building's roof, depriving the model of crucial information for effective decision-making.

Our approach consistently outperformed in terms of IoU values, even in residential areas. It capably identified auxiliary units such as garden sheds or detached garages, elements often overlooked by Microsoft, as exemplified in the fifth row of Figure 7. This might be attributable to the resolution of optical images, canopy obstructions, or a deficiency in training data for these units. Comprehensive discussions on auxiliary units are provided in Section III-D.

### *C. Results of the NYC dataset*

*1) Conventional quantitative results:* As tabulated in Table III, the LiDAR building map displayed superior accuracy compared to Microsoft Building Footprints across all metrics except for precision. The lower precision can largely be attributed to the building boundary refinement undertaken with dilation kernel K3. A more comprehensive discussion regarding this is presented in Section IV-B. These findings emphasize that building maps produced by our workflow are consistently more accurate than Microsoft Building Footprints, even when factoring in the broader temporal discrepancy with the ground-truth.

*2) Tiling comparison:* From the 1428 tiles of NYC, we chose five tiles that show pronounced discrepancies in different building maps. Figure 8 displays RGB images alongside their 3D building map, LiDAR building map, Microsoft Building

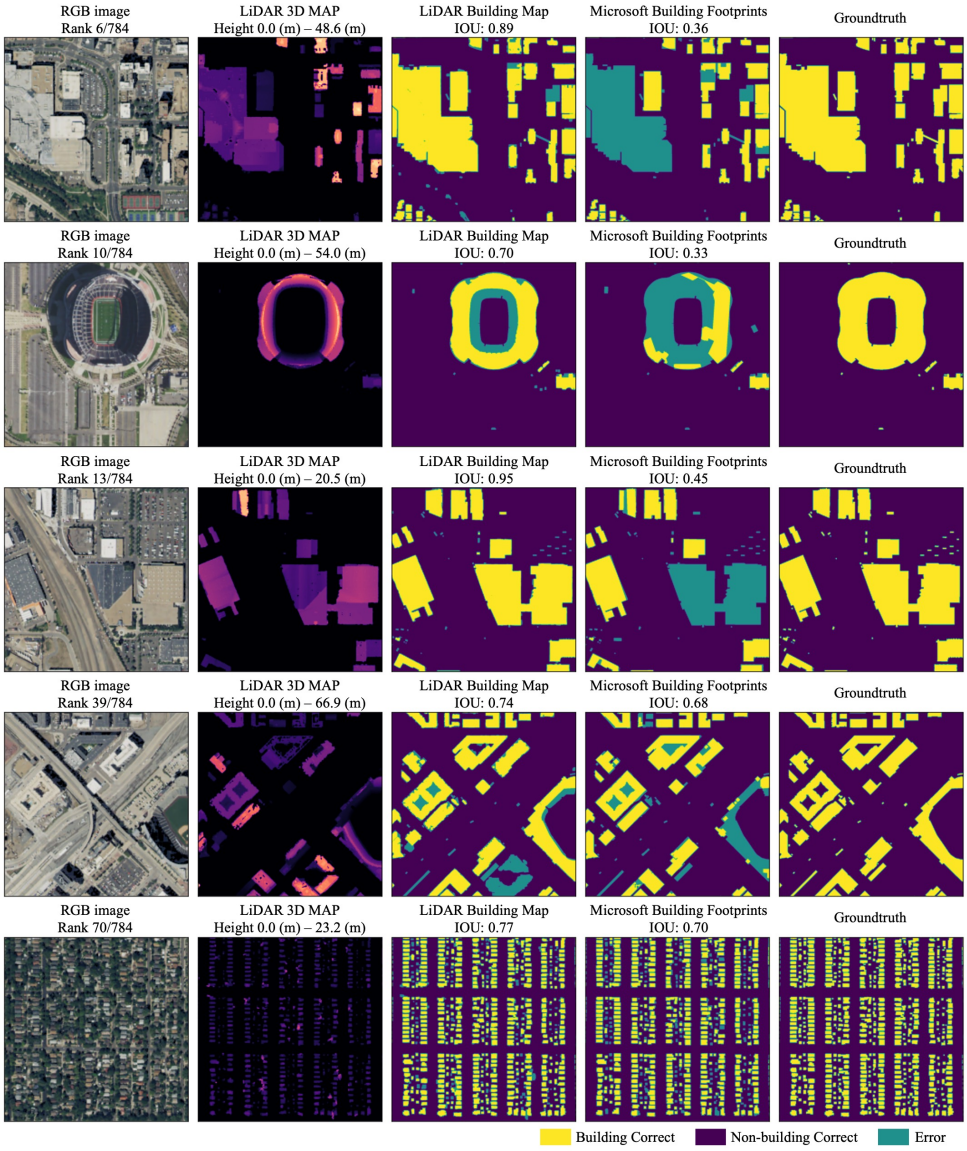



Fig. 7: Comparative visual assessment of five selected tiles from the Denver dataset. Displayed are the RGB images, our 3D building map outputs, LiDAR building maps, Microsoft Building Footprints, and the actual ground-truth. Our workflow precisely delineates large and uniquely shaped buildings, while the deep learning-based output (i.e. Microsoft Building Footprints) sometimes introduced unexplainable artifacts in such structures.

TABLE III: Conventional quantitative results of the NYC dataset

| | IoU | Precision | Recall | F1-score |
|---|---|---|---|---|
| Our workflow | **75.9** | 80.3 | **93.2** | **86.3** |
| Microsoft's | 72.8 | **84.6** | 83.9 | 84.3 |

Footprints, and ground-truth. The RGB imagery is sourced from NAIP's orthoimagery taken in 2015. Given this vintage, buildings present in the RGB image and ground-truth should also appear in the 2019 Microsoft Building Footprints. Similar to the Denver scenario, the RGB image ranking highlights key differences between the LiDAR building map and Microsoft footprints. IoU values, which compare the LiDAR and Microsoft building maps with the ground truth, are also displayed.

Unlike the Denver dataset, most top-ranked errors in New York arose from the sea. Microsoft's map frequently exhibited artifacts in water bodies, an issue also observed by [22]. On the other hand, our LiDAR map rarely showed errors over water bodies, due to a water body masking process. However, it did occasionally identify large ships as buildings. Errors near coastlines sometimes emerged due to DTM inaccuracies; these are elaborated upon in Section IV-C.

On land, mirroring Denver's findings, many Microsoft Building Footprint errors related to sizable or uniquely contoured structures. For instance, airports were problematic, perhaps due to scarce training data samples. The LiDAR map occasionally misinterpreted airplanes as buildings due to their similar physical properties to buildings in LiDAR signatures.

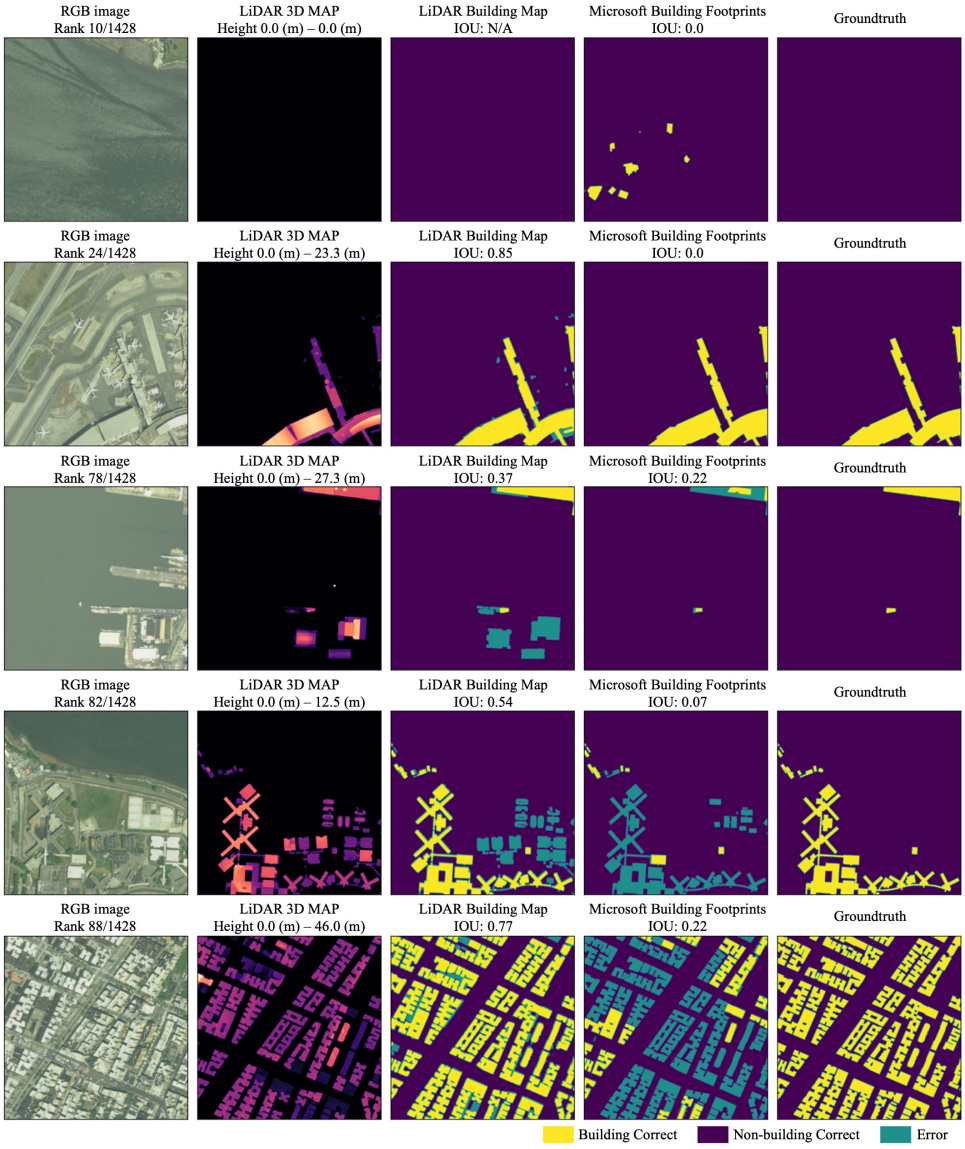



Fig. 8: Comparative visual assessment of five selected tiles from the NYC dataset. Displayed are the RGB images, our 3D building map outputs, LiDAR building maps, Microsoft Building Footprints, and the actual ground-truth. Our workflow sometimes misidentifies large, non-building objects, such as airplanes or ships, as buildings. Conversely, the deep learning-based output sporadically produced artifacts over water bodies and other errors that are difficult to clearly explain.

Other misclassifications included trailers and cargos.

Temporal dataset discrepancies also caused errors. An instance from Figure 8's fourth row shows commission errors on the LiDAR building map, attributed to the temporal gap. We found that some buildings in the LiDAR building map were demolished after the LiDAR scanning. Microsoft's method, however, failed to detect those buildings, which must have existed at the time of the mapping. This failure might be either due to the unique shape of the buildings or the non-availability of cloud-free optical imagery. The fifth row of Figure 8 reveals a marked disparity between the building maps in the downtown area. The discrepancy in Microsoft's mapping doesn't seem to arise from either the temporal differences or the architecture of the buildings.

In contrast to these unexpected errors in Microsoft's image-based approach, the errors in the LiDAR building maps are generally more explainable. This underscores the merit of our method, which centers on the physical representations of buildings rather than intricate modeling that may not always generalize well. Also, as LiDAR employs an active sensor, its accuracy isn't as susceptible to atmospheric interferences like clouds. This positions LiDAR-based building mapping at a distinct advantage over its optical image-based counterparts. Because the errors in our method are more discernible and foreseeable than in image-based techniques, there's a substantial reduction in the uncertainties tied to building maps. This clarity is crucial as it renders the errors more manageable, bolstering the reliability of subsequent analyses that leverage these maps. One limitation of our approach is that it can generate "fat" buildings, as depicted in the fifth row of Figure

8. This issue stems from the building boundary refinement process using dilation (K3), which also primarily accounts for the lower precision of the LiDAR building map in comparison to Microsoft Building Footprints, as indicated in Table III. A more discussion on K3 can be found in Section IV-B.

### D. Error Analysis by Building Size

*1) Omission error:* We observed both maps from our workflow and Microsoft's method often miss small buildings. To evaluate the performance, focusing on small-to-medium-sized buildings (< 800 $m^2$), we counted the number of correctly detected buildings for different building areas. Here, the building area refers to the area of each building's footprint. We defined a "correctly detected building" as an instance that exists in the ground-truth and its overlapped portion with generated building instances is more than 50%. Figure 9 and Figure 10 show the number of correctly detected buildings according to the building area for the Denver dataset and NYC dataset, respectively. The "Number of Buildings" in those graphs refers to the number of buildings in ground-truth for each building area category. Only buildings smaller than 800 $m^2$ were illustrated as the performance significantly varied in relatively small buildings (< 100 $m^2$) and the trend was generally maintained for larger buildings.

In the Denver dataset, both maps often omit buildings under 50 $m^2$, like sheds or detached garages. Conversely, the NYC dataset shows a higher detection rate because its authoritative map doesn't classify storage sheds as buildings, unlike Denver's. Since our workflow uses an erosion with K1 = 7, it removes some buildings under around 10 $m^2$, mostly storage sheds. While reducing K1 can detect smaller buildings, it risks misclassifying non-building objects. This K1 trade-off is discussed in Section IV-B.

The significant difference between the datasets highlights the potential inconsistency in building definitions across various authoritative maps from different states and countries. In light of this, our physical-property-driven, deterministic workflow presents an avenue for producing more standardized building maps.

Figure 11 provides the building counts in ground-truths of the two datasets. Each category is to represent the different types of buildings. The categorization consists of four classes with different ranges of building area: 0-50 $m^2$ ("accessorial class"), 50-500 $m^2$ ("residential class"), 500-10,000 $m^2$ ("commercial class"), and 10,000 $m^2$– ("mega-size class").

The first category represents accessorial buildings. It can include storage sheds, detached garages, trailers, portable cabins, and so forth. Some building types in this category are often excluded in some classification systems. Indeed, the authoritative map of Denver categorized the storage shed as a building while that of NYC excluded it. The second category, the residential class, represents most of the typical residential buildings. This category accounts for the majority of buildings. The commercial class, represents large commercial buildings. This category may include office buildings, retails, hospitals, warehouses, and industrial buildings. The last category represents mega-size buildings such as large shopping malls, factories, train stations, airports, and sports complexes. Based on this categorization, we compared the detection performances of our workflow and Microsoft's method. Although the type of building cannot be classified by only their areas, this categorization can provide a sense of the difference in performance according to the building types.

Table IV and Table V show the detection (true positive) rate of each category respectively for each dataset. The detection rate refers to the ratio of the number of correctly detected buildings to the number of buildings in ground-truth.

TABLE IV: Detection (true positive) rates according to the building category in the Denver dataset

| | Detection rates (%) | | | |
|---|---|---|---|---|
| | Accessorial- | Residential- | Commercial- | Mega-size |
| Our workflow | **25.8** | **96.1** | **98.4** | **97.6** |
| Microsoft's | 20.6 | 93.3 | 97.1 | 92.4 |

In the Denver dataset, both methods struggled with the accessorial class. But in the NYC dataset, our method had a 67.1% detection rate compared to Microsoft's 22.7% in this class. This variance is attributed to differing classification

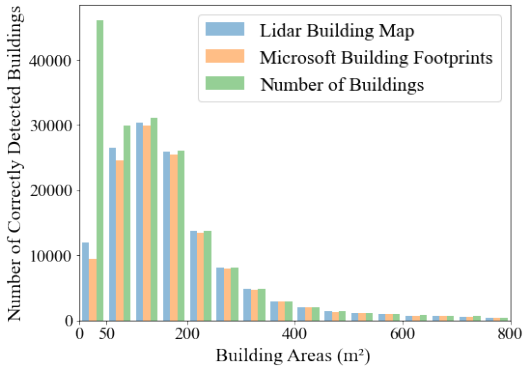



Fig. 9: The numbers of correctly detected buildings and the number of buildings in the ground-truth according to the building area (0-800 $m^2$) in the Denver dataset

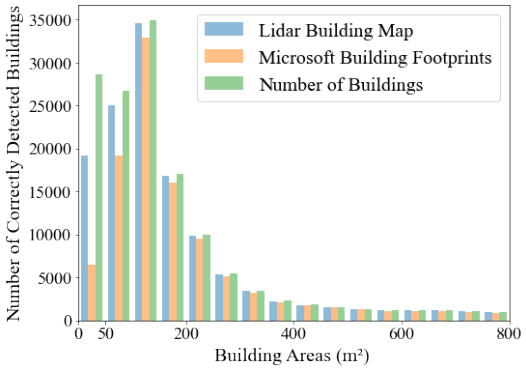



Fig. 10: The numbers of correctly detected buildings and the number of buildings in the ground-truth according to the building area (0-800 $m^2$) in the NYC dataset

| | Accessorial Building (0-50 $m^2$) | Residential Building (50-500 $m^2$) | Commercial Building (500-10,000 $m^2$) | Mega-size Building (10,000 $m^2$-) |
|---|---|---|---|---|
| Denver | 46071 | 120271 | 10335 | 170 |
| New York City | 28647 | 103259 | 23621 | 435 |

(Icon copyright: Flaticon)

Fig. 11: The number of buildings in ground-truths according to the building area

TABLE V: Detection (true positive) rates according to the building area in the NYC dataset

| | Detection rates (%) | | | |
|---|---|---|---|---|
| | Accessorial- | Residential- | Commercial- | Mega-size |
| Our workflow | **67.1** | **97.4** | **97.7** | **96.1** |
| Microsoft's | 22.7 | 88.4 | 94.2 | 89.0 |

systems: Denver labeled storage sheds (a significant part of the accessorial class) as buildings, while NYC did not. Since both methods found it challenging to detect these sheds, they both yielded low accessorial building detection rates in Denver. However, in NYC, where sheds weren't classified as buildings, the performance disparity between the two methods was stark.

While the exact building type statistics were unclear, there was a notable distribution difference between the two datasets. In Denver, 52.0% of accessorial class buildings were under 20 $m^2$, compared to just 17.9% in NYC. Given that most storage sheds are below 20 $m^2$, it's evident that the classification system differences contributed to the significant performance gap in the accessorial class. Except for the accessorial category, our workflow produced a detection rate of over 95.0% in all cases.

*2) Commission error:* Similar to the omission error, we tabulated commission (false positive error) rates of the two methods per building class. Table VI and Table VII show the commission rates respectively for the two datasets. The commission rate refers to the number of incorrectly detected buildings out of the total number of buildings in ground-truth. The "incorrectly detected building" is defined as an instance that exists in generated building map but its overlapped portion with the ground-truth is less than 50%.

Overall, our workflow had higher commission errors compared to Microsoft's. The commission rate was especially elevated for the accessorial class in the NYC dataset. This correlates with the LiDAR building map's high detection rate in this class. In essence, our method detected more accessorial buildings than Microsoft's, leading to more positives, whether true or false.

Commission error accounts for a relatively small portion of the total error in both methods. The time discrepancy to ground-truth accounts for some portions of the reason. Except for this, most of the commission errors from our workflow were overhanging trees or DTM-related artifacts. Detailed analyses of these errors are described in Section IV-C.

TABLE VI: Commission (false positive) rates according to the building area in the Denver dataset

| | Commission rates (%) | | | |
|---|---|---|---|---|
| | Accessorial- | Residential- | Commercial- | Mega-size |
| Our workflow | 4.3 | 2.6 | 2.8 | 1.2 |
| Microsoft's | **2.4** | **0.6** | **0.8** | **0.6** |

TABLE VII: Commission (false positive) rates according to the building area in the NYC dataset

| | Commission rates (%) | | | |
|---|---|---|---|---|
| | Accessorial- | Residential- | Commercial- | Mega-size |
| Our workflow | 19.1 | 4.6 | 2.5 | 6.0 |
| Microsoft's | **6.5** | **1.5** | **1.3** | **1.8** |

### *E. Compared to Other Deep Learning Methods*

In earlier evaluations, our proposed method demonstrated superior accuracy in extracting building footprints when compared to Microsoft Building Footprints. It's important to note the differences in data sources: our approach utilizes airborne LiDAR data, whereas Microsoft's employs satellite imagery. To address potential concerns regarding this difference in data sources, this section delves into a comprehensive comparison, covering methods from those that use LiDAR data to those employing RGB-NIR imagery.

As a robust benchmark, we employed U-Net [69], which has proven superior to traditional, non-deep learning techniques in building extraction tasks [12], [40], [41]. Drawing from the literature [40], [70]–[73], we adopted the following input data configurations:

1) Only nDSM
2) nDSM + LiDAR Intensity
3) nDSM + RGB-NIR
4) nDSM + LiDAR Intensity + RGB-NIR

While the incorporation of RGB-NIR imagery arguably offers an advantage—since it's an additional data source our workflow doesn't harness—we included this to establish a highly competitive baseline.

We subjected the deep learning models to two evaluation scenarios:

1) Same City Scenario: Here, both training and test datasets come from the same city. For this setup, both Denver and NYC datasets were split into Western (W) and Eastern (E) regions, as delineated in Figure 12. The evaluation involved training the model on the West and testing on the East, later reversing these roles. This was executed for both cities.
2) Different City Scenario: In this scenario, the model was trained using the entire Denver dataset (W+E) and subsequently tested on the full NYC dataset (W+E), and the reverse. Although this allows the model to leverage a larger set of training samples, it poses increased challenges compared to the same city scenario, owing to a more distinct shift in data distribution.

While leveraging randomly distributed training samples from the same city will enhance deep learning results, our

TABLE VIII: Comparative performance summary of the U-Net model, Microsoft Building Footprints, and our workflow. The table illustrates U-Net's mean IoU values across different input configurations (nDSM, LiDAR Intensity, and RGB-NIR) and evaluation scenarios (Same City and Different City)

| **Test Area** | **Same City Scenario** | | | | **Different City Scenario** | |
|---|---|---|---|---|---|---|
| | **Denver** | | **NYC** | | **Denver** | **NYC** |
| | (W) | (E) | (W) | (E) | (W + E) | (W + E) |
| Train Area | Denver (E) | Denver (W) | NYC (E) | NYC (W) | NYC (W+E) | Denver (W+E) |
| **U-Net** (inputs are listed below) | | | | | | |
| nDSM | 77.1 ± 0.9 | 77.9 ± 0.8 | 57.5 ± 5.5 | 64.5 ± 2.3 | 48.5 ± 9.6 | 63.3 ± 1.8 |
| nDSM+Intensity | 75.8 ± 2.1 | 77.8 ± 0.6 | 59.4 ± 4.3 | 63.9 ± 1.9 | 47.7 ± 5.3 | 60.1 ± 3.5 |
| nDSM+RGB-NIR | 79.4 ± 0.6 | 82.3 ± 0.4 | 73.9 ± 2.8 | 68.1 ± 4.1 | 57.2 ± 3.7 | 63.2 ± 1.6 |
| nDSM+Intensity+RGB-NIR | **80.5 ± 0.3** | **83.1 ± 0.4** | 73.4 ± 0.6 | 71.5 ± 0.3 | 57.7 ± 4.7 | 59.7 ± 4.6 |
| **Microsoft's** | 75.5 | 79.4 | 73.6 | 71.6 | 77.3 | 72.8 |
| **Our workflow** | **80.9** | **82.9** | **77.6** | **73.7** | **81.8** | **75.9** |

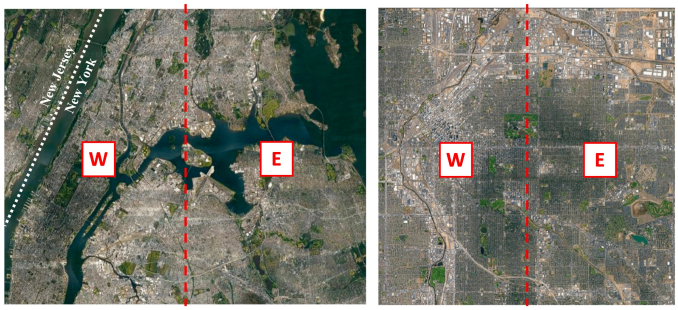



Fig. 12: Graphical representation of Denver and NYC highlighting the division into Western (W) and Eastern (E) sectors for the purpose of evaluation in Table VIII and Figure 13

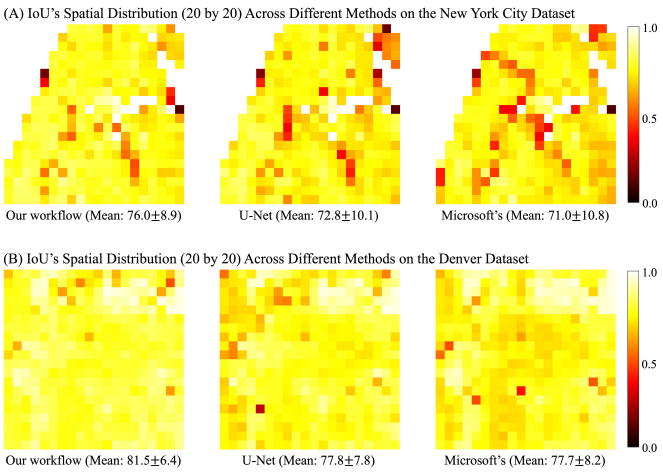



Fig. 13: Spatial distribution of IoUs for three distinct methods: our workflow, U-Net, and Microsoft's method, using datasets from New York City and Denver. Each dataset is represented in a 20 by 20 grid. The heatmap intensity indicates the IoU value, with associated mean and standard deviation values presented below. Differences in uniformity across the methods highlight variations in model performance.

scenarios were designed to reflect the expected real-world performance of deep models—a commonly embraced approach since training samples aren't always feasibly collected from all regions.

For our U-Net model, we used a direct approach: all inputs were stacked and processed via the classic U-Net structure. This decision was rooted in our intent to set a basic yet clear benchmark for comparison. While more sophisticated techniques like custom deep learning modules or multi-stream fusion methods might offer enhanced results [71]–[73], our main goal was to establish a transparent baseline to gauge the effectiveness of our proposed method. For optimization, we used the Adam optimizer, utilized a batch size of 16, and tested various learning rates ($1 \times 10^{-3}$, $1 \times 10^{-4}$, $1 \times 10^{-5}$) across five iterations for each scenario. We applied early stopping with a patience of 5.

The mean IoU from the optimal learning rate is presented in Table VIII, contrasting it with the IoUs from Microsoft's Building Footprints and our method. Results reaffirmed the robustness of our method. Although alternative deep learning models with expansive training data might potentially surpass the current U-Net benchmark, the significance of our workflow's competitive edge remains undiminished. Notably, contrasting our results with the Different City Scenario, it becomes evident that while U-Net suffers from data distribution shifts, our approach remains robust, offering noteworthy advantages.

Figure 13 depicts the spatial distribution of IoUs for three different methods: our workflow, U-Net, and Microsoft's method. Each dataset, both for New York City and Denver, was divided into a 20 by 20 grid. Building maps generated from these methods were compared to the ground-truth, resulting in the IoUs for each grid being visualized as a heatmap. The U-Nets utilized nDSM, LiDAR intensity, and RGB-NIR as their inputs. We employed different training and testing parts within the same city (trained in the west and inferenced in the east, and vice versa) for building map generation. Below the heatmap, the mean and standard deviation of the gridded IoUs are displayed. The mean IoU presented signifies the mean IoU across the 20 by 20 grids.

Notably, the spatial distribution of IoUs from our workflow is more consistent compared to other methods, as also evidenced by the IoU's standard deviation. This indicates

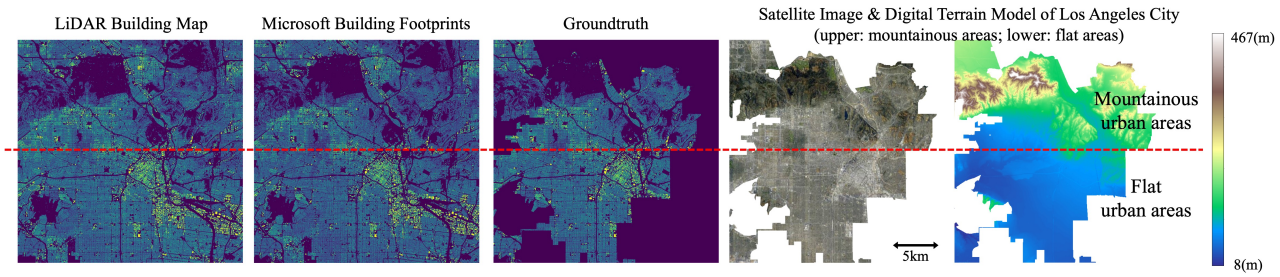



Fig. 14: Assessment of our workflow in mountainous terrains: The figure showcases the LiDAR building map derived from our method, Microsoft Building Footprints, Groundtruth, and the RGB satellite imagery along with the digital terrain model of Los Angeles. The experimental area's northern region predominantly features mountainous landscapes, whereas the southern region is characterized by flatter urban terrains.

that both U-Net and Microsoft's performances exhibit greater variability across regions. The variation in U-Net's performance can be attributed to the data distribution shift, given our experimental setup of west and east division. Intriguingly, even with Microsoft's method trained on millions of building labels nationwide, covering a vast range of building types, its performance consistency lags behind ours. This highlights the robustness and scalability of our workflow that capitalizes on the 3D physical representations of objects as they are captured in airborne LiDAR measurements.

### *F. Performance in Mountainous Areas*

The proposed method relies on the 3D coordinate data derived from ALS. Given this, assessing its performance in regions with intricate topographies, such as mountainous areas, becomes imperative. To this end, we selected Los Angeles as our test city. Los Angeles presents a diverse landscape for evaluation: the northern section is predominantly mountainous, while the southern region features a more urban and flat terrain, as depicted in Figure 14. We generated a LiDAR building map for Los Angeles and compared it with Microsoft Building Footprints, using the ground-truth sourced from Los Angeles GeoHub (https://geohub.lacity.org/). The ground-truth's last update was in 2017, our LiDAR data is from 2016-2017, and Microsoft's data is from 2018-2019. Yet, this data provides crucial insights into the performance variances between mountainous and flat urban regions at large-scale.

Table IX details the results. Overall, our approach surpasses Microsoft Building Footprints in accuracy across both terrains, highlighting our workflow's robustness and adaptability. Nevertheless, we acknowledge that in extremely rugged terrains, DTM* might, in theory, yield errors, thereby potentially affecting the precision of building extraction. While these instances are infrequent in typical urban settings, additional attention should be required when a project's target area includes heavily rugged mountains.

### *G. Summary of results*

Extensive quantitative and qualitative assessments confirm that our workflow consistently outperforms Microsoft Building Footprints across most metrics. Notably, our approach shines in metrics like IoU, recall, and F1-score, and demonstrates superior detection rates across all building categories. Moreover, its performance remained robust and superior, even when benchmarked against highly competitive deep learning methods utilizing LiDAR data or in challenging terrains like mountainous areas.

TABLE IX: Comparison of building extraction performance between the proposed workflow and Microsoft Building Footprints in the Los Angeles dataset, categorized by mountainous (Upper) and flat urban (Lower) terrains

| | IoU | Precision | Recall | F1-score |
|---|---|---|---|---|
| **Upper** | | | | |
| Our workflow | **75.1** | 86.0 | **85.5** | **85.8** |
| Microsoft's | 73.5 | **89.2** | 80.7 | 84.8 |
| **Lower** | | | | |
| Our workflow | **76.5** | 86.8 | **86.6** | **86.7** |
| Microsoft's | 75.9 | **89.9** | 83.0 | 86.3 |

Importantly, our workflow operates unsupervised, eliminating the need for training procedures. Yet, our optimized method outshines other competitive approaches without additional parameter tuning. Our workflow's outputs are largely predictable and explainable. This transparency is particularly invaluable in extensive mapping projects where exhaustive validation of every output isn't feasible, ensuring reduced uncertainties.

## IV. Discussion

### *A. Computational Complexity Analysis*

The following analysis evaluates the computational complexity of our workflow, focusing on key operations that notably impact processing time:

1) Creation of DSM: Reading and distributing each LiDAR point to an array results in a complexity of $O(n)$, where $n$ represents the number of points. The nearest neighbor interpolation, which is dependent on the point count, similarly exhibits an $O(n)$ complexity.
2) Creation of DTM: Both the Sobel operator and the connected component analysis are determined by the number of pixels in the image, resulting in an $O(n)$

complexity. The subsequent interpolation also has an $O(n)$ complexity.

3) Creation of Building Candidates Map: This step consists of evaluating height conditions and matrix multiplication, resulting in an overall $O(n)$ complexity.
4) Creation of Building Map:
   a) Water body masking: This involves a sliding window operation for water body layer generation, resulting in an $O(n)$ complexity.
   b) Morphological filtering: The morphological opening operation has an $O(n)$ complexity.
   c) Planarity filtering: This also involves a sliding window operation, approximated as $O(n)$.
   d) Boundary refining: Analogous to morphological dilation, this step has a complexity of $O(n)$.

For typical large-area building mapping tasks, it's advisable to segment the entire project area into smaller tiles, such as 1 km by 1 km tiles. In this context, with 4 points/m$^2$ as an example, there would be 4 million points for each 1 km by 1 km tile. This equates to an array of 2000 by 2000 pixels, given a 0.5-m resolution. Under these conditions, the creation of the DSM file often consumes the most time due to the need to read all the LiDAR data points. Following that, interpolation can also be time-consuming as its computation increases non-linearly for larger rasters. While real-world computational times can be affected by various factors, employing parallel computation can greatly accelerate processing.

### B. Suggestion for parameter tuning

Fine-tuning parameters can improve performance for specific scenarios. Crucial parameters to consider are K1, which influences preferences for accessory classes; DT, essential for densely forested regions; and K3, which impacts building boundary metrics. This section delves into these parameters and provides tuning recommendations.

*1) K1 parameter tuning for handling accessorial buildings:* Increasing K1 typically reduces commission error but raises omission error due to the erosion of smaller structures with larger K1 values. Specifically, buildings smaller than K1$*$GSD will be eroded. With a default K1 value of 7 for 0.5-m GSD, typical residential buildings are retained, but some accessorial buildings might be removed. Erosion primarily removes trees which are represented noise-like patches. However, some dense trees or shrubs might remain, underscoring the need for planarity-based filtering with DT.

Figure 15 displays three sample areas from Denver to highlight the effects of different K1 values. Generally, K1 $= 7$ offers optimal performance, but there are trade-offs. The first row in Figure 15 shows commission error with a smaller K1 $= 5$; containers are mislabeled as buildings. The second row illustrates omission errors when K1 $= 9$, resulting in the removal of detached garages. The third row of Figure 15 shows the trade-off between the omission of the storage shed and commission errors of large, dense trees.

We also quantitatively evaluated the impact of K1 by mapping the Denver dataset using different K1 values. Table X and Table XI show the detection and commission rates,

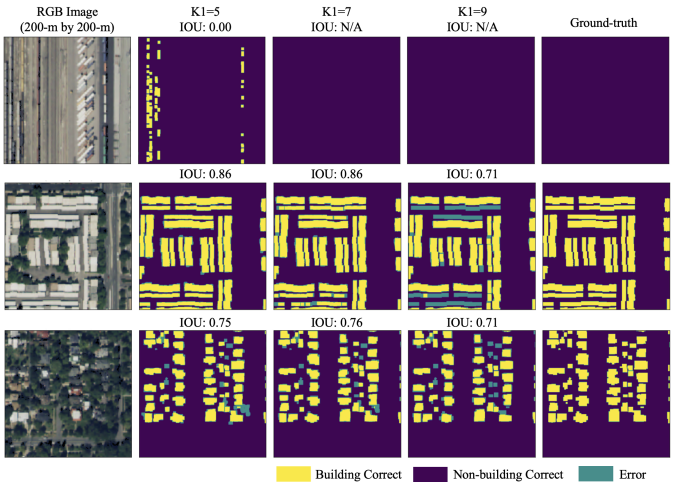



Fig. 15: Generated building maps with different K1 values

respectively. The accuracy significantly varied in accessorial class (0-50 m$^2$) for different K1 values. A smaller K1 increased the detection rate but also raised the commission rate. In simpler terms, a smaller K1 is more inclined to map small, accessory buildings, but it is also more prone to misclassifying non-building small objects as buildings. Conversely, a larger K1 can reduce commission errors but may erase some small buildings.

TABLE X: The impact of parameter K1 on the workflow's detection (true positive) rates in the Denver dataset

| | Detection rates (%) | | | |
|---|---|---|---|---|
| | Accessorial- | Residential- | Commercial- | Mega-size |
| K1=5 | **40.5** | **97.4** | **98.5** | **97.6** |
| K1=7(default) | 25.8 | 96.1 | 98.4 | **97.6** |
| K1=9 | 13.2 | 91.5 | 98.1 | **97.6** |

TABLE XI: The impact of parameter K1 on the workflow's commission (false positive) rates in the Denver dataset

| | Commission rates (%) | | | |
|---|---|---|---|---|
| | Accessorial- | Residential- | Commercial- | Mega-size |
| K1=5 | 12.9 | 3.5 | 3.2 | **1.2** |
| K1=7(default) | 4.3 | 2.6 | 2.8 | **1.2** |
| K1=9 | **1.9** | **2.0** | **2.6** | **1.2** |

*2) Dense trees (DT) parameter tuning for forested areas:* The DT parameter serves as the threshold for the planarity-based filtering algorithm, primarily aimed at eliminating dense trees. As DT increases, it's more likely both dense trees and buildings beneath overhanging trees will be removed. We examined this phenomenon in Callaway, Florida, an 11 km$^2$ region characterized by residential zones and dense tropical evergreen trees. The LiDAR surveying was conducted during the full leaf-on period in April-May 2017, providing a point density of about 7-points/m$^2$. Figure 16 provides the satellite image and the generated building map from our workflow.

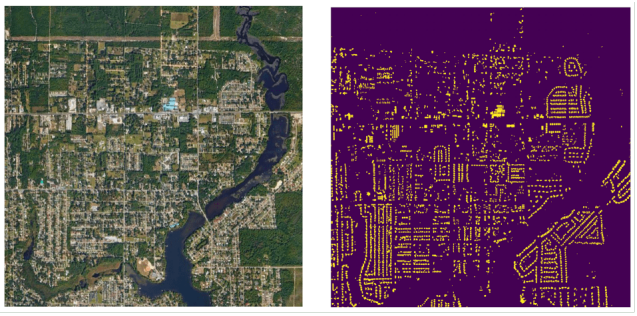


Fig. 16: Satellite image and LiDAR building map of the Callaway dataset (3.3-km by 3.3-km)

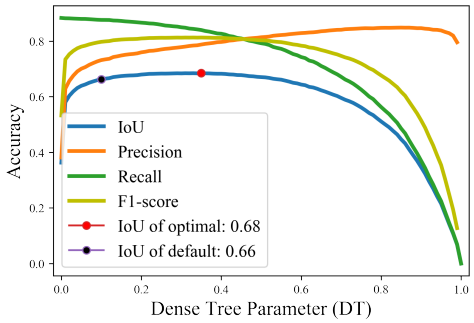



Fig. 17: IoU, precision, recall, and F1-score according to the parameter DT in the Callaway dataset

To assess the influence of DT values, we varied DT and calculated metrics like IoU, precision, recall, and F1-score, as seen in Figure 17. We used Microsoft Building Footprints as the ground-truth due to the lack of a more reliable source. An optimal DT value of 0.35 achieved an IoU of 0.68, a modest improvement from the default's 0.66 IoU. This suggests that while tweaking DT can slightly enhance accuracy, the default setting remains broadly effective. Also, the plateau in the IoU graph between DT values of 0.05-0.5 indicates that the default DT value typically yields near-optimal results within this range.

Although DT $= 0.35$ produced the highest IoU value, it does not necessarily guarantee the best building map. Figure 18 depicts the study area in Callaway, Florida, showcasing maps generated at various DT levels. At DT $= 0$, all building candidates are shown prior to planarity-based filtering. At the default DT $= 0.1$, most dense trees are removed, but some non-building objects remain. At DT $= 0.2$, only buildings are extracted, but DT $= 0.3$ erroneously removes two buildings under overhanging trees, as validated via Google Street View. These suggest the optimal DT value varies by study area and is particularly sensitive in tropical regions, where dense foliage can affect results. This limitation is more pronounced with low point density, as further discussed in Section IV-C.

*3) K3 parameter tuning for refining building boundaries:* The K3 parameter refines building boundaries by addressing underestimated building areas. This underestimation is attributed to the radial whiskbroom pattern of ALS. As illustrated in Figure 19, the blue grid captures multiple points

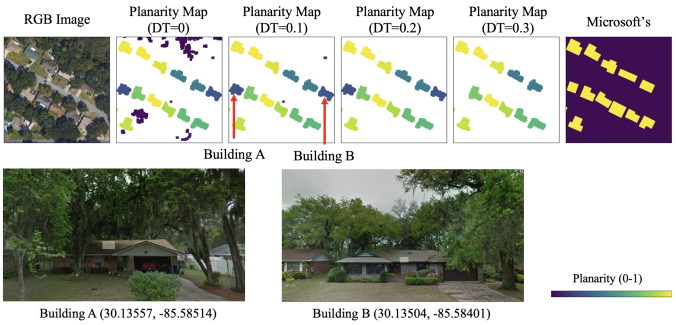



Fig. 18: Generated building maps (planarity maps) according to the parameter DT and examples of buildings under overhanging trees in Google Street View (Image copyright: Google Inc.). The numbers in parentheses indicate the latitude and longitude of the building in decimal degrees, respectively.

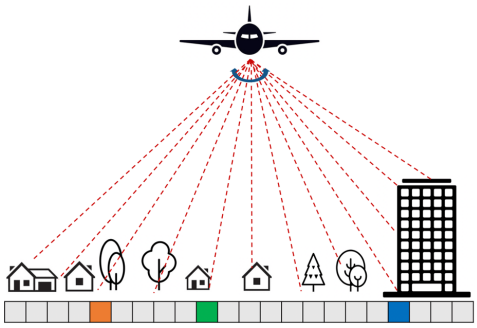


Fig. 19: Graphical representation of airborne laser scanning

reflecting mixed elevations from the bottom, side, and top of the building. Using the lowest elevation values to produce the DSM leads to an underestimation of the total building area. Such underestimation can occur even without acquiring multiple points. Both the green and orange grids, for instance, will be categorized as non-building pixels, as the laser captures the building's base, leading to boundary underestimation. Factors like point density, scan angle, and laser beam position influence this. Ideally, K3 would be tailored for each pixel based on these factors, but data constraints make this computationally formidable. So, we use a universal K3 value although it isn't optimal at the individual object level. Quantitative tests suggest a default K3 of 5 offers good accuracy in general, but optimal values vary by area, as seen in the Denver and NYC datasets (Table XII and Table XIII). Variations in optimal K3 values may also stem from inconsistencies in ground-truth.

TABLE XII: The impact of parameter K3 on the workflow's quantitative metrics in the Denver dataset

| | IoU | Precision | Recall | F1-score |
|---|---|---|---|---|
| K3=1 | 70.1 | **95.6** | 72.4 | 82.4 |
| K3=3 | 77.6 | 94.2 | 81.4 | 87.4 |
| K3=5(default) | **81.8** | 91.2 | 88.8 | **90.0** |
| K3=7 | 80.5 | 85.6 | **93.2** | 89.2 |

## C. Limitation

While most errors in our workflow can be mitigated through parameter tuning as discussed in Section IV-B,

TABLE XIII: The impact of parameter K3 on the workflow's quantitative metrics in the NYC dataset

| | IoU | Precision | Recall | F1-score |
|---|---|---|---|---|
| K3=1 | 77.0 | **89.4** | 84.7 | 87.0 |
| K3=3 | **77.9** | 85.3 | 90.0 | **87.6** |
| K3=5(default) | 75.9 | 80.3 | 93.2 | 86.3 |
| K3=7 | 72.2 | 75.1 | **94.8** | 83.9 |

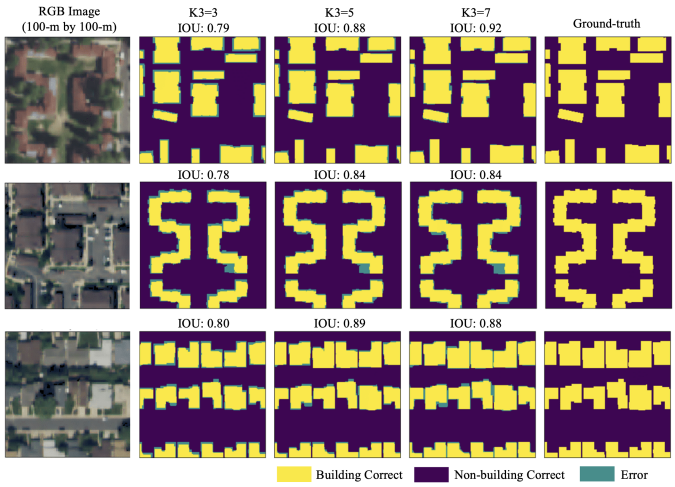



Fig. 20: Generated building maps with different K3 values

there are intrinsic limitations not directly related to the commission-omission error trade-off. This section devotes to discussing those limitations.

*1) Error near skyscrapers:* Though LiDAR isn't impacted by shadows, it's susceptible to occlusion due to its slant scan angle. This can lead to missed data points between closely spaced tall buildings, especially near skyscrapers. Figure 21, sample area from Manhattan, New York, exemplifies occlusions causing blurred boundaries or gaps in the building map.

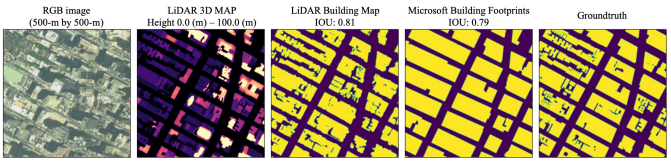



Fig. 21: Errors due to occlusion caused by skyscrapers

*2) Low point density scenarios:* In low point density situations, distinguishing between buildings and nearby overhanging trees becomes challenging. For instance, when a laser penetrates a tree, as highlighted by the orange pixel in Figure 19, it gets classified as a non-building entity, allowing differentiation between the tree and the building. But if trees are too dense to allow laser penetration, they may merge with adjacent buildings. This can result in either omission or commission errors, especially in low point density contexts.

We simulated scenarios with reduced point density by subsampling the original LiDAR data. Figure 22 presents various examples of dense trees during the leaf-on season. Remarkably, even at very low densities (<1-points/m$^2$),

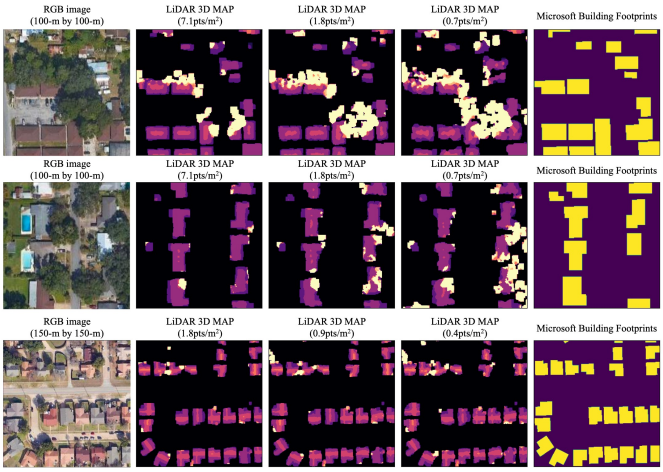



Fig. 22: Errors due to overhanging trees in low point density scenarios

buildings without overhanging trees were identified correctly. However, with diminishing point density, trees are increasingly misidentified as buildings. This suggests that as point density rises, the accuracy of our workflow will likely enhance.

*3) Commission errors due to the DTM:* The workflow's robustness is partially due to the DTM* generation algorithm's integration. DTM* classifies bridges and overpasses as terrain, reducing potential errors compared to other DTMs, as demonstrated in Figure 3. However, certain DTM-related issues persist. Errors can occur at the edge of the input LiDAR tile (the geographic boundaries of the input LiDAR data file). This is because DTM* assumes areas enclosed by a certain level of a steep slope as non-grounds, but the enclosed area near the edge of the LiDAR could be a disconnected terrain [59]. Similarly, terrains bordered by steep slopes might be mistakenly identified as buildings.

Figure 23 showcases such DTM-related inaccuracies. The red dashed line signifies the boundaries of the LiDAR tiles. Misclassifications are evident: in the first row, a bridge is wrongly labeled due to tile disconnections; in the second, lands near shorelines are erroneously recognized as buildings. The third row displays a suspension bridge; its middle section, surrounded by steep slopes in the DSM, is incorrectly extracted as a building. Most of these errors can be addressed by processing a more extensive input dataset and subsequently subsetting on its central portion.

*4) Deformation during the morphological filtering:* Our workflow includes the process that extracts buildings from among building candidates using morphological operations. One problem is that dilation after erosion may not recover the original shape of the object. If maintaining the precise boundary is paramount, it's recommended to explore alternative boundary-regularizing algorithms [33], [67], [74].

*5) Limitations in 3D building map:* Our workflow effectively generates 3D building maps, but limitations exist. Inaccuracies are especially pronounced near high-rise buildings (Figure 22) and near residential buildings obscured by

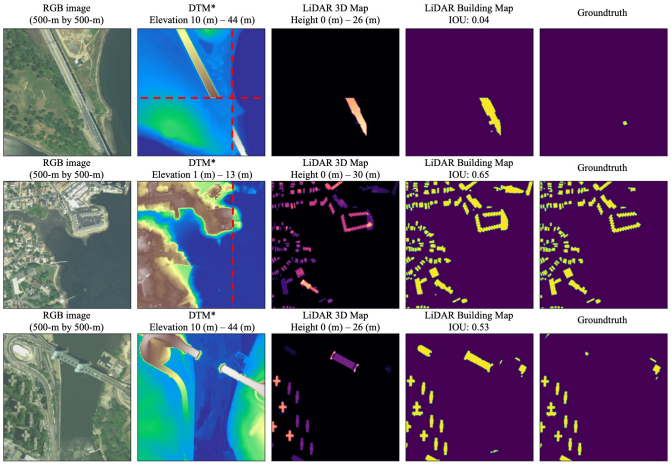



Fig. 23: Commission errors caused by artifacts in DTM (The red dashed line indicates the data boundary of the LiDAR tile)

overhanging trees (Figure 21). While smoothing filters help reduce noise, they might inadvertently remove minor substructures, such as chimneys. The map's quality hinges on the LiDAR data's accuracy, and refining building details requires more precise references. The primary focus of our workflow is on the mass production of rasterized 2D and 3D building maps. Users should be mindful of these inherent constraints.

## V. Conclusion

We present an end-to-end open-source workflow for 2D and 3D building mapping from raw airborne LiDAR data. To our knowledge, there has been no well-established open-source workflow for generating 2D and 3D building maps. Our system operates fully unsupervised, is computationally efficient, and produces accurate results without the need for intensive parameter tuning. Moreover, this study is the first to evaluate a LiDAR-based building map against deep learning-based and hand-digitized products on a large scale (> 550 $km^2$). Our work not only provides a practical solution for mass producing 2D and 3D building maps, but also furnishes valuable insights into the strengths and weaknesses of different methodologies, informing expectations for major large-area building mapping techniques.

In recent years, deep learning-based methods, bolstered by advancements in deep learning and the ready availability of image data, have dominated the literature. However, such methods have notable drawbacks. They typically require training procedures, are computationally expensive, and their errors can be unpredictable and difficult to explain. While deep learning-based methods can achieve high accuracy according to quantitative metrics, if users cannot anticipate or understand the errors or biases these models may introduce, subsequent studies based on these maps may lead to skewed outcomes.

Our workflow, on the other hand, is based on a largely universal representation of buildings in airborne LiDAR data that is a ground-standing object with a relatively smooth, laser-impermeable surface. As such, our workflow is highly scalable and its results are readily explainable. Also, since the operation is completely unsupervised and straightforward, all procedures are transparent, unlike the "black box" nature of deep learning.

Despite its exemplary performance, our workflow has limitations. While many can be addressed with higher quality (greater point density) LiDAR data, challenges remain, such as unclear boundaries between dense trees and small buildings, misclassification of non-building structures, deformations from morphological filtering, and limited detail in building models. Further research is required to tackle these issues. Adding more sophisticated rules to our workflow may fix some errors. However, it's vital to ensure that scalability and generalization aren't sacrificed.

The source code of our workflow is available at https://github.com/hunsoosong/airborne-lidar-building-mapping. Leveraging this workflow, we won first place in the GIS Cup, further underscoring its robustness and effectiveness [75]. We will continue to refine the workflow, and the generated 2D and 3D building maps will be released and kept updated. Our open-source workflow will enable the community to generate 2D and 3D building maps more readily and accurately, and we hope our workflow and findings contribute to supporting various studies with better building maps.

## Acknowledgement

All LiDAR data used in this paper are from the U.S. Geological Survey's 3D Elevation Program (3DEP). This research was funded by the National Geospatial-Intelligence Agency (NGA). Federal Award/Contract No. HM157522D0009/HM157523F0135 (Subaward No. SPC-1000012108/GR132562).

## References

[1] Z. Zhu, Y. Zhou, K. C. Seto, E. C. Stokes, C. Deng, S. T. Pickett, and H. Taubenböck, "Understanding an urbanizing planet: Strategic directions for remote sensing," *Remote Sensing of Environment*, vol. 228, pp. 164–182, 2019.

[2] S. Ghaffarian, N. Kerle, E. Pasolli, and J. Jokar Arsanjani, "Post-disaster building database updating using automated deep learning: An integration of pre-disaster openstreetmap and multi-temporal satellite data," *Remote sensing*, vol. 11, no. 20, p. 2427, 2019.

[3] J.-W. Hong, J. Hong, E. E. Kwon, and D. Yoon, "Temporal dynamics of urban heat island correlated with the socio-economic development over the past half-century in seoul, korea," *Environmental Pollution*, vol. 254, p. 112934, 2019.

[4] G. Herbert and X. Chen, "A comparison of usefulness of 2d and 3d representations of urban planning," *Cartography and Geographic Information Science*, vol. 42, no. 1, pp. 22–32, 2015.

[5] S.-s. Wu, X. Qiu, and L. Wang, "Population estimation methods in gis and remote sensing: A review," *GIScience & Remote Sensing*, vol. 42, no. 1, pp. 80–96, 2005.

[6] B. Herfort, S. Lautenbach, J. Porto de Albuquerque, J. Anderson, and A. Zipf, "The evolution of humanitarian mapping within the openstreetmap community," *Scientific reports*, vol. 11, no. 1, pp. 1–15, 2021.

[7] O. Harig, R. Hecht, D. Burghardt, and G. Meinel, "Automatic delineation of urban growth boundaries based on topographic data using germany as a case study," *ISPRS International Journal of Geo-Information*, vol. 10, no. 5, p. 353, 2021.

[8] R. Hecht, C. Kunze, and S. Hahmann, "Measuring completeness of building footprints in openstreetmap over space and time," *ISPRS International Journal of Geo-Information*, vol. 2, no. 4, pp. 1066–1091, 2013.

[9] J. E. Vargas-Munoz, S. Srivastava, D. Tuia, and A. X. Falcao, "Openstreetmap: Challenges and opportunities in machine learning and remote sensing," *IEEE Geoscience and Remote Sensing Magazine*, vol. 9, no. 1, pp. 184–199, 2020.

[10] H. Dorn, T. Törnros, and A. Zipf, "Quality evaluation of vgi using authoritative data—a comparison with land use data in southern germany," *ISPRS International Journal of Geo-Information*, vol. 4, no. 3, pp. 1657–1671, 2015.

[11] J. Jokar Arsanjani, P. Mooney, A. Zipf, and A. Schauss, "Quality assessment of the contributed land use information from openstreetmap versus authoritative datasets," in *OpenStreetMap in GIScience*. Springer, 2015, pp. 37–58.

[12] A. Van Etten, D. Lindenbaum, and T. M. Bacastow, "Spacenet: A remote sensing dataset and challenge series," *arXiv preprint arXiv:1807.01232*, 2018.

[13] Z. Huang, Q. Liu, H. Zhou, G. Gao, T. Xu, Q. Wen, and Y. Wang, "Building detection from panchromatic and multispectral images with dual-stream asymmetric fusion networks," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 16, pp. 3364–3377, 2023.

[14] L. Ma, Y. Liu, X. Zhang, Y. Ye, G. Yin, and B. A. Johnson, "Deep learning in remote sensing applications: A meta-analysis and review," *ISPRS journal of photogrammetry and remote sensing*, vol. 152, pp. 166–177, 2019.

[15] X. X. Zhu, D. Tuia, L. Mou, G.-S. Xia, L. Zhang, F. Xu, and F. Fraundorfer, "Deep learning in remote sensing: A comprehensive review and list of resources," *IEEE Geoscience and Remote Sensing Magazine*, vol. 5, no. 4, pp. 8–36, 2017.

[16] H. L. Yang, J. Yuan, D. Lunga, M. Laverdiere, A. Rose, and B. Bhaduri, "Building extraction at scale using convolutional neural network: Mapping of the united states," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 11, no. 8, pp. 2600–2614, 2018.

[17] P. Dias, Y. Tian, S. Newsam, A. Tsaris, J. Hinkle, and D. Lunga, "Model assumptions and data characteristics: Impacts on domain adaptation in building segmentation," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–18, 2022.

[18] N. Makkar, L. Yang, and S. Prasad, "Adversarial learning based discriminative domain adaptation for geospatial image analysis," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 15, pp. 150–162, 2021.

[19] H. Song, L. Yang, and J. Jung, "Self-filtered learning for semantic segmentation of buildings in remote sensing imagery with noisy labels," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 16, pp. 1113–1129, 2022.

[20] "Microsoft Building Footprints," https://www.microsoft.com/en-us/maps/building-footprints, accessed: 2022-09-27.

[21] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *International conference on machine learning*. PMLR, 2019, pp. 6105–6114.

[22] M. P. Heris, N. L. Foks, K. J. Bagstad, A. Troy, and Z. H. Ancona, "A rasterized building footprint dataset for the united states," *Scientific data*, vol. 7, no. 1, pp. 1–10, 2020.

[23] M. Li, E. Koks, H. Taubenböck, and J. van Vliet, "Continental-scale mapping and analysis of 3d building structure," *Remote Sensing of Environment*, vol. 245, p. 111859, 2020.

[24] Y. Park, J.-M. Guldmann, and D. Liu, "Impacts of tree and building shades on the urban heat island: Combining remote sensing, 3d digital city and spatial regression approaches," *Computers, Environment and Urban Systems*, vol. 88, p. 101655, 2021.

[25] F. Macchione, P. Costabile, C. Costanzo, and R. De Santis, "Moving to 3-d flood hazard maps for enhancing risk communication," *Environmental modelling & software*, vol. 111, pp. 510–522, 2019.

[26] S. Wang, Y. Tian, Y. Zhou, W. Liu, and C. Lin, "Fine-scale population estimation by 3d reconstruction of urban residential buildings," *Sensors*, vol. 16, no. 10, p. 1755, 2016.

[27] F. Biljecki, K. Arroyo Ohori, H. Ledoux, R. Peters, and J. Stoter, "Population estimation using a 3d city model: A multi-scale country-wide study in the netherlands," *PloS one*, vol. 11, no. 6, p. e0156808, 2016.

[28] H. Song, G. Cervini, and J. Jung, "Assessment of local climate zone products via simplified classification rule with 3d building maps," in *IGARSS 2023-2023 IEEE International Geoscience and Remote Sensing Symposium*. IEEE, 2023, pp. 3679–3682.

[29] N. Kim and Y. Yoon, "Regionalization for urban air mobility application with analyses of 3d urban space and geodemography in san francisco and new york," *Procedia Computer Science*, vol. 184, pp. 388–395, 2021.

[30] H. Lehner and L. Dorffner, "Digital geotwin vienna: towards a digital twin city as geodata hub," 2020.

[31] J.-Y. Han, Y.-C. Chen, and S.-Y. Li, "Utilising high-fidelity 3d building model for analysing the rooftop solar photovoltaic potential in urban areas," *Solar Energy*, vol. 235, pp. 187–199, 2022.

[32] Z. Zhou, J. Gong, and X. Hu, "Community-scale multi-level post-hurricane damage assessment of residential buildings using multi-temporal airborne lidar data," *Automation in Construction*, vol. 98, pp. 30–45, 2019.

[33] V. Kamra, P. Kudeshia, S. ArabiNaree, D. Chen, Y. Akiyama, and J. Peethambaran, "Lightweight reconstruction of urban buildings: Data structures, algorithms, and future directions," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 16, pp. 902–917, 2022.

[34] R. Qin, "A critical analysis of satellite stereo pairs for digital surface model generation and a matching quality prediction model," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 154, pp. 139–150, 2019.

[35] D. Huang, Y. Tang, and R. Qin, "An evaluation of planetscope images for 3d reconstruction and change detection–experimental validations with case studies," *GIScience & Remote Sensing*, vol. 59, no. 1, pp. 744–761, 2022.

[36] E. M. Domínguez, C. Magnard, E. Meier, D. Small, M. E. Schaepman, and D. Henke, "A back-projection tomographic framework for vhr sar image change detection," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 7, pp. 4470–4484, 2019.

[37] F. Rottensteiner, G. Sohn, J. Jung, M. Gerke, C. Baillard, S. Benitez, and U. Breitkopf, "The isprs benchmark on urban object classification and 3d building reconstruction," *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences I-3 (2012), Nr. 1*, vol. 1, no. 1, pp. 293–298, 2012.

[38] F. Rottensteiner, G. Sohn, M. Gerke, J. D. Wegner, U. Breitkopf, and J. Jung, "Results of the isprs benchmark on urban object detection and 3d building reconstruction," *ISPRS journal of photogrammetry and remote sensing*, vol. 93, pp. 256–271, 2014.

[39] D. Tuia, C. Persello, and L. Bruzzone, "Domain adaptation for the classification of remote sensing data: An overview of recent advances," *IEEE geoscience and remote sensing magazine*, vol. 4, no. 2, pp. 41–57, 2016.

[40] E. Maltezos, A. Doulamis, N. Doulamis, and C. Ioannidis, "Building extraction from lidar data applying deep convolutional neural networks," *IEEE Geoscience and Remote Sensing Letters*, vol. 16, no. 1, pp. 155–159, 2018.

[41] Y. Xu, L. Wu, Z. Xie, and Z. Chen, "Building extraction in very high resolution remote sensing imagery using deep learning and guided filters," *Remote Sensing*, vol. 10, no. 1, p. 144, 2018.

[42] F. Wang, G. Zhou, J. Xie, B. Fu, H. You, J. Chen, X. Shi, and B. Zhou, "An automatic hierarchical clustering method for the lidar point cloud segmentation of buildings via shape classification and outliers reassignment," *Remote Sensing*, vol. 15, no. 9, p. 2432, 2023.

[43] K. Liu, H. Ma, L. Zhang, X. Liang, D. Chen, and Y. Liu, "Roof segmentation from airborne lidar using octree-based hybrid region growing and boundary neighborhood verification voting," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 16, pp. 2134–2146, 2023.

[44] F. Wang, G. Zhou, H. Hu, Y. Wang, B. Fu, S. Li, and J. Xie, "Reconstruction of lod-2 building models guided by façade structures from oblique photogrammetric point cloud," *Remote Sensing*, vol. 15, no. 2, p. 400, 2023.

[45] J. Huang, J. Stoter, R. Peters, and L. Nan, "City3d: Large-scale building reconstruction from airborne lidar point clouds," *Remote Sensing*, vol. 14, no. 9, p. 2254, 2022.

[46] E. Lewandowicz, F. Tarsha Kurdi, and Z. Gharineiat, "3d lod2 and lod3 modeling of buildings with ornamental towers and turrets based on lidar data," *Remote Sensing*, vol. 14, no. 19, p. 4687, 2022.

[47] X. Li, F. Qiu, F. Shi, and Y. Tang, "A recursive hull and signal-based building footprint generation from airborne lidar data," *Remote Sensing*, vol. 14, no. 22, p. 5892, 2022.

[48] M. Bizjak, D. Mongus, B. Žalik, and N. Lukač, "Novel half-spaces based 3d building reconstruction using airborne lidar data," *Remote Sensing*, vol. 15, no. 5, p. 1269, 2023.

[49] E. K. Dey, M. Awrangjeb, F. Tarsha Kurdi, and B. Stantic, "Machine learning-based segmentation of aerial lidar point cloud data on building roof," *European Journal of Remote Sensing*, vol. 56, no. 1, p. 2210745, 2023.

[50] Z. Li and J. Shan, "Ransac-based multi primitive building reconstruction from 3d point clouds," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 185, pp. 247–260, 2022.

[51] W. Zhang, Z. Li, and J. Shan, "Optimal model fitting for building reconstruction from point clouds," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 14, pp. 9636–9650, 2021.

[52] L. Li, N. Song, F. Sun, X. Liu, R. Wang, J. Yao, and S. Cao, "Point2roof: End-to-end 3d building roof modeling from airborne lidar point clouds," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 193, pp. 17–28, 2022.

[53] F. Tarsha Kurdi, Z. Gharineiat, G. Campbell, M. Awrangjeb, and E. K. Dey, "Automatic filtering of lidar building point cloud in case of trees associated to building roof," *Remote Sensing*, vol. 14, no. 2, p. 430, 2022.

[54] G. Gröger and L. Plümer, "Citygml–interoperable semantic 3d city models," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 71, pp. 12–33, 2012.

[55] R. Volk, J. Stengel, and F. Schultmann, "Building information modeling (bim) for existing buildings—literature review and future needs," *Automation in construction*, vol. 38, pp. 109–127, 2014.

[56] J. Hyyppa, O. Kelle, M. Lehikoinen, and M. Inkinen, "A segmentation-based method to retrieve stem volume estimates from 3-d tree height models produced by laser scanners," *IEEE Transactions on geoscience and remote sensing*, vol. 39, no. 5, pp. 969–975, 2001.

[57] J. Jung, E. Pasolli, S. Prasad, J. C. Tilton, and M. M. Crawford, "A framework for land cover classification using discrete return lidar data: Adopting pseudo-waveform and hierarchical segmentation," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 7, no. 2, pp. 491–502, 2014.

[58] S. Oh, J. Jung, G. Shao, G. Shao, J. Gallion, and S. Fei, "High-resolution canopy height model generation and validation using usgs 3dep lidar data in indiana, usa," *Remote Sensing*, vol. 14, no. 4, p. 935, 2022.

[59] H. Song and J. Jung, "An object-based ground filtering of airborne lidar data for large-area dtm generation," *Remote Sensing*, vol. 15, no. 16, p. 4105, 2023.

[60] X. Meng, N. Currit, and K. Zhao, "Ground filtering algorithms for airborne lidar data: A review of critical issues," *Remote Sensing*, vol. 2, no. 3, pp. 833–860, 2010.

[61] B. Höfle, M. Vetter, N. Pfeifer, G. Mandlburger, and J. Stötter, "Water surface mapping from airborne laser scanning using signal intensity and elevation data," *Earth Surface Processes and Landforms*, vol. 34, no. 12, pp. 1635–1649, 2009.

[62] H. Song and J. Jung, "Scalable surface water mapping up to fine-scale using geometric features of water from topographic airborne lidar data," *arXiv preprint arXiv:2301.06567*, 2023.

[63] C. Liu, B. Shi, X. Yang, N. Li, and H. Wu, "Automatic buildings extraction from lidar data in urban area by neural oscillator network of visual cortex," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 6, no. 4, pp. 2008–2019, 2013.

[64] S. Du, Y. Zhang, Z. Zou, S. Xu, X. He, and S. Chen, "Automatic building extraction from lidar data fusion of point and grid-based features," *ISPRS journal of photogrammetry and remote sensing*, vol. 130, pp. 294–307, 2017.

[65] J. Niemeyer, F. Rottensteiner, and U. Soergel, "Contextual classification of lidar data and building object detection in urban areas," *ISPRS journal of photogrammetry and remote sensing*, vol. 87, pp. 152–165, 2014.

[66] F. Rottensteiner, "Automatic generation of high-quality building models from lidar data," *IEEE Computer Graphics and Applications*, vol. 23, no. 6, pp. 42–50, 2003.

[67] R. C. dos Santos, M. Galo, and A. C. Carrilho, "Extraction of building roof boundaries from lidar data using an adaptive alpha-shape algorithm," *IEEE Geoscience and Remote Sensing Letters*, vol. 16, no. 8, pp. 1289–1293, 2019.

[68] R. C. Dos Santos, A. F. Habib, and M. Galo, "Weighted iterative cd-spline for mitigating occlusion effects on building boundary regularization using airborne lidar data," *Sensors*, vol. 22, no. 17, p. 6440, 2022.

[69] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.

[70] J. Huang, X. Zhang, Q. Xin, Y. Sun, and P. Zhang, "Automatic building extraction from high-resolution aerial images and lidar data using gated residual refinement network," *ISPRS journal of photogrammetry and remote sensing*, vol. 151, pp. 91–105, 2019.

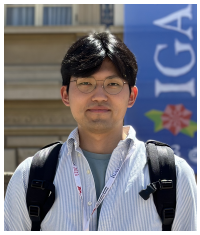
[71] P. Zhang, P. Du, C. Lin, X. Wang, E. Li, Z. Xue, and X. Bai, "A hybrid attention-aware fusion network (hafnet) for building extraction from high-resolution imagery and lidar data," *Remote Sensing*, vol. 12, no. 22, p. 3764, 2020.

[72] Q. Yuan, H. Z. M. Shafri, A. H. Alias, and S. J. b. Hashim, "Multiscale semantic feature optimization and fusion network for building extraction using high-resolution aerial images and lidar data," *Remote Sensing*, vol. 13, no. 13, p. 2473, 2021.

[73] H. Hosseinpour, F. Samadzadegan, and F. D. Javan, "Cmgfnet: A deep cross-modal gated fusion network for building extraction from very high-resolution remote sensing images," *ISPRS journal of photogrammetry and remote sensing*, vol. 184, pp. 96–115, 2022.

[74] R. C. dos Santos, M. Galo, and A. F. Habib, "Regularization of building roof boundaries from airborne lidar data using an iterative cd-spline," *Remote Sensing*, vol. 12, no. 12, p. 1904, 2020.

[75] H. Song and J. Jung, "Challenges in building extraction from airborne lidar data: ground-truth, building boundaries, and evaluation metrics," in *Proceedings of the 30th International Conference on Advances in Geographic Information Systems*, 2022, pp. 1–4.



**Hunsoo Song** earned his B.S. and M.S. in Civil and Environmental Engineering from Seoul National University, South Korea. He is a Ph.D. candidate in Geomatics with a specialization in Computational Science and Engineering at the Lyles School of Civil Engineering, Purdue University, USA. His research focuses on the innovative use of geospatial science to enhance sustainable and resilient environments, notably through the development of an advanced 3D urban mapping system leveraging aerial optical and LiDAR data for geospatial digital twin modeling.

**Jinha Jung** received the B.S. and M.S. degree in Civil, Urban, and Geosystem Engineering from Seoul National University, Seoul, South Korea, in 2003 and 2005, respectively, and the Ph.D. degree in the Lyles School of Civil Engineering, Purdue University, West Lafayette, USA. He is an Associate Professor in the Lyles School of Civil Engineering, Purdue University, West Lafayette, USA. His research interest is the advanced LiDAR data analysis for interdisciplinary research leveraging his specialties in remote sensing, geospatial data science, and high-performance computing.