Research Journal of Agricultural Science, 57 (3), 2025; ISSN: 2668-926X

http://doi.org/10.59463/RJAS.2025.3.15

INTEGRATING UAV PHOTOGRAMMETRY AND PYTHON BASED WORKFLOWS FOR BUILDING FOOTPRINT EXTRACTION

George CRISTIAN, Sorin HERBAN, Carmen GRECEA, Clara – Beatrice VÎLCEANU, Andreea Diana CLEPE

Politehnica University of Timişoara,
2 Traian Lalescu Street, 300223, Timişoara, Romania
Corresponding author:george.cristian@student.upt.ro

Abstract. *Unmanned aerial vehicles (UAVs) have become important tools in modern geospatial data acquisition, enabling rapid and high-resolution mapping of urban and rural environments. This paper investigates the efficiency of UAV-based photogrammetry and Python based workflows for extracting building footprints in areas with high buildings density. The primary objective is to assess how using Python based algorithms in data processing influences efficiency and precision in deliverables in comparison to human operators. Despite the maturity of UAV photogrammetric techniques, the extraction of vector features from derived datasets remains a bottleneck in urban mapping workflows. This research focuses on optimizing the feature extraction stage, particularly building footprint vectorization, by integrating Python-based automation into the workflow. The proposed approach streamlines the transition from processed photogrammetric datasets to structured geospatial outputs, reducing labor and costs. This study introduces an innovative approach by integrating photogrammetric data with automated Python-based feature extraction algorithms, providing a reproducible and scalable solution for urban and rural mapping. Results demonstrate that automated workflows can significantly reduce processing time while maintaining comparable accuracy to manual vectorization. The deliverables generated through this workflow, especially updated orthophoto plans, can be directly used by local authorities for a wide range of applications including urban planning, infrastructure management, cadastral updates and tax collection. The study highlights the importance of maintaining up-to-date geospatial data for informed decision-making and efficient governance.*

Keywords*: UAV Photogrammetry, Building Extraction, Python, Urban Mapping, GIS,*

INTRODUCTION

Surveying buildings is one of the most labour intensive and expensive tasks in suveying and mapping. The complexity arises from the need to coduct measurements on private property and the technical challanges that buildings impose such as loss of GNSS positioning accuracy in proximity to buildings (WuDunn & Zakhor, 2021). Recent advances in technology such as tilt compensated GNSS receivers, integrated laser measument systems and photogrammetric based distance estimation tools tried to solve the issue, the process remains time cosuming and cumbersome to execute in the field (Andaru et al., 2024). The use of unmanned aerial vehicles (UAVs) is a viable and efficient method for data acquisition, offering an alternative to traditional surveying (ZHOU ET AL., 2022). UAVs enable rapid coverage of complex or inaccessible areas without the logistical constraints associated with entering private properties. Among UAV-based technologies, LiDAR and photogrammetry are the two primary approaches for data collection. LiDAR systems provide highly accurate and detailed point clouds with consistent geometric reliability, making them particularly suitable for feature extraction (LIU ET AL., 2025) . However, the high cost of LiDAR sensors and data acquisition limits their widespread use in large-scale projects. In contrast, UAV photogrammetry offers a cost effective solution, capable of producing 3D reconstructions from overlapping images (SADEQ ET AL., 2024). Despite its lower geometric precision and sensitivity to lighting and texture conditions, photogrammetric data remain sufficiently accurate for a wide range of mapping applications (Ji et al., 2024). This study will evaluate whether the photogrammetric approach can produce satisfactory results. By assessing multiple factors this study aims to contribute to the development of scalable, reproducible and reliable method for building footprint extraction.

MATERIAL AND METHODS

A UAV photogrammetric survey was conducted using a DJI Mavic 3 Enterprise that captured nadir and oblique images(at 30°) with a 80% farward overlap and 80% side overlap at an altitude of 100m. Data collected this way ensured comprehensive coverage of building facades and roofs, improving 3D reconstruction (ZHOU ET AL., 2022).

Ground control points were surveyed using a GNSS receiver ensuring accurate georeferencing. The images were processed in Agisoft Metashape to produce a dense point cloud and an orthofotoplan (WUDUNN & ZAKHOR, 2021).

The colored point cloud was classified in several classes: ground, vegetation, noise and buildings. Ground extraction was done using the Cloth Simulation Filter (CSF), which simulates a virtual cloth that drapes over the inverted point cloud (ZHANG ET AL., 2016).

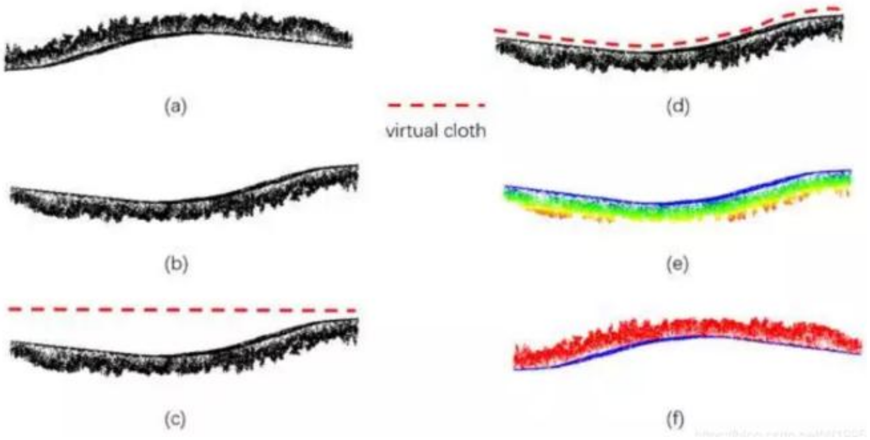

Figure 1. Representation of CSF

As described by (ZHANG ET AL., 2016), the CSF algorithm iteratively relaxes a simulated cloth mesh over the inverted point cloud until it conforms to the terrain surface, separating ground and non ground points based on height differences and geometric smoothness. The algorithm's performance is governed by several key parameters, including cloth resolution, maximum iteration count, and rigidity coefficient, which control the level of surface detail the simulated cloth can capture. Once the simulation reaches the end, the cloth's final surface is treated as the estimated Digital Terrain Model (DTM), and all points located above a specified threshold are classified as non-ground points.

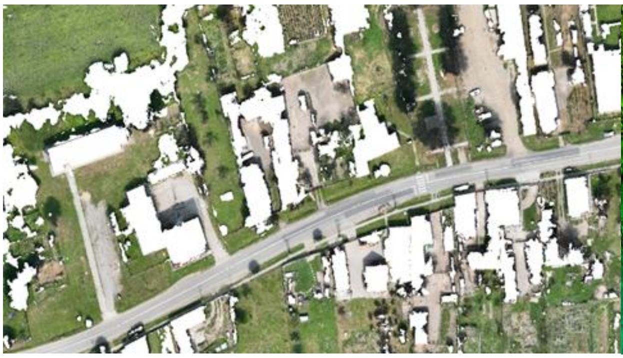

Figure 2. Classified ground points sample

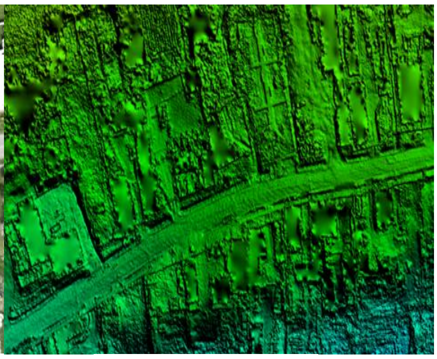

Figure 3. DEM Sample

This physically inspired approach enables the CSF to perform robustly across diverse landscapes without requiring prior knowledge of the terrain morphology, making it especially valuable for photogrammetric point cloud classification (LIU ET AL., 2025). After the ground points were extracted, the remaining point cloud consisted primarily of above-ground features such as buildings, vegetation, and noise.

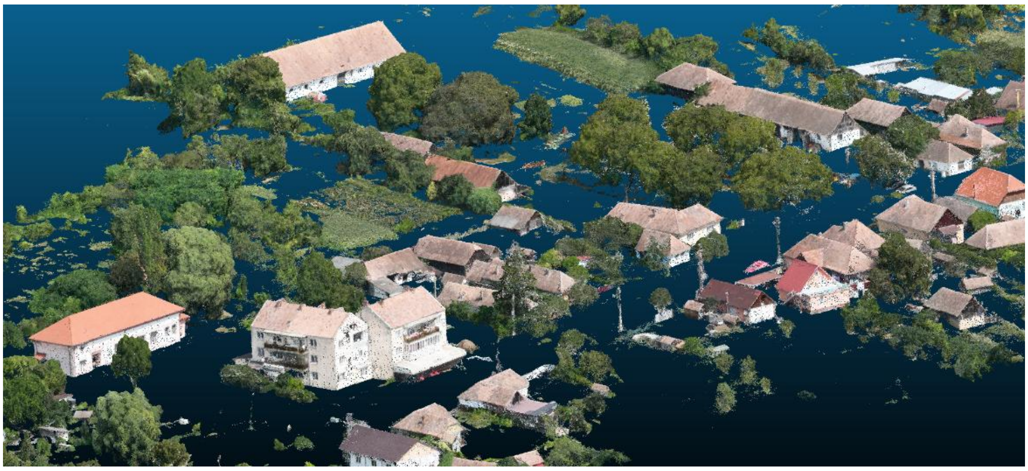

Figure 4. Point cloud sample with the ground points filtered out

To further refine the dataset, vegetation classification was performed based characteristics derived from the RGB imagery associated with each point. The point colors were converted to the HSV (Hue, Saturation, Value) color space, which provides a more perceptually uniform representation of color compared to the standard RGB model (JI ET AL., 2024). In this space, vegetation typically exhibits a Hue range between 90° and 180°, corresponding to various shades of green, with Saturation and Value values between 0 and 100, depending on vegetation type, density, and illumination conditions. Points falling within this spectral range were classified as vegetation, effectively separating them from artificial surfaces such as roofs or walls, which tend to display different color signatures.

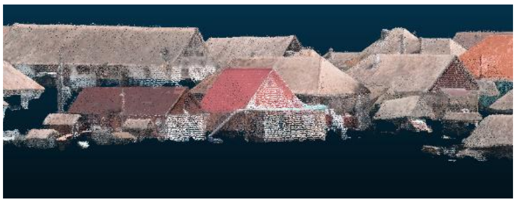

Figure 5. Point cloud after vegetation filter - side view

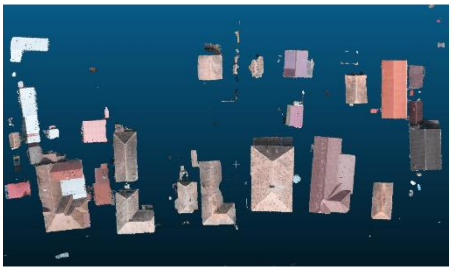

Figure 2. Point cloud after vegetation filter - top view

After vegetation classification, the remaning subset was subjected to additional spatial filtering and noise reduction. Small, isolated point clusters, often caused by reconstruction artifacts or mismatched features in the photogrammetric process, were identified and removed using density-based clustering and distance thresholds. Point clusters with less then 50 points/0.5 m^2 were classified as noise (LI ET AL., 2023). As a result, the refined point cloud predominantly contained building structures, with minimal interference from vegetation or noise. This clean dataset served as a reliable foundation for subsequent stages of analysis, including roof segmentation and building footprint extraction.

After the classification is complete, two raster models were generated:

- DTM (Digital terrain model) interpolated from the points in the ground class;
- DEM (Digital elevation model) interpolated from both ground and building points;

Both model were created at a resolution of 7.7cm/pixel and they form the core input for the extraction algorithm.

The proposed workflow starts with the computation of the normalized Digital Surface Model (nDSM), derived as the difference between the Digital Elevation Model (DEM)—which includes both terrain and above-ground objects—and the Digital Terrain Model (DTM)—which represents only the bare ground surface (Sadeq et al., 2024). Mathematically, this relationship can be expressed as:

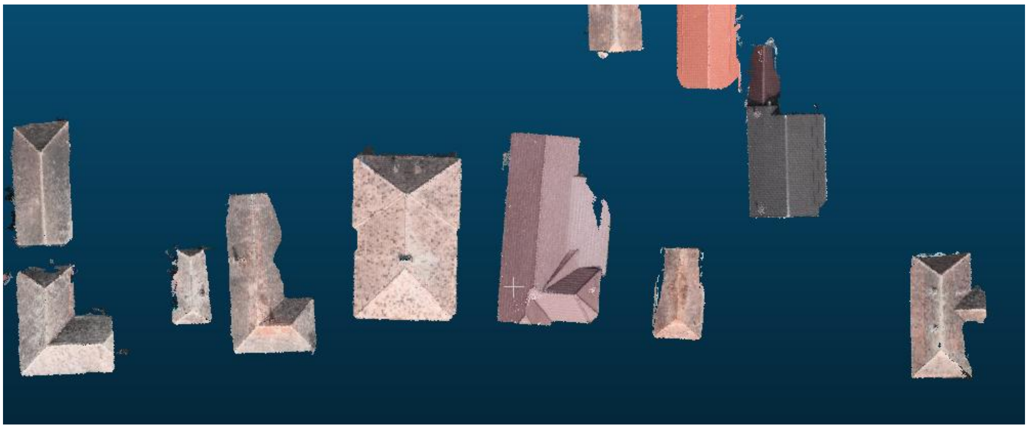

Figure 6. Cleaned point cloud containing buildings class

$$nDSM = DEM – DTM$$

The resulting nDSM quantifies the relative elevation of surface features above ground level, effectively isolating the vertical component associated with buildings. This height-normalized raster serves as the foundation for distinguishing built-up features from terrain.

The next stage involves height thresholding, a process that only retains pixels above a minimum elevation above the ground by using cv2.threshold() function from the openCV library to apply a binary threshhold to the raster (in this case all pixels with an elevation >2.5m are atributed the value 1 and the other pixels become 0) (LIU ET AL., 2025). This threshold can be adjusted depending on the characteristics of the study area and the expected building heights. For this study the chosen value 2.5m so that smaller buildings are not overlooked.

After thresholding, the binary raster representing potential building regions often contains imperfections, including small gaps, spurious artifacts, and irregular boundaries. To address some of these issues, the workflow employs morphological filtering operations, specifically opening and closing (LI ET AL., 2023).

- The opening operation performs an erosion followed by a dilation, effectively removing small noise patches and refining the edges of building regions using thr cv2.MORPH_OPEN function.
- The closing operation performs a dilation followed by an erosion, thereby filling small voids or gaps within individual buildings and bridging discontinuities between closely adjacent segments of the same building using the cv2.MORPH_CLOSE function.

An additional GAP parameter is introduced to control the merging of nearby building segments that belong to a single structure but may appear disconnected due to photogrammetric or rasterization artifacts. This parameter defines a maximum allowable separation distance for merging, ensuring more continuous and geometrically coherent footprints.

Once the morphological refinement is complete, the resulting binary roof mask is converted from raster to vector format through polygonization using OpenCV and GDAL libraries. This step transforms contiguous pixel clusters into vector polygons using the cv2.findContours() function to detect continuous boundaries in the binary mask and the cv2.approxPolyDP() to simplify each contour by reducing vertex density while mentaining overall shape (Andaru et al., 2024). The GDAL library is used to transform detected contours into vector polygons, simplifying and saving them as a shapefile.

RESULTS AND DISCUSSIONS

The proposed workflow successfully generated building footprints with varying fidelity. In the cases where the buildings were isolated and there was no vegetation the output was similar to the shape that was manually drawn.

In areas where vegetation was present, the fidelity of the extracted building footprints was noticeably reduced, a result that aligns with expectations given the characteristics of the input data (JI ET AL., 2024). The irregular geometry and variable height of vegetation introduce local distortions in the surface models derived from photogrammetric reconstruction, leading to inconsistencies along roof boundaries and partial occlusion of building features (SADEQ ET AL., 2024).

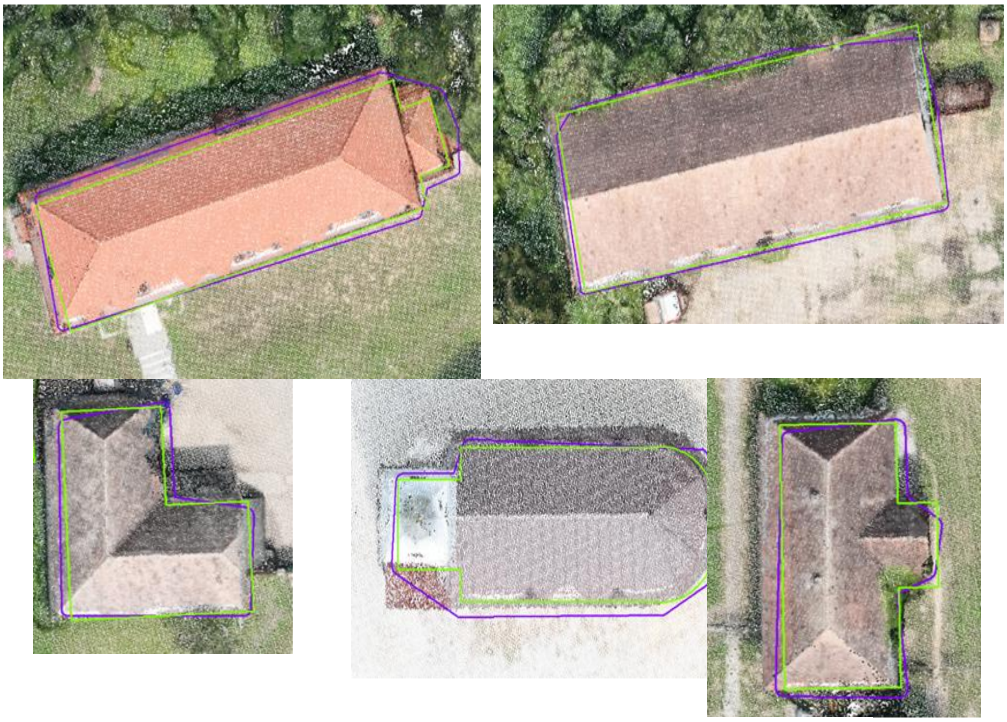

Figure 7. High fidelity outputs, yellow – manually drawn, blue - generated

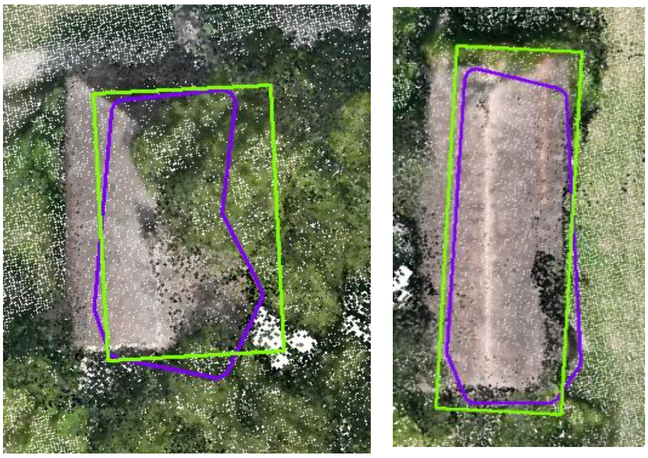

Figure 8. Output affected by vegetation, yellow – manually drawn, blue – generated

The algorithm performed the worst in cases where building density was high. The predominant source of error was the merging of adjacent buildings into a single polygon, resulting in the under-segmentation of individual structures (LIU ET AL., 2025). This limitation is most likely attributed to the inherent constraints of the photogrammetric data acquisition process, where, narrow gaps between buildings are difficult to reconstruct reliably from aerial imagery (ZHOU ET AL., 2022).

Occlusions, limited image overlap, and unfavorable viewing angles reduce the point density and geometric accuracy in these inter-building spaces, leading to ambiguous elevation values in the derived surface models and, consequently, to segmentation errors during footprint extraction.

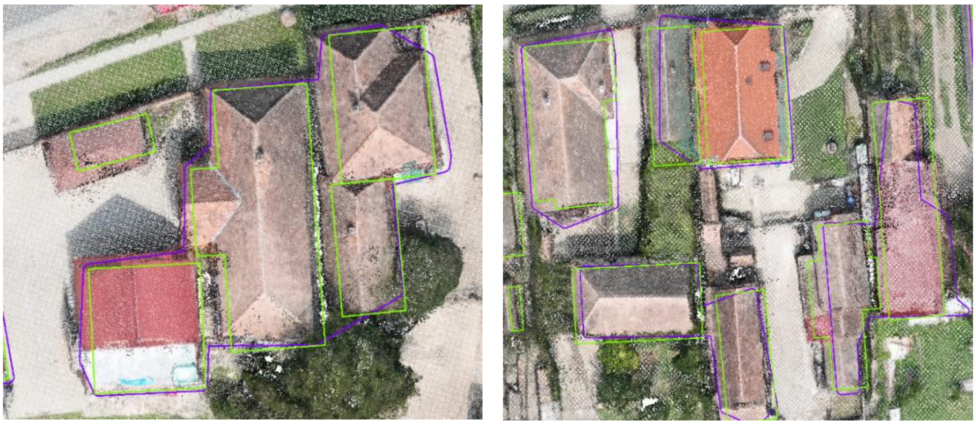

Figure 9. Errors in ouput data, yellow – manually drawn, blue - generated

The results were assesed by comparing the areas of manually digitized buildings with those of the generated polygons. This comparison enabled evaluation of geometric consistency of the extraction process, relative to human interpretation. The results are presented in the following table:

Table 1

Area analysis of building shapes

ID	Area1 [m^2]	Area2 [m^2]	Δ Area[m^2]	Δ Area[%]	Obserervations
1	2.420	0.000	2.420	100	Area corectly filtered out < 20 m^2
2	3.003	0.000	3.003	100	Area corectly filtered out < 20 m^2
3	3.156	0.000	3.156	100	Area corectly filtered out < 20 m^2
4	3.197	0.000	3.197	100	Area corectly filtered out < 20 m^2
5	4.086	0.000	4.086	100	Area corectly filtered out < 20 m^2
6	4.093	0.000	4.093	100	Area corectly filtered out < 20 m^2
7	4.386	0.000	4.386	100	Area corectly filtered out < 20 m^2
8	4.460	0.000	4.460	100	Area corectly filtered out < 20 m^2
9	4.474	0.000	4.474	100	Area corectly filtered out < 20 m^2
10	4.701	0.000	4.701	100	Area corectly filtered out < 20 m^2
11	4.751	0.000	4.751	100	Area corectly filtered out < 20 m^2
12	4.842	0.000	4.842	100	Area corectly filtered out < 20 m^2
13	4.955	0.000	4.955	100	Area corectly filtered out < 20 m^2
14	5.450	0.000	5.450	100	Area corectly filtered out < 20 m^2

15	5.478	0.000	5.478	100	Area corectly filtered out < 20 m^2
16	5.749	0.000	5.749	100	Area corectly filtered out < 20 m^2
17	6.298	0.000	6.298	100	Area corectly filtered out < 20 m^2
18	6.311	0.000	6.311	100	Area corectly filtered out < 20 m^2
19	7.217	0.000	7.217	100	Area corectly filtered out < 20 m^2
20	7.292	0.000	7.292	100	Area corectly filtered out < 20 m^2
21	7.403	0.000	7.403	100	Area corectly filtered out < 20 m^2
22	8.096	0.000	8.096	100	Area corectly filtered out < 20 m^2
23	8.450	0.000	8.450	100	Area corectly filtered out < 20 m^2
24	10.396	0.000	10.396	100	Area corectly filtered out < 20 m^2
25	10.543	0.000	10.543	100	Area corectly filtered out < 20 m^2
26	11.009	0.000	11.009	100	Area corectly filtered out < 20 m^2
27	11.197	0.000	11.197	100	Area corectly filtered out < 20 m^2
28	12.140	0.000	12.140	100	Area corectly filtered out < 20 m^2
29	13.358	0.000	13.358	100	Area corectly filtered out < 20 m^2
30	13.853	0.000	13.853	100	Area corectly filtered out < 20 m^2
31	14.715	0.000	14.715	100	Area corectly filtered out < 20 m^2
32	14.818	0.000	14.818	100	Area corectly filtered out < 20 m^2
33	16.583	0.000	16.583	100	Area corectly filtered out < 20 m^2
34	17.299	0.000	17.299	100	Area corectly filtered out < 20 m^2
35	18.280	0.000	18.280	100	Area corectly filtered out < 20 m^2
36	18.505	0.000	18.505	100	Area corectly filtered out < 20 m^2
37	18.563	0.000	18.563	100	Area corectly filtered out < 20 m^2
38	18.666	0.000	18.666	100	Area corectly filtered out < 20 m^2
39	21.640	0.000	21.640	100	Area incorectly filtered out > 20 m^2
40	22.527	0.000	22.527	100	Area incorectly filtered out > 20 m^2
41	23.954	0.000	23.954	100	Area incorectly filtered out > 20 m^2
42	24.121	0.000	24.121	100	Area incorectly filtered out > 20 m^2
43	26.804	0.000	26.804	100	Area incorectly filtered out > 20 m^2
44	30.107	0.000	30.107	100	Area incorectly filtered out > 20 m^2
45	30.645	0.000	30.645	100	Area incorectly filtered out > 20 m^2
46	33.708	0.000	33.708	100	Area incorectly filtered out > 20 m^2
47	37.647	0.000	37.647	100	Area incorectly filtered out > 20 m^2
48	37.691	0.000	37.691	100	Area incorectly filtered out > 20 m^2
49	38.618	0.000	38.618	100	Area incorectly filtered out > 20 m^2
50	40.832	0.000	40.832	100	Area incorectly filtered out > 20 m^2
51	42.959	0.000	42.959	100	Area incorectly filtered out > 20 m^2

52	46.141	0.000	46.141	100	Area incorectly filtered out > 20 m^2
53	57.931	0.000	57.931	100	Area incorectly filtered out > 20 m^2
54	24.640	22.324	2.316	9.4	Area within desired threshold of ± 10%
55	42.530	26.817	15.713	36.9	Area difference out of the desired threshold of ±10%
56	40.592	31.681	8.911	22.0	Area difference out of the desired threshold of ±10%
57	43.099	31.812	11.287	26.2	Area difference out of the desired threshold of ±10%
58	34.756	35.822	-1.066	-3.1	Area within desired threshold of ±10%
59	50.803	42.433	2.316	4.6	Area within desired threshold of ±10%
60	2.011	43.432	2.191	4.8	Area within desired threshold of ±10% Failed element separation
61	43.612				
62	49.747	43.900	5.846	11.8	Area difference out of the desired threshold of ±10%
63	58.897	47.370	11.527	19.6	Area difference out of the desired threshold of ±10%
64	57.668	49.645	8.023	13.9	Area difference out of the desired threshold of ±10%
65	45.899	52.486	-6.587	-14.4	Area difference out of the desired threshold of ±10%
66	65.585	57.724	7.861	12.0	Area difference out of the desired threshold of ±10%
67	68.354	58.094	10.260	15.0	Area difference out of the desired threshold of ±10%
68	75.024	59.695	15.329	20.4	Area difference out of the desired threshold of ±10%
69	66.317	72.622	-6.306	-9.5	Area within desired threshold of ±10%
70	59.019	78.553	-19.534	-33.1	Area difference out of the desired threshold of ±10%
71	81.538	79.425	2.114	2.6	Area within desired threshold of ±10%
72	101.685	90.350	11.336	11.1	Area difference out of the desired threshold of ±10%
73	80.225	90.608	-10.383	-12.9	Area difference out of the desired threshold of ±10%
74	8.740	96.976	-6.606	-7.3	Area within desired threshold of ±10% Failed element separation
75	81.630				
76	7.752	97.951	-5.282	-5.7	Area within desired threshold of ±10% Failed element separation
77	84.917				
78	88.355	100.059	-11.704	-13.2	Area difference out of the desired threshold of ±10%
79	86.473	102.432	79.489	43.7	Area difference out of the desired threshold of ±10% Failed element separation
80	95.448				
81	37.519	104.319	-8.539	-8.9	Area within desired threshold of ±10% Failed element separation
82	58.261				
83	122.333	107.810	14.523	11.9	Area difference out of the desired threshold of ±10%
84	104.327	110.377	-6.050	-5.8	Area within desired threshold of ±10% Failed element separation
85	95.706	111.899	-16.193	-16.9	Area difference out of the desired threshold of ±10%
86	154.338	121.773	32.615	21.1	Area difference out of the desired threshold of ±10%
87	53.237		3.853	2.9	Area within desired threshold of ±10%

88	79.224	128.608			Failed element separation
89	110.960	135.581	-24.621	-22.2	Area difference out of the desired threshold of ± 10%
90	92.703	140.174	-47.471	-51.2	Area difference out of the desired threshold of ± 10%
91	25.162	143.449			Area difference out of the desired threshold of ± 10%
92	93.873		-24.414	-20.5	Failed element separation
93	138.739	148.900	-10.161	-7.3	Area within desired threshold of ± 10%
94	53.318	151.856			Area within desired threshold of ± 10%
95	89.783		-8.755	-6.1	Failed element separation
96	70.425	157.956			Area within desired threshold of ± 10%
97	101.871		14.340	8.3	Failed element separation
98	37.646	176.042			Area within desired threshold of ± 10%
99	130.922		-7.474	-4.4	Failed element separation
100	77.255	193.165			Area difference out of the desired threshold of ± 10%
101	139.133		23.223	10.7	Failed element separation
102	43.184	206.040			Area within desired threshold of ± 10%
103	154.384		-8.472	-4.3	Failed element separation
104	85.713	208.838			Area within desired threshold of ± 10%
105	122.963		-0.162	-0.1	Failed element separation
106	37.858	219.043			Area within desired threshold of ± 10%
107	162.641		-18.544	-9.2	
108	115.546	232.043			Area difference out of the desired threshold of ± 10%
109	148.164		31.667	12.0	Failed element separation
110	16.472	237.366			Area difference out of the desired threshold of ± 10%
111	98.074				Failed element separation
112	98.081		-24.739	-11.6	
113	185.007	237.712	-52.705	-28.5	Area difference out of the desired threshold of ± 10%
114	72.044	238.244			Area difference out of the desired threshold of ± 10%
115	136.803		-29.397	-14.1	Failed element separation
116	223.039	253.243	-30.204	-13.5	Area difference out of the desired threshold of ± 10%
117	186.113	253.565	-67.452	-36.2	Area difference out of the desired threshold of ± 10%
118	11.489	310.219			Area difference out of the desired threshold of ± 10%
119	256.079		-42.651	-15.9	Failed element separation
120	178.627	324.067	-145.44	-81.4	Area difference out of the desired threshold of ± 10%
121	23.650	339.858			Area within desired threshold of ± 10%
122	74.012				Failed element separation
123	242.788		0.592	0.2	
124	349.722	368.708	-18.986	-5.4	Area within desired threshold of ± 10%

125	172.258	385.071	5.018	1.3	Area within desired threshold of + 10% Failed element separation
126	217.831				
127	388.223	391.476	-3.253	-0.8	Area within desired threshold of + 10%
128	13.021	404.336	-80.299	-24.8	Area difference out of the desired threshold of + 10% Failed element separation
129	43.297				
130	68.316				
131	199.403				
132	13.189	411.323	-72.072	-21.2	Area difference out of the desired threshold of + 10% Failed element separation
133	44.927				
134	74.284				
135	75.334				
136	131.517				
137	115.605	433.410	-62.666	-16.9	Area difference out of the desired threshold of + 10% Failed element separation
138	255.139				
139	28.954	549.353	-90.656	-19.8	Area difference out of the desired threshold of + 10% Failed element separation
140	77.312				
141	91.340				
142	101.759				
143	159.332				

The analysis presented in Table 1 provides an evaluation of the performance of the building extraction algorithm (Area2) relative to manually digitized reference data(Area 1). The table also includes observations describing wheather the extracted footprint met the required criteria or exhibited speciffic issues, such as incorrect filtering or failed feature separation.

The results indicate that from the initial data set of 143 buildings, 100% of buildings with areas <20m^2 were filtered out. From total number of buildings 15, representing 10.49% of total were wrongly filtered out. This indicates that the filtering stage is overly restrictive in certain contexts, or there is insufficient differentiation between small buildings and adjacent structures.

When analyzing feature segmentation we observe that 57 buildings, representing 39.86% of the total data set, failed this step resulting in only 25 polygons.

Looking at the Δ Area values we observe that only 37 out of the 143, representing 25.87% of shapes fall within the area difference threshold of + 10%.

These results align with findings from recent studies emphasizing the difficulty of footprint delineation in dense or vegetated areas using photogrammetry alone (WuDunn & Zakhor, 2021; Andaru et al., 2024). They suggest that integrating cadastral boundaries, adaptive thresholds, or machine-learning segmentation models (Li et al., 2023; Mohamed et al., 2022) could significantly enhance feature differentiation and footprint accuracy.

CONCLUSIONS

The workflow evaluated by this study revealed important limitations: 100% of sub 20 m² objects were correctly discarded, yet 15 buildings (10.49%) above this threshold were wrongly filtered out. 57 buildings (39.86%) failed the segmentation step due to under-segmentation/merging, and only 37 of 143 footprints (25.87%) met the ±10% area-difference criterion. These outcomes indicate that, while UAV photogrammetry is a viable, cost-effective data source for footprint extraction, the current parameterization and separation logic are not sufficiently robust.

The main sources of error were consistent with the characteristics of image-based 3D reconstruction: reduced fidelity in vegetated areas and ambiguous geometry in narrow inter-building gaps led to merged roofs and over-aggressive filtering. Addressing these weaknesses will require both data and algorithmic refinements.

On the data side, strengthening acquisition with increased oblique coverage and cross-track overlap to better resolve facades and alleys and fusing auxillary constraints such as cadastral parcel boundaries to limit cross-parcel merges.

On the algorithmic side, promising directions include adaptive, context-aware thresholds tied to local height statistics, instance-level separation using distance-transform watershed or graph-based splitting on the binary roof mask.

While LiDAR remains superior for automation and structural precision, UAV photogrammetry provides a cost-effective alternative suitable for municipal-scale applications. Future improvements should focus on integrating contextual information, such as cadastral parcel data and adaptive morphological parameters, to reduce over-filtering and enhance roof separation. Incorporating deep learning approaches for object detection and feature segmentation may further improve consistency and scalability, moving UAV photogrammetry closer to the automation performance traditionally associated with LiDAR-based workflows.

BIBLIOGRAPHY

ANDARU, R., ADITYA, T., CAHYONO, B. K., & SANTOSA, P. B. (2024). Automatic DTM and Building Footprint Extraction from Imageries and Point Clouds in Indonesia’s Land Registration Drone Survey: A Roadmap.

BURAY KARSLI, FERRUH YILMAZTURK, MURAT BAHADIR, FEVZI KARSLI, EMIRHAN OZDEMIR (2023). Automatic building footprint extraction from photogrammetric and point-cloud data with I-Octree. ScienceDirect.

CHOLLET, F. (2018). Deep Learning with Python. Manning Publications.

JI, Y., WU, W., WAN, G., ZHAO, Y., WANG, W., YIN, H., TIAN, Z., & LIU, S. (2024). Segment Anything Model-Based Building Footprint Extraction for Residential Complex Spatial Assessment Using LiDAR Data and Very High-Resolution Imagery. Remote Sensing, 16(14), 2661.

LI, R., LIU, W., YANG, L., SUN, S., & HU, W. (2019). DeepUNet: A deep fully convolutional network for pixel-level building extraction. Remote Sensing, 11(9), 1120.

LI, K., DENG, Y., KONG, Y., LIU, D., CHEN, J., MENG, Y., & MA, J. (2023). Prompt-Driven Building Footprint Extraction in Aerial Images with Offset-Building Model. arXiv.

LIU, C., ET AL. (2025). An Improved Point Cloud Filtering Algorithm Applies in Urban Environments (CAP algorithm combining CSF + PTD). Remote Sensing, 17(8), 1452.

LIU, C., WANG, H., FENG, B., WANG, C., LEI, X., & CHANG, J. (2025). Integrating Elevation Frequency Histogram and Multi-Feature Gaussian Mixture Model for Ground Filtering of UAV LiDAR Point Clouds in Densely Vegetated Areas. Remote Sensing, 17(18), 3261.

MOHAMED, S. A., MAHMOUD, A. S., MOUSTAFA, M. S., HELMY, A. K., & NASR, A. H. (2022). Building Footprint Extraction in Dense Area from LiDAR Data using Mask R-CNN. International Journal of Advanced Computer Science and Applications, 13(6). (PDF)

SADEQ, H. A., ET AL. (2024). A Building Footprint Extraction from UAV Imagery Using Deep Learning.

SUN, Y., SALVAGGIO, C., & GUINDON, B. (2019). A robust approach for building footprint extraction from high-resolution satellite imagery using morphological operations and edge detection. Remote Sensing, 11(19), 2266.

WUDUNN, M., & ZAKHOR, A. (2021). 3D Building Detection and Reconstruction from Aerial Drone Imagery. Technical Report UCB/EECS-2021-230.

YANG, Q., ET AL. (2024). *Intelligent processing of UAV remote sensing data for …* (Fusing UAV LiDAR with aerial photogrammetric imagery) ScienceDirect.

ZHAO, K., KAMRAN, M., & SOHN, G. (2020). Boundary Regularized Building Footprint Extraction From Satellite Images Using Deep Neural Network. arXiv.

ZHENG, X., ZHONG, Y., WANG, X., & MA, A. (2020). Building footprint extraction from high-resolution images via spatial residual inception convolutional neural network. Remote Sensing, 12(3), 441.

ZHOU, J., LIU, Y., NIE, G., CHENG, H., YANG, X., CHEN, X., & GROSS, L. (2022). Building Extraction and Floor Area Estimation at the Village Level in Rural China Via a Comprehensive Method Integrating UAV Photogrammetry and the Novel EDSANet. Remote Sensing, 14(20), 5175.