BUILDING DETECTION FROM AERIAL LIDAR POINT CLOUD USING DEEP LEARNING

Shu Su [1, *], Kazuya Nakano [1], Kazune Wakabayashi [1]

[1] AERO ASAHI CORPORATION - (shuu-so, kazuya-nakano, kazune-wakabayashi)@aeroasahi.co.jp
3-14-4, Minamidai, Kawagoe, Saitama, 350-1165, Japan

Commission II, WG II/3

KEY WORDS: Building detection, Neighborhood size, Intensity, RGB, Normal vectors, Aerial LiDAR, Point cloud, Deep Learning.

ABSTRACT:

With the development and widespread application of aerial LiDAR, point cloud data can easily be acquired and used in many fields. The accurate detection of buildings from an aerial LiDAR point cloud has attracted much attention owing to its wide range of applications, such as updating building maps and constructing 3D city models. However, such applications remain challenging in the fields of photogrammetry, remote sensing, and computer vision. In this paper, we discuss the features that contribute to building detection accuracy from an aerial LiDAR point cloud using a deep-learning-based method (KPConv). We evaluated the influence of neighborhood size, intensity, RGB, and normal vectors on building detection. The study area was approximately 6 km^2, consisting of 133 million points and 8,099 buildings. The density of the point cloud data was eight points/m^2. We compared search radii of 4, 10, 25, and 50 m for finding neighboring points. The results suggest that an optimal neighborhood size improves the accuracy of building detection. For searching neighboring points, a radius of 25 m is optimal when the building area is less than 1000 m^2, whereas a radius of 50 m is optimal when the building area is larger than 1000 m^2. We also compared different features as inputs to KPConv for training and testing, such as i) 3D coordinates only, ii) 3D coordinates and intensity, iii) 3D coordinates and RGB, iv) 3D coordinates and normal vectors, and v) 3D coordinates, intensity, RGB, and normal vectors. The results suggest that neither intensity nor normal vectors contribute to the accuracy of building detection, while the features of RGB have a limited effect on the results.

1. INTRODUCTION

With the development and widespread application of laser measuring equipment, point cloud data can be easily acquired. Highly accurate point cloud data obtained through an aerial light detection and ranging (LiDAR) surveys are widely used to create digital elevation models of terrain and 3D city models, as well as for forest management, disaster prevention, and urban planning. Three-dimensional city models are typically constructed at different levels of detail (LOD), which makes it possible to abstract a geometrical object with appropriate detail for its purpose. For example, the CityGML 2.0 defines five LODs for building models. LOD0 is a representation of footprints, LOD1 is a block model with a flat roof, LOD2 is a model with a roof shape, LOD3 is an architecturally detailed model with windows and doors, and LOD4 is LOD3 supplemented with indoor features (Biljecki, 2016). If the building is properly detected from the aerial point cloud, it will be possible to construct a building model from LOD0 to LOD2.

Over the past few decades, researchers have developed automatic building-detection techniques. However, the accurate detection of buildings remains a challenging task, particularly in urban areas. Buildings in urban areas have complex shapes that are close to each other and are often obscured by nearby objects such as trees and shadows. Based on the input data, there are three types of techniques for building detection: i) image only, ii) LiDAR point data only, and iii) fusion of LiDAR point data and other data, such as orthoimages and multispectral images (Huang et al., 2019). However, as the fusion method needs to register different data sources in the same spatial region, accurately registering a point cloud with a wide array of data sources stemming from the same area remains a challenge. For constructing 3D city models, the data sources using LiDAR point clouds are still considered a mainstream approach for building detection. In this study, we focused on techniques that use only LiDAR point data.

Deep learning methods outperform image processing, such as image classification, object detection, and instance detection. However, deep learning methods cannot be easily and directly applied to point clouds; as datasets, point clouds are irregularly sampled, unstructured, and unordered (Hu et al., 2020). PointNet is a novel deep learning architecture that directly uses point clouds as inputs and outputs (Qi et al., 2017). Recently, inspired by the idea of PointNet, many deep learning methods have been developed to directly use point clouds as inputs and outputs (Thomas et al., 2019; Boulch, 2020; Hu et al., 2020). KPConv is regarded as an effective network architecture, which also directly uses point clouds as inputs and outputs for point cloud semantic segmentation, and has achieved robust and accurate results on benchmark datasets, such as Semantic3D, Paris-Lille-3D, and DALES (Thomas et al., 2019). In this study, the KPConv was used for building detection.

A point cloud is an unordered set of spatially localized data. Each point in the point cloud is represented by its 3D coordinates. The simplest and most frequently used spatial relationship between points is their neighborhood. When judging a point as either a building or a non-building point, it is difficult to judge it solely by this point itself. This designation depends strongly on the

* Corresponding author

properties of the neighboring points. Therefore, the selection of the neighborhood and its size is crucial for the accuracy of building detection from point clouds.

Some researchers have used additional features of point clouds to improve the accuracy of building detection. Lodha classified aerial LiDAR data into buildings, trees, roads, and grass using the machine learning algorithm, AdaBoost. Five features, height, height variation, normal variation, LiDAR return intensity, and image intensity were used for classification (Lodha et al., 2007). Maltezos used the raw LiDAR point cloud, as well as entropy, height variation, intensity, distribution of the normal vectors, number of returns, planarity, and standard deviation as inputs for the CNN model to classify the point cloud into buildings, vegetation, and the ground (Maltezos et al., 2019).

Different materials often reflect electromagnetic radiation (specifically infrared in the case of LiDAR) at different intensities. For example, road markings often reflect infrared at higher intensities than nearby points due to most of them usually being white in color. When buildings are obscured by nearby objects such as trees, it is difficult to distinguish them only based on 3D coordinates. With additional information (e.g., color), we can easily distinguish the boundaries between trees and buildings. In such a case, the normal vectors would be used to infer the slope of the surface pointing out a perpendicular direction; for example, a flat-roofed building would have a normal vector pointing upwards, while normal vectors of the vegetation near it would point out in arbitrary and different directions (Maltezos et al., 2019). Therefore, intensity, RGB, and normal vectors may be useful for building detection. However, only a few researchers have considered the impact of each feature (such as neighborhood size, intensity, RGB, and normal vectors) on the accuracy of building detection. In this study, we focused on features that contribute to building detection accuracy from an aerial LiDAR point cloud using a deep-learning-based method. We evaluated the effect of neighborhood size, intensity, RGB, and normal vectors on building detection.

2. EXPERIMENTS

2.1 Study area

The building detection performance was evaluated using aerial laser point cloud data collected by the Leica TerrainMapper aerial LiDAR sensor. The data were acquired over the area of Mashiki Town, Kumamoto Prefecture, Japan. The density of the point cloud data was eight points per square meter. The point cloud data contain 3D coordinates, intensity, and RGB information. A subarea of approximately 6 km^2 was selected as the study area, as shown in Figure 1 (red frame).

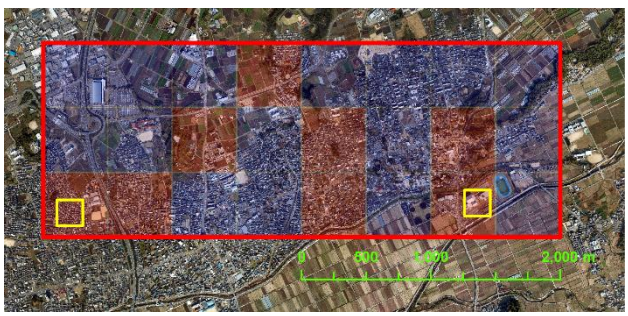


Figure 1. Aerial imagery of the study area.

We divided the point cloud data of the study area into 24 tiles. Each tile was 500 m × 500 m in size. The 16 tiles used for training (blue meshes) and 8 tiles used for testing (red meshes) are shown in Figure 1. The training dataset area was approximately 4 km^2 in size and consisted of 4,736 buildings. The test dataset area was approximately 2 km^2 in size and consisted of 3,363 buildings. Both training and test data contained urban, suburban, and rural scenes. Buildings in urban scenes have typical urban features such as being relatively close to each other and having complex shapes. To qualitatively evaluate the results of building detection, two areas, marked in yellow in Figure 1, were selected. Additional details on the qualitative evaluation are provided in Section 3.2.

A histogram of the building area for the training and test data is shown in Figure 2. Buildings were divided into eight areal ranges: 0–50 m^2 (0 < building area <= 50, the same method applies below), 50–100 m^2, 100–150 m^2, 150–200 m^2, 200–300 m^2, 300–500 m^2, 500–1000 m^2, and larger than 1000 m^2. Buildings less than or equal to 50 m^2 in the area are mostly warehouses, accounting for almost 32% of all buildings. Buildings with areas larger than 300 m^2 accounted for less than 3% of the total number of buildings.

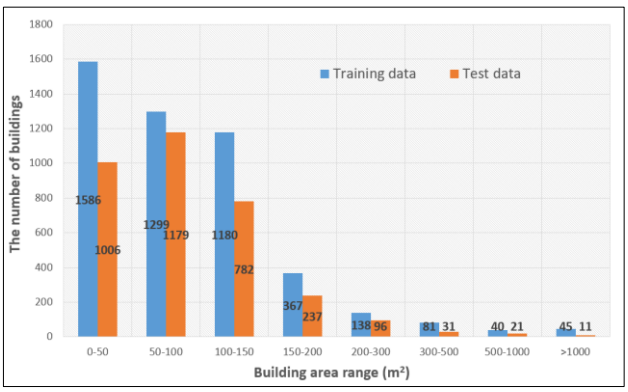


Figure 2. Building area histogram for the training and test data.

Each point in the training area was manually classified as either a building or non-building. To evaluate the automatic building detection results, we also manually labeled the point cloud of the test area. Table 1 lists the number of points per class for the training and test data. The training dataset consisted of approximately 88 million points, of which, 10 million were building points and 78 million were non-building points. The total test dataset consisted of approximately 45 million points, 6 million of which were building points, and 39 million were non-building points. The training and test datasets had a similar distribution, that is, the non-building class accounted for over 87% of all points, and the building class accounted for less than 13% of the total number of points.

	Building	Non-Building	Total
Training data	10,007,984 (11%)	77,943,901 (89%)	87,951,885
Test data	5,891,997 (13%)	38,840,115 (87%)	44,732,112

Table 1. The number of points in the training and the test datasets.

2.2 Performance measures

In this study, we evaluated performance based on three widely used measures: recall, precision, and F1-score for the point-base (number of points), which is given by Equation 1.

$$Recall = \frac{TP}{TP+FN}$$

$$Precision = \frac{TP}{TP+FP} \quad (1)$$

$$F1-score = \frac{2\times Precision\times Recall}{Precision+Recall}$$

where TP is the number of points correctly detected as building points by KPConv and FN is the number of misdetections, that is, points that KPConv failed to detect as building points. FP is the number of over-detections; that is, points were incorrectly detected as building points by KPConv. A high recall value indicates a low misdetection rate, whereas a high precision indicates a low over-detection rate. The F1-score is the harmonic mean of recall and precision. A high F-score indicates both low misdetection and low over-detection rates.

2.3 Comparative experiments

To verify the effectiveness of the neighborhood size and the features of the points, such as intensity, RGB, and normal vectors, we designed two comparative experiments. One compared different search radii of neighborhoods, and the other compared different features as inputs to KPConv for training and testing.

2.3.1 Experiment 1 — different neighborhood radii

There are three types of neighborhood definitions in most common applications: i) spherical neighborhoods, ii) K-nearest neighborhoods, and iii) cylindrical neighborhoods (Weinmann, 2015). KPConv uses a spherical neighborhood because this method improves the robustness of the features and the spatial consistency of the neighborhood (Thomas, 2019). In this study, we also use spherical neighborhoods. We designed neighborhoods with different radii to evaluate the results of building detection, as shown in Table 2. The R4, R10, Baseline, and R50 patterns used neighborhood radii of 4, 10, 25, and 50 m, respectively. For these four patterns, only 3D coordinates were used as inputs to KPConv for training and testing. The intensity, RGB, and normal vector information were not used in this experiment.

Pattern	Radius (m)	3D Coordinates	Intensity	RGB	Normal Vectors
R4	4	○			
R10	10	○			
Baseline	25	○			
R50	50	○			

Table 2. Comparison of different neighborhood radii.

2.3.2 Experiment 2 — different features

In this experiment, we compared the effectiveness of different features listed in Table 3. The baseline pattern in this experiment (highlighted in green in Table 3) was the same as the baseline pattern in Experiment 1 (highlighted in green in Table 2). The baseline-pattern inputs consisted of 3D coordinates only. In the CI pattern, the input consisted of the 3D coordinates and intensity. In the CC pattern, the inputs consisted of 3D coordinates and RGB. In the CN pattern, the inputs consist of 3D coordinates and normal vectors. In the CICN pattern, the inputs consisted of 3D coordinates, intensity, RGB, and normal vectors. Normal vectors were computed using free and open-source software CloudCompare. In this experiment, the same neighborhood size (radius of 25 m) was used for all five patterns.

Pattern	Radius (m)	3D Coordinates	Intensity	RGB	Normal Vectors
Baseline	25	○			
CI	25	○	○		
CC	25	○		○	
CN	25	○			○
CICN	25	○	○	○	○

Table 3. Different features for comparison.

All the patterns in Experiments 1 and 2 were determined using the same training data and test data, as mentioned in Section 2.1. KPConv was trained using 200 epochs for each pattern. All the patterns were trained and tested on a machine equipped with two NVIDIA Quadro RTX 8000 (48 GB) GPUs.

3. RESULTS AND DISCUSSIONS

3.1 Quantitative results

3.1.1 Experiment 1 — different neighborhood radii

Figure 3 shows the results of point-based evaluation comparison for Experiment 1. The minimum and maximum values of precision were 97.7% and 98.0%, respectively. Only a slight difference was observed in the precision values of the four patterns. The R50 pattern achieved the highest recall and F1-score than the other patterns, followed by the baseline pattern, the R10 pattern, and the R4 pattern. Specifically, the R10 pattern yielded a 4.4% recall and had an F1-score greater than the R4 pattern by 2.5%. Furthermore, the baseline pattern achieved a 3.1% recall and an F1-score 1.5% greater than the R10 pattern. This indicates that neighborhood size has a significant impact on the results of building detection. A larger radius of the neighborhood results in improved performance, characterized by a lower misdetection rate and a lower over-detection rate.

Compared to the baseline pattern, the R50 pattern was only slightly better, with a 0.2% recall and 0.1% F1-score improvement over the baseline pattern. However, the time required for the training process was approximately 13 h for the R50 pattern, but only 4.5 h for the baseline pattern. The training duration of the R50 pattern was approximately three times longer than that of the baseline pattern. This implies that a search neighborhood with a radius of 25 m is optimal for building detection, displaying an optimal balance of lower training processing time and higher accuracy.

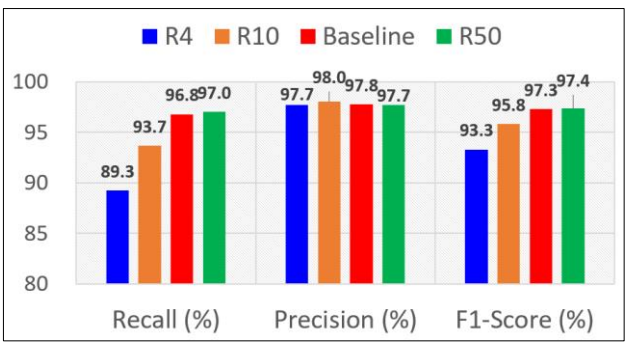
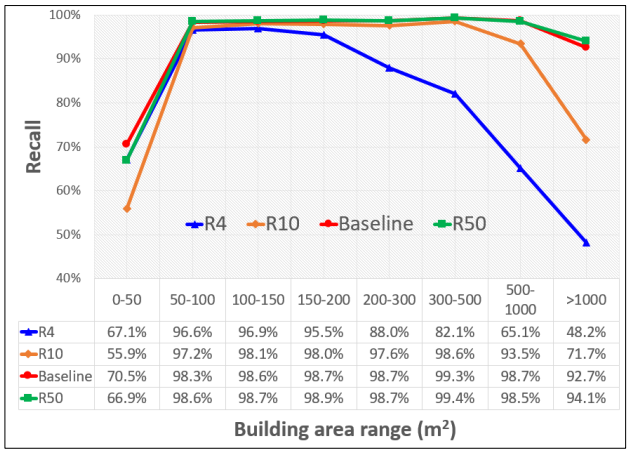


Figure 3. Effect of different neighborhood radii.

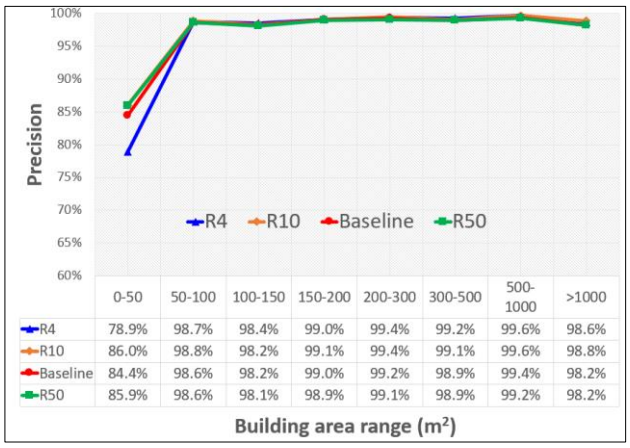
To further analyze the impact of neighborhood size, we plotted the different building area ranges of recall, precision, and F1-score, as shown in Figures 4, 5, and 6, respectively. For all four patterns, with the building area in the range of 50–150 m^2, there was only a minor difference the in recall, precision, and F1-score. In the case of a building area greater than 150 m^2, the recall and

F1-score of the R4 pattern were obviously worse than those of the other patterns, and the R4 pattern also yielded the lowest precision in a building area smaller than 50 m^2. The recall and F1-score of the R10 pattern were also significantly worse than those of the baseline and R50 patterns when the area of the building was greater than 500 m^2. Compared to the baseline pattern, the recall and F1-score of the R50 pattern were slightly better when the area of the building was greater than 1000 m^2 and slightly worse when the area of the building was less than 50 m^2. This implies that a radius of 25 m is optimal for building areas less than 1000 m^2, while a radius of 50 m is optimal for building areas larger than 1000 m^2.



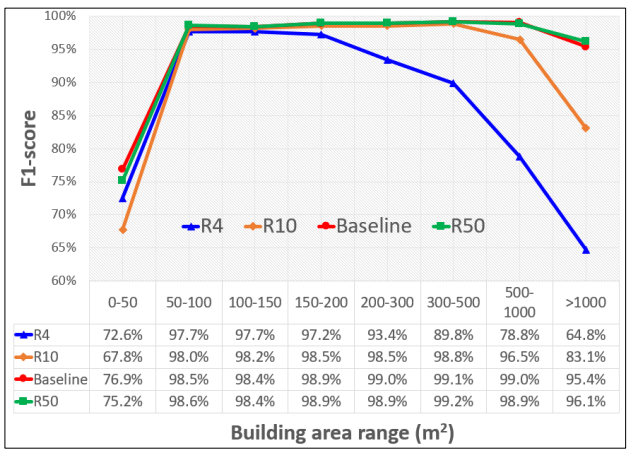
	0-50	50-100	100-150	150-200	200-300	300-500	500-1000	>1000
R4	67.1%	96.6%	96.9%	95.5%	88.0%	82.1%	65.1%	48.2%
R10	55.9%	97.2%	98.1%	98.0%	97.6%	98.6%	93.5%	71.7%
Baseline	70.5%	98.3%	98.6%	98.7%	98.7%	99.3%	98.7%	92.7%
R50	66.9%	98.6%	98.7%	98.9%	98.7%	99.4%	98.5%	94.1%

Figure 4. Effect of building area on recall.



	0-50	50-100	100-150	150-200	200-300	300-500	500-1000	>1000
R4	78.9%	98.7%	98.4%	99.0%	99.4%	99.2%	99.6%	98.6%
R10	86.0%	98.8%	98.2%	99.1%	99.4%	99.1%	99.6%	98.8%
Baseline	84.4%	98.6%	98.2%	99.0%	99.2%	98.9%	99.4%	98.2%
R50	85.9%	98.6%	98.1%	98.9%	99.1%	98.9%	99.2%	98.2%

Figure 5. Effect of building area on precision.



	0-50	50-100	100-150	150-200	200-300	300-500	500-1000	>1000
R4	72.6%	97.7%	97.7%	97.2%	93.4%	89.8%	78.8%	64.8%
R10	67.8%	98.0%	98.2%	98.5%	98.5%	98.8%	96.5%	83.1%
Baseline	76.9%	98.5%	98.4%	98.9%	99.0%	99.1%	99.0%	95.4%
R50	75.2%	98.6%	98.4%	98.9%	98.9%	99.2%	98.9%	96.1%

Figure 6. Effect of building area on F1-score.

3.1.2 Experiment 2 — different features

In Experiment 2, we compared the effectiveness of the different features as inputs of KPConv. For all the patterns, the time required for each training process was approximately 4.5 h. The point-based evaluation comparison results for Experiment 2 are shown in Figure 7. Compared to the baseline pattern, the CC pattern yielded only slightly better results for recall and F1-score, while the CI, CN, and CICN patterns performed slightly worse. These results confirm that neither the intensity nor normal vectors account for the accuracy of the building detection task. The features of RGB contribute little to the results.

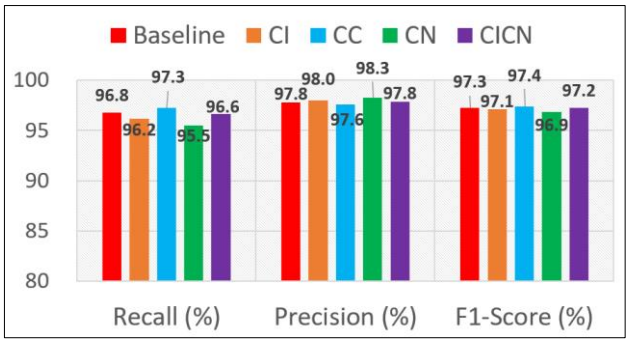
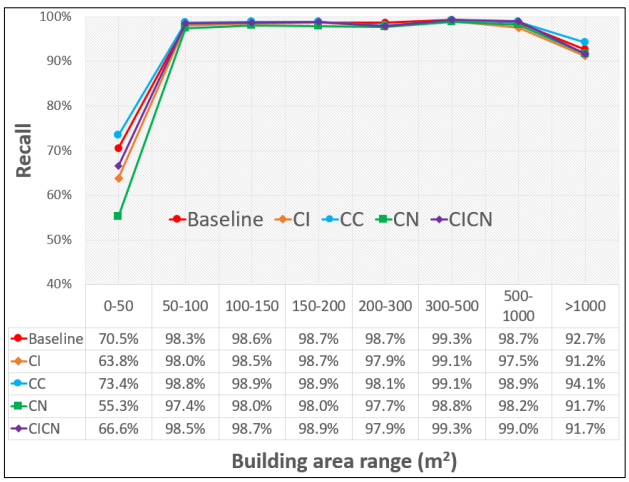
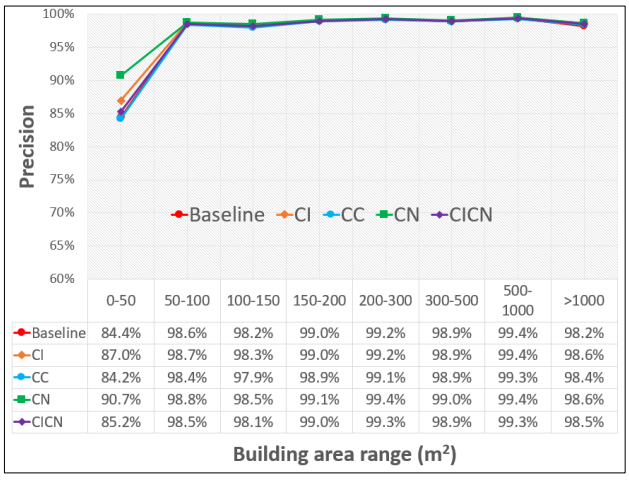


Figure 7. The effect on different features on building detection.



	0-50	50-100	100-150	150-200	200-300	300-500	500-1000	>1000
Baseline	70.5%	98.3%	98.6%	98.7%	98.7%	99.3%	98.7%	92.7%
CI	63.8%	98.0%	98.5%	98.7%	97.9%	99.1%	97.5%	91.2%
CC	73.4%	98.8%	98.9%	98.9%	98.1%	99.1%	98.9%	94.1%
CN	55.3%	97.4%	98.0%	98.0%	97.7%	98.8%	98.2%	91.7%
CICN	66.6%	98.5%	98.7%	98.9%	97.9%	99.3%	99.0%	91.7%

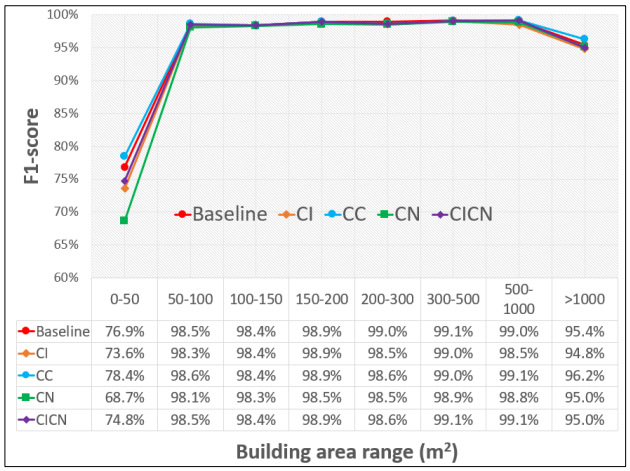
Figure 8. Effect of building area on recall.



	0-50	50-100	100-150	150-200	200-300	300-500	500-1000	>1000
Baseline	84.4%	98.6%	98.2%	99.0%	99.2%	98.9%	99.4%	98.2%
CI	87.0%	98.7%	98.3%	99.0%	99.2%	98.9%	99.4%	98.6%
CC	84.2%	98.4%	97.9%	98.9%	99.1%	98.9%	99.3%	98.4%
CN	90.7%	98.8%	98.5%	99.1%	99.4%	99.0%	99.4%	98.6%
CICN	85.2%	98.5%	98.1%	99.0%	99.3%	98.9%	99.3%	98.5%

Figure 9. Effect of building area on precision.

This contribution has been peer-reviewed.
https://doi.org/10.5194/isprs-archives-XLIII-B2-2022-291-2022 | © Author(s) 2022. CC BY 4.0 License.



	0-50	50-100	100-150	150-200	200-300	300-500	500-1000	>1000
Baseline	76.9%	98.5%	98.4%	98.9%	99.0%	99.1%	99.0%	95.4%
CI	73.6%	98.3%	98.4%	98.9%	98.5%	99.0%	98.5%	94.8%
CC	78.4%	98.6%	98.4%	98.9%	98.6%	99.0%	99.1%	96.2%
CN	68.7%	98.1%	98.3%	98.5%	98.5%	98.9%	98.8%	95.0%
CICN	74.8%	98.5%	98.4%	98.9%	98.6%	99.1%	99.1%	95.0%

Figure 10. F1-score of different building area ranges.

Similar to Experiment 1, we also plotted the different building area ranges of recall, precision, and F1-score, as shown in Figures 8, 9, and 10, respectively. For building areas in the range of 50–1000 m^2, there was not much difference in the recall, precision, and F1-score among all the patterns regarding building detection. When the building area was less than 50 m^2 or larger than 1000 m^2, the CC pattern yielded the highest recall and F1-score.

3.2 Qualitative results

Two examples of building detection for R4, R10, Baseline, R50, CI, CC, CN, and CICN patterns are plotted in Figures 11 and 12. The area in Figure 11 is marked with the left yellow frame in Figure 1, which denotes the urban scene. The area in Figure 12 is marked with the right yellow frame in Figure 1, which denotes a large building scene. The size of the areas shown in Figures 11 and 12 was 200 m × 200 m. The TP, FN (misdetections), and FP (over-detections) are marked in green, red, and blue, respectively. Figure 11 shows that the differences among the eight results (R4, R10, Baseline, R50, CI, CC, CN, and CICN) were not significant. Although there were some misdetections and over-detections, all eight patterns yielded results that were almost consistent with the actual classification of the areas. However, in the case of large-building detection (Figure 12), the results of the baseline pattern clearly outperformed the R10 and R4 patterns. Compared to the baseline pattern, the CC pattern yielded only slightly better results, whereas the CI, CN, and CICN patterns performed slightly worse. This confirms that neither the intensity nor normal vectors account for the accuracy of building detection. Comparing the intensity, RGB, and normal vectors, the results suggest that an optimal neighborhood size will improve the accuracy of building detection.

4. CONCLUSIONS

In this study, we focused on the features contributing to building detection accuracy from an aerial LiDAR point cloud using KPConv. We evaluated the effect of neighborhood size, intensity, RGB, and normal vectors on building detection using two experiments. We not only obtained the results of building detection but also analyzed the effect of different building area sizes. Our results demonstrate that an optimal neighborhood size improves the accuracy of building detection from the point cloud. For searching neighboring points, a radius of 25 m is optimal when the building area is less than 1000 m^2, whereas a radius of 50 m is optimal when the building area is larger than 1000 m^2. Our results also suggest that neither the intensity nor normal vectors contribute to the accuracy of building detection. The RGB feature contributes slightly to detection precision when the building area is less than or equal to 50 m^2 or when the building area is greater than 1000 m^2. All the tested patterns obtained a lower recall, precision, and F1-score for building areas less than 50 m^2, which means it remains difficult and challenging to improve the accuracy of detection of small buildings from point clouds.

ACKNOWLEDGMENTS

We thank the City Bureau of the Ministry of Land, Infrastructure, Transport, and Tourism, Japan, for providing the data used in this work.

REFERENCES

Biljecki, F., Ledoux, H., Stoter, J., 2016. An improved LOD specification for 3D building models. Computers, Environment, and Urban Systems, vol. 59, pp. 25-37.

Boulch, A., 2020. ConvPoint: Continuous convolutions for point cloud processing. Computers & Graphics, vol. 88, pp. 24-34.

Hu, Q., Yang, B., Xie, L., Rosa, S., Guo, Y., Wang, Z., 2020. RandLA-Net: Efficient semantic segmentation of large-scale point clouds. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020, pp. 11105–11114.

Huang, J., Zhang, X., Xin, Q., Sun, Y., Zhang, P., 2019. 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.

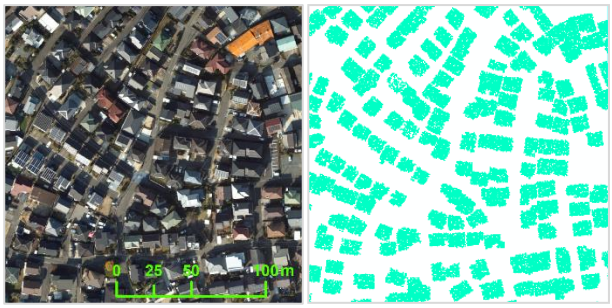
Lodha, S.K., Fitzpatrick, D.M., Helmbold, D.P., 2007. Aerial lidar data classification using AdaBoost. In Proceedings of the International Conference on 3-D Digital Imaging and Modeling. IEEE, Montreal, Canada, 21–23 August, pp. 435–442.

Maltezos, E., Doulamis, A., Doulamis, N., Ioannidis, C., 2019. Building extraction from LiDAR data applying deep convolutional neural networks. IEEE Geoscience and Remote Sensing Letters, vol. 16, pp. 155–159.

Qi, C. R., Su, H., Mo, K. and Guibas, L. J., 2017. Pointnet: Deep learning on point sets for 3d classification and segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 652–660.

Thomas, H., Qi, C. R., Deschaud, J.E., Marcotegui, B., Goulette, F. and Guibas, L. J., 2019. Kpconv: Flexible and deformable convolution for point clouds. In Proceedings of the IEEE International Conference on Computer Vision, pp. 6411–6420.

Weinmann, M., Jutzi, B., Hinz, S., Mallet, C., 2015. Semantic point cloud interpretation based on optimal neighbourhoods, relevant features and efficient classifiers. ISPRS Journal of Photogrammetry and Remote Sensing, vol. 105, pp. 286-304.



(a) Aerial image (b) Ground Truth

(c) R4 (d) R10

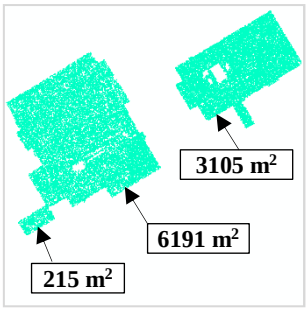
(e) Baseline (f) R50

(g) CI (h) CC

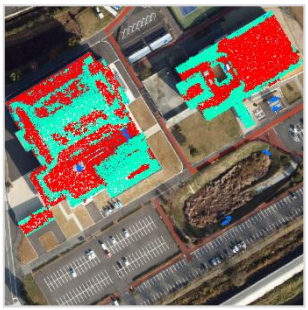
(i) CN (g) CICN

Figure 11. Examples of building detection.
, TP; , FN (misdetections); , FP (over-detections)

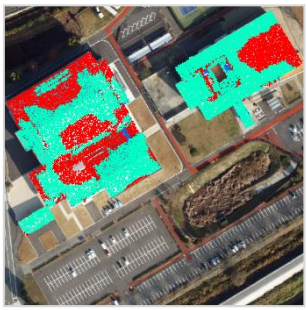
(a) Aerial image



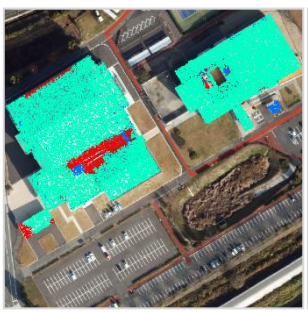
(b) Ground Truth



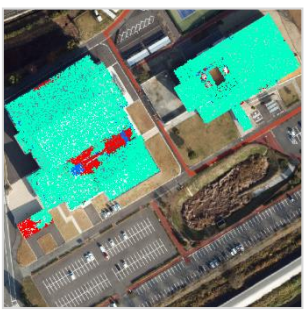
(c) R4



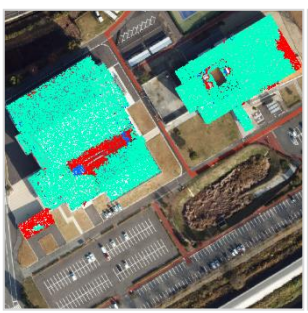
(d) R10



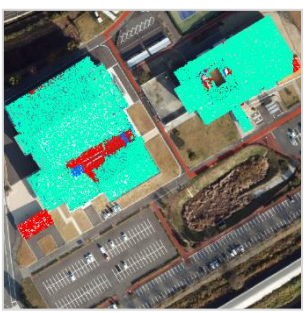
(e) Baseline



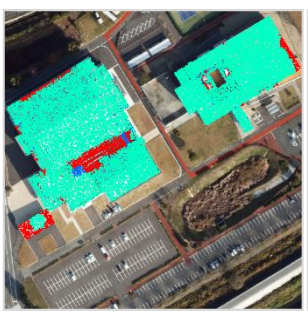
(f) R50



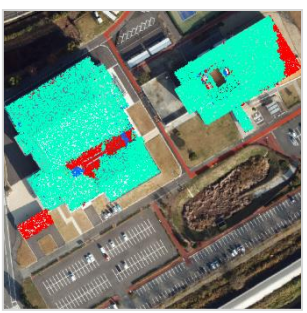
(g) CI



(h) CC



(i) CN



(g) CICN

Figure 12. Examples of large building detection.
, TP; , FN (misdetections); , FP (over-detections)