BUILDING CHANGE DETECTION IN AIRBORNE LASER SCANNING AND DENSE IMAGE MATCHING POINT CLOUDS USING A RESIDUAL NEURAL NETWORK

F. Politz[1]*, M. Sester[1]

[1] Institute of Cartography and Geoinformatics, Leibniz University Hannover, Germany - (politz, sester)@ikg.uni-hannover.de

Commission II, WG II/6

KEY WORDS: Building Change Detection, Point Cloud Processing, Airborne Laser Scanning, Dense Image Matching, Jensen-Shannon-distance, density-independent, Deep Learning.

ABSTRACT:

National Mapping Agencies (NMAs) acquire nation-wide point cloud data from Airborne Laser Scanning (ALS) sensors as well as using Dense Image Matching (DIM) on aerial images. As these datasets are often captured years apart, they contain implicit information about changes in the real world. While detecting changes within point clouds is not a new topic per se, detecting changes in point clouds from different sensors, which consequently have different point densities, point distributions and characteristics, is still an on-going problem. As such, we approach this task using a residual neural network, which detects building changes using height and class information on a raster level. In the experiments, we show that this approach is capable of detecting building changes automatically and reliably independent of the given point clouds and for various building sizes achieving mean F_1-Scores of 80.5% and 79.8% for ALS-ALS and ALS-DIM point clouds on an object-level and F_1-Scores of 91.1% and 86.3% on a raster-level, respectively.

1. INTRODUCTION

Every few years NMAs have to acquire point cloud data to fulfil their tasks and responsibilities in keeping public databases such as Digital Terrain Models or cadastre data up-to-date. Previously, those point clouds were exclusively acquired from ALS sensors. In recent years, deriving point clouds from aerial images as a secondary product using Dense Image Matching became more common. As those point clouds are often years apart, they indirectly contain all changes on the surface, which happened in between the recordings. However, due to a lack of automation in this context, the NMAs often extract building changes manually either by on-site surveying teams or by manually searching for changes in aerial images.

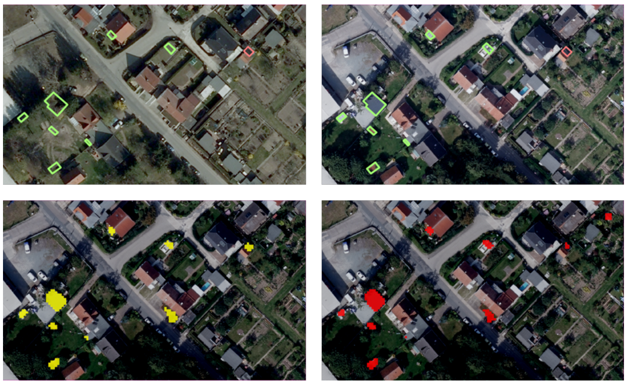


Figure 1. Change detection results using the $HC + C_{building}$ input at $\tau = 0.5$. Top row: Orthophotos from 2010 (left) and 2016 (right) with the reference classes *new* in green and *demolished* in red. Bottom row: binary change detection results for ALS-ALS (left) and ALS-DIM point clouds (right).

While ALS and DIM point clouds contain the 3D information about the earth's surface, they have unique characteristics as pointed out by Mandlburger et al. (2017). Besides different point accuracies and point densities, which in case of DIM point clouds rely on the resolution of the aerial images as well as on the texture of the objects themselves, their behaviour towards vegetation may be the most challenging difference in context of change detection. While a laser beam penetrates the foliage and consequently returns point information from the ground below and places on a tree itself, a DIM point cloud only contains points, which are visible in multiple aerial images. As such, they often only represent the tree top. Similar to vegetation, a laser beam is also able to penetrate transparent roofs and return points from the inside of a building, while DIM point clouds capture the geometry of the roof itself. Consequently, the geometry between those two point cloud types appears like a change, when in reality nothing changed. Height-based change detection methods may detect these as false positives. An example of this issue is visible in fig. 1, where a garden house with a transparent roof in the north-east of that area was detected by our method in the test case involving ALS and DIM point clouds, but not in the one where only ALS point clouds are used.

Change detection between two or more point clouds has been researched for more than two decades. Murakami et al. (1999) detected binary height changes between two normalised Digital Terrain Models using a manually set threshold. Later, these detected regions were classified into building and non-ground classes, from which the latter one got filtered out of the detection results (see also Teo and Shih (2013)). Instead of working on 2D rasters, the Iterative Closest Point algorithm is used by Matikainen et al. (2010) or Scott et al. (2018) to detect 3D translations between two point clouds. Detecting changes directly between two point clouds by comparing the point distances became a popular research topic in the last decade, e.g. by Richter et al. (2013) or Williams et al. (2021).

* Corresponding author

This contribution has been peer-reviewed.
https://doi.org/10.5194/isprs-archives-XLIII-B2-2022-625-2022 | © Author(s) 2022. CC BY 4.0 License.

More complex methods like the multi-scale model to model cloud comparison algorithm (M3C2) focus on point pairs along a cylinder in normal vector direction to improve the point matching and to provide a statistical significance test to the detection (Lague et al., 2013). Recently, Winiwarter et al. (2021) enhanced M3C2 with error propagation considering the measurement and registration uncertainty on terrestrial laser scans. Instead of detecting changes separately from the classification task, Tran et al. (2018) proposed a joint approach using manually calculated features as well as a Random Forest classifier to classify the points directly into complex change classes such as *new building*. Concerning change detection using DIM point clouds, Zhou et al. (2020) included ALS point cloud data in the construction process of DIM generation to verify unchanged buildings, but also to detect small changes in the buildings. Zhang et al. (2019) proposed a convolutional neural network to detect building changes using the Digital Surface Model from ALS and DIM point clouds in raster format with an additional RGB image provided by the DIM point cloud as input. The network outputs a single binary result for each image patch. Patches are later aggregated to the final detection result if certain thresholds are met in the detection process. While their work is the closest research to this study in respect to the point clouds used, there are some major differences in our approach.

To support the workflow of the NMAs, we propose a change detection algorithm, which is able to detect building changes independent of its area sizes and which results in a reliable and comparable detection accuracy for ALS and DIM point clouds. For maximal practical usability, the goal of our change detection is to achieve a high recall rate while keeping false positive numbers as low as possible. Consequently, the correct boundary of a change is not the focus of this work. To be as general as possible, we use a 2D raster to become independent of the underlying point densities, but also to be able to process large areas more efficiently. Each raster cell has the ability to detect a change without requiring additional assumptions about the building area size like Zhang et al. (2019) did. As such, it is able to detect point clouds as small as a single raster cell. However, compared to prior work of us in Politz et al. (2021), where each cell was independently analysed, this study uses a residual neural network to introduce non-linear feature extraction as well as neighbour information to the change detection task. Height and class information are pre-processed from the point clouds on the raster-level and serve as input to the network, which outputs change probabilities. The contributions of this paper can be summarised as follows:

- A modified calculation of the Jensen-Shannon distance (JSD) provides a smaller binning on a technical level and consequently reduces the amount of false positive detections on building borders. This is an enhancement of the JSD algorithm proposed by Politz et al. (2021), which already demonstrated superior detection results and flexibility when compared to a normal height threshold.
- A simple residual neural network is trained on ALS-ALS point clouds and is able to detect building changes in ALS-ALS and ALS-DIM scenarios while retaining a comparable quality independent of the point cloud types used.
- Different height and class input combinations are explored and analysed to understand their influence on the change detection results in regards to the overall accuracy, to the point cloud types used, to different building area sizes and to various change class types.

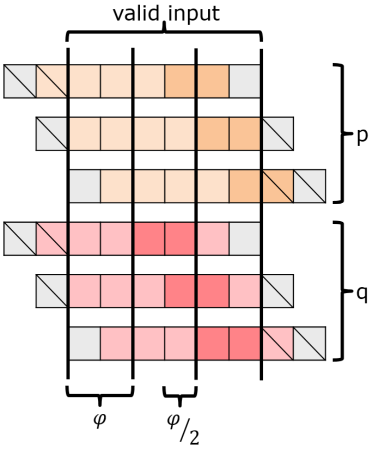


Figure 2. Schematic visualisation of the proposed height change calculation using JSD. The bins of the original histogram of p and q are coloured with orange and red, respectively, where a darker hue indicate higher values. Padded bins are visualised in grey. Bins with a slash are not considered in the JSD calculation.

2. METHODOLOGY

2.1 Input

In order to work on any point cloud type and also on different point densities, a regular 2D raster is created and used as input for the change detection network, which outputs a change detection probability for each cell. For each point cloud at time t_1 and t_2 a 2D raster with a cell size of 1m² is created, which is cut into 100x100m² disjoint raster images. For each cell, height and class values are determined, which serve as input for the network.

2.1.1 Height information The height change probability (HC) is calculated using the $\log_2$ on the Jensen-Shannon Distance (JSD). JSD is the square root of the Jensen-Shannon divergence, which is the symmetric version of the Kullback-Leibler divergence $D(\cdot)$. It is using the mean distribution m and can be calculated using eq. 1 and 2:

$$HC = \log_2 JSD(p \,||\, q) \tag{1}$$

$$JSD(p \,||\, q) = \sqrt{\frac{D(p \,||\, m) + D(q \,||\, m)}{2}}, \tag{2}$$

where p = normalised height distribution of t_1
q = normalised height distribution of t_2
m = mean height distribution with $m = (p+q)/2$.

$\log_2$ is used to transform $JSD(p \,||\, q)$ in the interval between 0 and 1, where values close to 1 signal a change in the underlying distributions p and q, while values close to 0 indicate similar distributions and consequently no change (Lin, 1991). The normalised height distributions p and q are calculated for each raster cell as histograms, which count the amount of points within a bin of size φ. The valid range for those histograms is defined by the extreme values of both point clouds within a 1km² tile. As Politz et al. (2021) already pointed out, having a fixed width in vertical direction can cause false positives whenever point clusters fall into two adjacent bins at one point in time, but not the other time resulting in two different distributions and consequently higher JSD values.

Fig. 2 demonstrates this issue on the middle rows of p and q, where high values are completely within a bin of size φ in p, but are contained in two bins in q. While setting φ to a smaller value may solve this problem, Politz et al. (2021) showed that larger φ perform better overall. As such, this work only sets φ smaller on a technical level. While the overall JSD is still calculated on a bin size of φ, p and q are constructed using $\varphi/2$ and aggregated to φ bins for JSD calculation. The histograms are padded by 0's on both ends and are shifted by ± 1 bin on both distributions p and q allowing maximal flexibility. Within the valid input range, all values are normalised to the sum 1. Then, the JSD is calculated for all possible combinations of p and q variants, where the lowest JSD value is used as final value. Similar to Politz et al. (2021), we set φ to 0.5m.

2.1.2 Class information Besides the height, class information is also given to the network as input. The approach assumes that each point has a class label c_n given a class distribution $C_N = \{c_1, ..., c_n\}$ with a set amount of classes $n \in N$. Let C_{t_i} be the class distribution of all points within a raster cell at time t_i with $i \in \{1, 2\}$. Furthermore, let $C_{t_i,max}$ be the majority class of C_{t_i} and $C_{t_i,n}$ be the relative frequency for the n-th class in that cell.

In this work, several ways to include class information as network input have been investigated. As neural networks have the ability to extract high level features on their own, we define $C_{t_{1/2}}$ as the concatenated class information of C_{t_1} and C_{t_2}. We further define $C_{t_{1/2},building}$ as the concatenated version of $C_{t_1,n}$ and $C_{t_2,n}$, where n equals the building class. Lastly, we define $CC_{t_{1/2}}$ similar to Politz et al. (2021), where either time t_1 or t_2 have to be exclusively building as majority class as shown in eq. 3 and 4:

$$CC_{t_{1/2}} = \begin{cases} 1, & if \quad f(C_{t_1,max}) \dot{\vee} f(C_{t_2,max}) \\ 0, & else \end{cases} \tag{3}$$

$$f(x) = \begin{cases} 1, & if \quad x = C_{t_i,building} \\ 0, & else \end{cases} . \tag{4}$$

For the remainder of this paper, we exclude the $t_{1/2}$ index for an easier understanding resulting in C, $C_{building}$ and CC as investigated options.

2.2 Residual Network

The proposed residual network used in this paper is shown in fig. 3. The network uses the residual blocks developed by He et al. (2016) as building blocks, which explicitly construct:

$$y = F(x, \{W_i\}) + W_s x. \tag{5}$$

The function $F(x, \{W_i\})$ represents the residual mapping, which consists of two 2D convolution layers ($conv$) with a filter size of $3 \times 3 \times c$, where c is the amount of channels. Each convolution is followed by batch normalisation (bn) and a rectified linear unit ($relu$). The result of F is concatenated with the original output x followed by another $relu$ operation. If the input and output dimensions are different, an additional identity mapping W_s is applied like it was suggested by He et al. (2016). Here, W_s is implemented as a $1 \times 1 \times c$ convolution.

Taking into account that training data sets for point cloud change detection are usually not that large, we constructed a small network with only around 750,000 trainable parameters (see fig. 3). The chosen height and class raster files at t_1 and t_2 represent the input of the network. The network consists of an encoder part and a small sigmoid classifier. The encoder has two levels, where the amount of channels c is doubled in each level. A level contains two basic residual blocks. The sigmoid classifier is implemented as a $3 \times 3 \times 1$ convolution followed by a sigmoid activation projecting the results in the range between $\{0, 1\}$. During our experiments, we tested different amounts of levels and blocks per level. In addition, other network structures with varying complexity such as a Siamese Neural Network with two shared encoders, which were concatenated before the classification, have been tested. However, our experiments yielded lower quality detections using these more complex networks indicating overfitting issues.

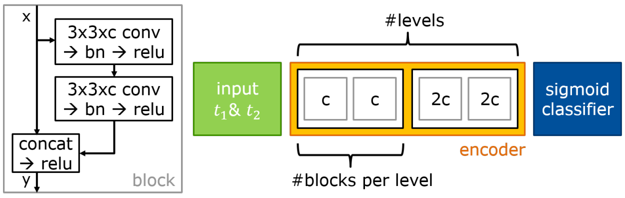


Figure 3. Proposed residual network.

As changes in buildings are rare, the ratio between detected and background pixels is quite unbalanced. Consequently, we used the Tversky loss during training. The Tversky loss proposed by Salehi et al. (2017) supports unbalanced datasets as it concentrates on detected and reference pixels and ignores the vast amount of background pixels. The Tversky loss is defined as

$$TverskyLoss = 1 - \frac{TP + \epsilon}{TP + \alpha FP + \beta FN + \epsilon}. \tag{6}$$

TP, FP and FN are the amount of true positives, false positives and false negatives in the prediction result, respectively. ϵ is used for mathematical stability and is set to $\epsilon = 10^{-6}$. α and β are weighting factors for FP and FN. Different values for α and β were tested. In the end, $\alpha = 0.3$ and $\beta = 0.7$ were chosen as they yielded high recall rates, which matches the observations from Salehi et al. (2017) using the same parameter values.

The Adam optimizer is used with an initial learning rate of 10^{-4}, which gets halved in value, whenever a plateau is reached (Kingma and Ba, 2014). During training, the input is randomly flipped horizontally and vertically to enhance the training set. The training is stopped early, once no significant improvement on the validation loss is detected. The batch size is set to 16. The network is trained five times in order to reduce the amount of randomness caused by initialisation. After training, the final detection value for a cell is determined by the maximal value of that cell over the predictions from the network ensemble. The maximal value is chosen to further support high recall rates.

2.3 Evaluation

The evaluation takes place on an object- and raster-level. As first step for the object-level evaluation, the prediction values are binarized in order to isolate detections from background. Outputs at or above a threshold τ are set to 1 and remain a detection, while every cell with a value below that threshold is set to 0.

In a second step, separate objects are extracted using a connected component algorithm with an 8-way neighbourhood. On the one hand, if a prediction overlaps two or more reference objects, the prediction is split and the predicted pixels are matched to the closest reference object. On the other hand, if multiple predicted regions overlap a reference object, those are treated as one object during the evaluation step. Finally, for each predicted and referenced object pair, the F_1-Score is calculated as:

$$F_1 - Score = \frac{TP}{TP + \frac{1}{2}(FP + FN)}. \tag{7}$$

On object-level, we evaluate the results using the mean F_1-Score of all matched object pairs regarding to the threshold τ. For the raster-level evaluation, the F_1-Score is calculated for the complete dataset independent of single objects.

3. EXPERIMENTS

To evaluate the influence of the network in respect to the detection quality, the input values defined in section 2 serve as a baseline detection method. For the remainder of this paper, the baseline versions have an additional index *raw* and will be called HC_{raw}, CC_{raw} and $HC_{raw} + CC_{raw}$, where the latter one is the product of HC_{raw} and CC_{raw}. Detections from the network will not have any additional index. We define ALS-ALS as the change detection between two ALS point clouds and ALS-DIM as a change detection between an ALS and DIM point cloud. Experiments involve results for the ALS-ALS and ALS-DIM test set in regards to the used input types, threshold τ, overall accuracy, ground area size and change type.

3.1 Dataset

Three point cloud datasets from 2012 and 2016 are used to evaluate the proposed change detection method. The point clouds are provided by the NMA of Mecklenburg-Vorpommern (AFGVK) and cover a 15km² area south-east of Rostock, Germany. Captured during national aerial flight missions, the dataset contains two ALS point clouds from 2012 and 2016 with an approximate point density of 5 and 12 points/m² as well as a DIM point cloud, which aerial images were captured in 2016 and processed with the SURE software, resulting in a point density of about 96 points/m² (Rothermel et al., 2012). The absolute point accuracy for the ALS point clouds is 30cm in horizontal and 15cm in vertical direction, while it is 20cm in horizontal and 30cm in vertical direction for the DIM point cloud. All point clouds are manually classified into the classes ground, building, water, non-ground and bridge. Non-ground contains vegetation, power supply lines and cars.

The region is characterised by high and low density, urban buildings and demonstrates several different building change types during this four year period. Reference data about detected changes were manually collected as vector data in four different classes: *new*, *demolished*, *construction* and *exchanged*. A single building can consist of multiple change objects and types. The complete distribution of the reference classes for the test set in respect to area size and change type can be found in fig. 6 in the top row. While *new* and *demolished* expect to have building points at time t_1 or t_2, cells with *construction* or *exchanged* contain building points at t_1 and t_2. *construction* is defined by a height change caused by constructions, which may added another floor to an existing building. *exchanged* is characterised by constructions, which yielded a similar height between new and old buildings like replacing a roof.

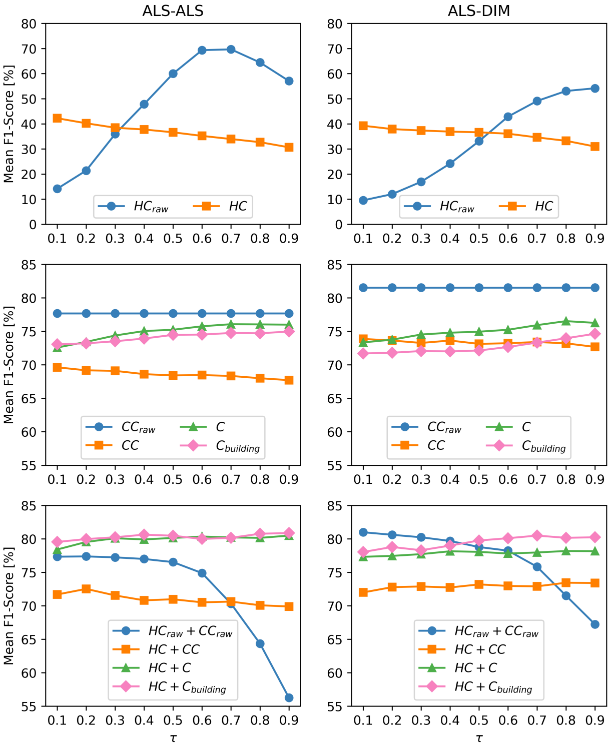


Figure 4. Mean F_1-Scores on object-level for all input types depending on threshold τ for ALS-ALS and ALS-DIM changes.

For training purposes, we split the data in a training, validation and test set. Once the point cloud data is rasterised into non-overlapping images with a 100x100m² size as explained in section 2, all images without any changes are removed from the dataset resulting in 385 images overall. As the images are cut according to their position, building objects may appear on different raster images. 20% of the images are randomly sampled and are reserved as test set. Within the remaining 80%, another 20% are split as a validation set. During any split, all change types are ensured to be represented in each set. In the end, the training, validation and test set contain 244, 63 and 78 raster images, respectively. 10,080 raster cells (1.3%) of the test set are labelled as changes. As the proposed height change input should mostly be independent of the point cloud type, the network is only trained on the ALS-ALS training set as it is more commonly available by NMAs. The same network is then used to detect changes for the ALS-ALS and ALS-DIM test sets.

3.2 Threshold τ

Threshold τ is used to transform the detection probabilities between 0 and 1 into two distinct change and non-change classes. Fig. 4 visualises the relationship between the input type and the threshold τ and its resulting mean F_1-Score based on the matched object pairs, which remain as changes after thresholding. In addition, the mean F_1-Scores for $\tau = 0.5$ are listed in table 1. While input types, which contain HC_{raw}, are sensitive towards τ, the mean F_1-Score of the remaining input combinations are quite indifferent towards τ and only fluctuate around a few %. When comparing the raw input variants with those from the network, the raw versions achieve higher mean F_1-Scores for the majority of τ values in both test setups.

Input combination	ALS-ALS	ALS-DIM
HC_{raw}	60.0	33.1
HC	36.6	36.6
CC_{raw}	77.7	**81.5**
CC	68.4	73.1
C	75.2	75.0
$C_{building}$	74.5	72.1
$HC_{raw} + CC_{raw}$	76.5	78.8
$HC + CC$	71.0	73.2
$HC + C$	80.1	78.1
$HC + C_{building}$	**80.5**	79.8

Table 1. Mean F_1-Scores [%] calculated on object-level at $\tau = 0.5$ for both test sets. Best results are marked bold.

Especially for HC and HC_{raw}, the mean F_1-Scores in the ALS-ALS test set are about 35% apart at $\tau = 0.7$. When compared to other variants, HC and HC_{raw} perform quite poorly, which is mostly caused by a large amount of FPs (see table 2 or 3 or fig. 5 for examples). Within the class information only variants, CC_{raw} yields the best results for both test sets. The mean F_1-Scores for C and $C_{building}$ are increasing proportional with τ, but remain about 2-3% in ALS-ALS and up to 10% for certain thresholds in ALS-DIM lower than for CC_{raw} . In contrast to the height or class exclusive inputs, the combined versions using C or $C_{building}$ with HC result in higher mean F_1-Scores than the comparable raw version. Similar to the class only variants, C and $C_{building}$'s mean F_1-Scores increase proportional to τ indicating that the cells with real changes have high detection probabilities. Finally, when comparing the mean F_1-Scores between ALS-ALS and ALS-DIM for the same input variant in table 1, the values are mostly within a range of 2%, which implies, that the network can detect changes independent of the given point clouds quite sufficiently.

The mean F_1-Scores discussed so far only exhibit the quality for the matched object pairs between detection and reference. However, it is quite difficult to gain an unbiased result as the mean F_1-Scores on object-level are simultaneously over-optimistic as well as over-pessimistic. On the one hand, those results are over-optimistic when compared to the overall raster-level as they do not consider false positive detections, which do not have a matching reference at all. On the other hand, they are over-pessimistic as they weight every building equally independent of the building size. However, a single false positive cell for a small building with only a couple of cells has a much higher impact on its F_1-Score than it would have for a large building with hundreds of cells. Finally, detection results may vary between different kinds of changes. Consequently, the following sections will analyse these factors to get a deeper understanding about the detection results. As most input variants are indifferent to τ as shown in fig. 4, the remainder of this study shows the results for $\tau = 0.5$, which is a compromise between HC_{raw} and $HC_{raw} + CC_{raw}$.

3.3 Overall detection accuracy

Table 2 and 3 show the TP, FN, FP and F_1-Scores on the raster-level for $\tau = 0.5$. Overall, the reference data contains 10,080 raster cells with known changes within the test set. The difference between raster-level and object-level F_1-Scores are the amount of FPs, which are not connected to any reference object and which are considered in the former, but not the latter. Table 2 and 3 illustrate a similar trend for both test sets. When comparing the raster-level results with those on the object-level, the F_1-Scores for the raw input variants appear especially over-optimistic on the object-level, while some other input variants are over-pessimistic; e.g. the $HC + C_{building}$ input has F_1-scores on the raster-level of 91.1% and 86.3% for ALS-ALS and ALS-DIM, respectively, compared to the 80.5% and 79.8% mean F_1-Scores on object-level. This is a 13% or 8% gain on the raster-level.

Input combination	TP	FN	FP	F_1
HC_{raw}	9,243	837	129,675	12.4
HC	8,121	1,959	11,214	55.2
CC_{raw}	7,399	2,681	3,040	72.1
CC	7,792	2,288	**1,146**	81.9
C	8,152	1,928	1,557	82.4
$C_{building}$	8,006	2,074	2,235	78.8
$HC_{raw} + CC_{raw}$	7,313	2,767	2,564	73.3
$HC + CC$	7,920	2,160	1,290	82.1
$HC + C$	9,544	536	1,399	90.8
$HC + C_{building}$	**9,564**	**516**	1,364	**91.1**

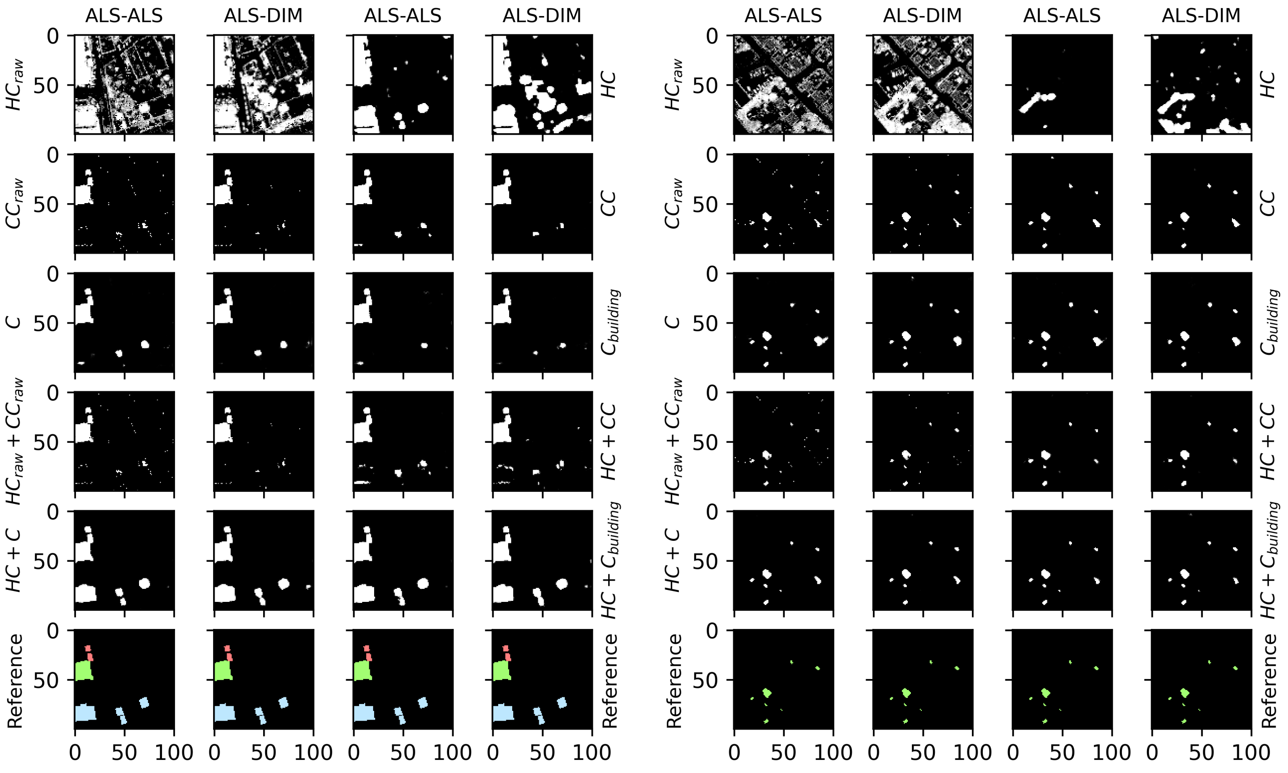
Table 2. Amount of TP, FN and FP pixels as well as F1-scores [%] calculated on raster-level for different inputs at $\tau = 0.5$ for the ALS-ALS test set. Best results are marked bold.

Input combination	TP	FN	FP	F_1
HC_{raw}	9,020	1,060	230,129	7.4
HC	8,171	1,909	97,997	14.1
CC_{raw}	7,401	2,679	2,164	75.3
CC	7,782	2,298	**1,139**	81.9
C	8,057	2,023	1,672	81.3
$C_{building}$	7,992	2,088	5,492	67.8
$HC_{raw} + CC_{raw}$	7,112	2,968	1,935	74.4
$HC + CC$	7,873	2,207	1,770	79.8
$HC + C$	9,501	579	2,233	**87.1**
$HC + C_{building}$	**9,561**	**519**	2,518	86.3

Table 3. Amount of TP, FN and FP pixels as well as F1-scores [%] calculated on raster-level for different inputs at $\tau = 0.5$ for the ALS-DIM test set. Best results are marked bold.

In average, the FP counts are lower for the ALS-ALS test set than for the ALS-DIM set. The largest difference between the two test sets can be found in the height only input variants, which also exhibit a high amount of FPs overall. Due to the special characteristics in regards to vegetation for ALS and DIM point clouds, as already discussed in section 1, the JSD algorithm detects a change and causes FPs to appear. But even in case of ALS-ALS this problem can occur, as random points on a tree at t_1 can be at completely different heights than for the other ALS point cloud at t_2. Due to using the network, HC is able to eliminate most of those FPs for both test sets when compared to their raw counterpart (see fig. 5). However, HC's FPs still remain higher than the amount of true reference cells. In both test sets, the CC input yields the lowest FP amount overall. Besides the amount of FPs, also the position is important. FPs surrounding a correct detection like a buffer may not be as harmful as separate and additional objects, which require manual control to validate as change. Examples of both FP behaviours are shown in fig. 5, where the raw input variants illustrate the random, speckle-like FPs, while most network variants demonstrate the buffer-like FPs for the majority of cells. Nevertheless, examples of those speckle-like FPs can also be found in the network-based inputs as demonstrated in fig. 1 in the north-east.

Similarly, the relation between FPs and FNs is also important. In order to reliably detect changes, a low FN count is required as it indicates a high recall rate. While the CC input variant may have the lowest FP count, they still contain 2,288 or 2,298 FNs in ALS-ALS and ALS-DIM, respectively, which are roughly 22% of all reference cells (see. table 2 and 3).



(a)

(b)

Figure 5. Predicted change detection results for two test images. While the top five rows show the detections for ALS-ALS and ALS-DIM for two examined input variants each, the bottom row displays the reference coloured by their respective change class: *new* in green, *demolished* in red and *construction* in blue.

Meanwhile, as a consequence of over-detection, the amount of FNs in HC_{raw} and HC is quite low. While the class only variants as well as most of the combined variants show a nearly balanced amounts of FP and FN, $HC + C$ and $HC + C_{building}$ achieve the lowest amounts of FNs with about 5% of the reference cells.

3.4 Influence of the area and the change type

Fig. 6 plots the mean F_1-Scores on object-level depending on their change classes and area sizes for the input $HC + C_{building}$ at $\tau = 0.5$. The area size is the ground area size of a reference object. As each raster cell covers 1m², the amount is equal to the raster cells of an object. The top row of fig. 6 illustrates the overall distribution of reference objects for each area size and change class in the test sets. The change class types are equal to those mentioned in section 3.1. The majority of changed objects have an area size between 1m² and 15m², from which the majority was newly built between 2012 and 2016. While there are examples of *new* and *demolished* in every size category, there are eleven *construction* objects mostly spread over all area sizes, but only two objects for *exchanged* in total.

The second and third row show the mean F_1-Scores for ALS-ALS and ALS-DIM, respectively. Unsurprisingly, the mean F_1-Scores increase proportional with the area size. As the influence of wrongly detected cells decrease with larger building sizes, this demonstrates the over-pessimism of fig. 4 concerning all area sizes. Regarding the area size and the change class, ALS-ALS and ALS-DIM display similar behaviours. The total mean F_1-Score for both test sets have values around and above 80% with the area range 1-5m² being the exception, where total mean F_1-Scores are closer to 65%. While ALS-ALS and ALS-DIM exhibit comparable results for the change classes *new* and *demolished* on all area sizes, the change class *exchanged* with its two objects was not successfully detected in any input type. Similarly, only ALS-DIM was able to detect the change class *construction* in every possible area category, while ALS-ALS was not able to do so in the 1-5m² and 16-20m² size category and also did quite poorly for the 6-10m² one.

4. DISCUSSIONS

As shown in section 3, the proposed method is able to detect different kinds of building changes for various area sizes if an appropriate input for the residual network is used. The results also demonstrate, that finding a common metric to determine and distinguish a good from a bad change detection result is quite difficult. While the raw input variants show superior mean F_1-Scores in fig. 4 and table 1 compared to their network counterparts, further analyses in fig. 5 and 6 as well as tables 2 and 3 reveal that using a neural network for the changed detection task improves the overall results.

Although ALS and DIM point clouds have different characteristics, their usage for change detection achieves comparable results in this study. The largest difference in quality for change detection tasks is shown in the height only variants, where a lot of FPs occur. Even the modified JSD still exhibits high false positive rates leading to the majority of the raster cells to return high values (see fig. 5). When compared to the raw baseline, the network version HC is able to suppress FPs quite effectively without any additional class information.

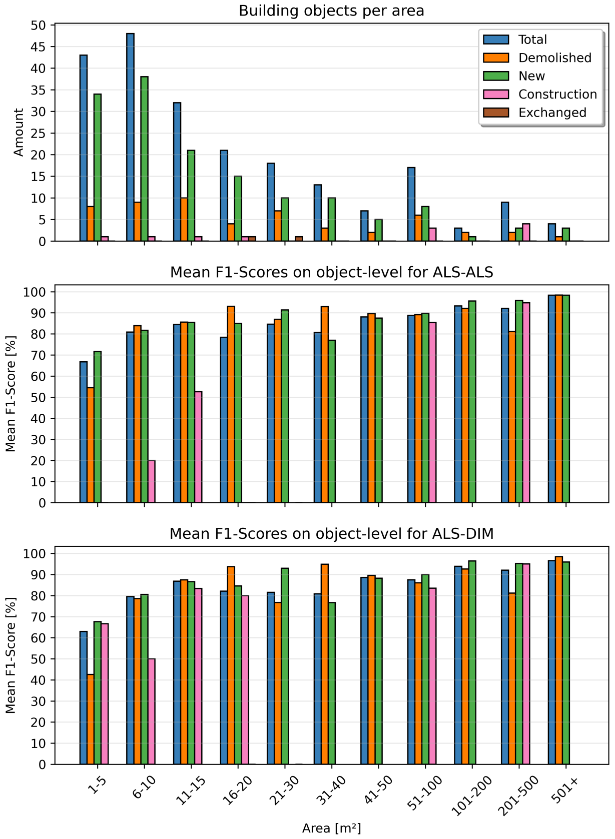


Figure 6. Change detection results using the $HC + C_{building}$ input at $\tau = 0.5$. Top row illustrates the overall amount of reference objects regarding their area size and change class. Middle and bottom row show the mean F_1-Scores for ALS-ALS and ALS-DIM with with respect to their area size and change class.

The class only variants achieve good results overall on the object- and raster-level for both test sets as discussed in section 3. However, without any height information, they fail at detecting *construction* changes, where both t_1 and t_2 contain building points. Even by adding height information like in $HC + CC$, there are still major problems in detecting *construction* changes. Only the input combinations $HC + C$ and $HC + C_{building}$ are able to detect *construction* objects successfully and reliably (see fig. 5).

The input combination $HC+C_{building}$ achieves the overall best detection results while demonstrating the most flexible solution. Only looking at matching pairs between detection and reference data, this input variant achieves mean F_1-Scores of 80.5% and 79.8% for the ALS-ALS and ALS-DIM test set as shown in table 1. Concerning all raster cells, it resulted in an overall F_1-Score of 91.1% and 86.3%, respectively, where only a small amount of buildings in the range of 5% of the test dataset were not detected (see table 2 and 3). While there are still a decent amount of FPs in the detection result, fig. 1 and 5(b) show that those often act as buffer around the correctly detected buildings instead of being separate speckle-like detections. As such, the remaining FPs should not be a problem in practical applications, where finding changed building objects is the main goal. Similarly, the network using $HC + C_{building}$ is able to detect building changes with a mean F_1-Score of about 65% for buildings at sizes up to 5m² and with a score of over 80% for any larger building sizes. It is also capable in detecting buildings belonging to different change types like *new*, *demolished* and *construction*. Finally, while $HC + C$ also exhibits equally good results, $HC + C_{building}$ only requires the knowledge of the frequency of building points, so a binary *building* and *non-building* classification, while $HC + C$ requires this fixed set of specific point classes, which may change from dataset to dataset and thus may force additional fine-tuning or even new training using another C.

There are still some open issues with the proposed change detection. First, even though the network variants do not demonstrate any single FPs, they still contain some objects, which are purely classified as FPs. An example of such an object is shown in the south-east of fig. 1 or 5(b), where some kind of elevated garage with a grass roof got falsely detected without a change between t_1 and t_2. Second, while the change class *exchanged* was expected to be a challenge due to not having any apparent differences below the set bin size φ, which was chosen proportional to the point clouds' absolute point accuracy, most input variants were not able to detect objects of that class entirely or did so very poorly. Finally, the class information of buildings was exactly projected from the reference onto the point clouds creating a nearly perfect classification accuracy for the change detection task in this study. Consequently, even CC_{raw} was able to achieve 77.7% and 81.5% mean F_1-Scores for the ALS-ALS and ALS-DIM test sets, respectively, using $\tau = 0.5$ as shown in table 1. Only due to some boundary issues and the indirect block of *construction* objects in eq. 3, this method was not able to correctly detect changes at a perfect score. However having such a high classification accuracy for buildings in the context of working with national mapping data covering large areas is far from realistic. At least in Germany, the majority of point clouds, which NMAs are using, are only classified into ground and non-ground classes, which are used to generate Digital Terrain Models. Even when the building class is provided, checking and maybe even correcting such a classification on large areas is not feasible without automation. Consequently, further studies are necessary, which examine the quality of the detection results regarding different classification accuracies.

There are some ideas for improvement. As DIM point clouds are derived products of aerial images, they also contain colour information, which was already successfully used in building change detection by Zhang et al. (2019). Similarly, newer ALS sensor systems are able to gather full waveform as well as reflectance information. All these point cloud type specific attributes are currently not included in the proposed change detection method. However, they provide some unique attributes about the roofs, which may further increase the detection results. Similarly, the network was only trained using ALS-ALS training data. Additional investigations regarding training with ALS-DIM or even some joint training data might improve the change detection results. Also, another enhancement on the JSD calculation using some kind of weighting may improve the results. Finally, instead of only detecting binary detection results, the network could be extended to detect multiple types of changes directly. This could also include changes in the vegetation or in the ground surface similar to Tran et al. (2018). Even though HC demonstrates a lot of FPs in their detections, often those detected regions can be directly connected to trees or even ground surface changes. So just adding more training examples for exactly those new categories may allow for an automatic change detection on surface changes to update Digital Terrain Models.

5. CONCLUSIONS

This study aimed at detecting building changes between two point clouds independent of their original sensor system, their point density, their point distribution or special characteristics. A residual neural network has been trained using height and class information as input. Extensive experiments provided an insight into the change detection quality, which concluded that the overall change detection accuracy is comparable independent of the used point cloud types, the observed building area sizes and the type of building changes. Using a residual neural network as feature extractor and non-linear classifier boosts the change detection results greatly. The results also demonstrate, that the proposed Jensen-Shannon distance as height detector supports the class information in detecting building changes; especially for building changes, which may only be visible in vertical direction. The input combination using the proposed height detector as well as the relative amount of building points within a raster cell at both points in time as a class detector, namely the combination $HC + C_{building}$, yields the best detection results in this study achieving overall F_1-Scores of 91.1% and 86.3% on the ALS-ALS and ALS-DIM test sets, respectively. Future work may improve on the discussed problems and possible additions.

ACKNOWLEDGEMENTS

We thank the Landesamt für Geoinformation und Landesvermessung Niedersachsen (LGLN), the Amt für Geoinformation, Vermessungs- und Katasterwesen (AFGVK) and the Landesamt für Vermessung und Geoinformation Schleswig-Holstein (LVermGeo) for providing the point cloud data and for their support of this project.

References

He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep Residual Learning for Image Recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 770–778.

Kingma, D. P., Ba, J., 2014. Adam: A method for stochastic optimization. *Published as a conference paper at ICLR 2015.*

Lague, D., Brodu, N., Leroux, J., 2013. Accurate 3D comparison of complex topography with terrestrial laser scanner: Application to the Rangitikei canyon (NZ). *ISPRS journal of photogrammetry and remote sensing*, 82, 10–26.

Lin, J., 1991. Divergence measures based on the Shannon entropy. *IEEE Transactions on Information Theory*, 37(1), 145-151.

Mandlburger, G., Wenzel, K., Spitzer, A., Haala, N., Glira, P., Pfeifer, N., 2017. Improved Topographic Models via Concurrent Airborne LiDAR and Dense Image Matching. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, IV-2/W4, 259–266.

Matikainen, L., Hyyppä, J., Ahokas, E., Markelin, L., Kaartinen, H., 2010. Automatic detection of buildings and changes in buildings for updating of maps. *Remote Sensing*, 2(5), 1217–1248.

Murakami, H., Nakagawa, K., Hasegawa, H., Shibata, T., Iwanami, E., 1999. Change detection of buildings using an airborne laser scanner. *ISPRS Journal of Photogrammetry and Remote Sensing*, 54(2-3), 148–152.

Politz, F., Sester, M., Brenner, C., 2021. Building Change Detection of Airborne Laser Scanning and Dense Image Matching Point Clouds using Height and Class Information. *AGILE: GIScience Series*, 2 (10), 1–14.

Richter, R., Kyprianidis, J., Döllner, J., 2013. Out-of-Core GPU-based Change Detection in Massive 3D Point Clouds. *Transactions in GIS*, 17(5), 724–741. 10.1111/j.1467-9671.2012.01362.x.

Rothermel, M., Wenzel, K., Fritsch, D., Haala, N., 2012. SURE: Photogrammetric Surface reconstruction from Imagery. *Proceedings of LC3D Workshop in Berlin*, 1–9.

Salehi, S. S. M., Erdogmus, D., Gholipour, A., 2017. Tversky loss function for image segmentation using 3d fully convolutional deep networks. *Machine Learning in Medical Imaging*, Springer International Publishing, 379–387.

Scott, C., Arrowsmith, J., Nissen, E., Lajoie, L., Maruyama, T., Chiba, T., 2018. The M7 2016 Kumamoto, Japan, Earthquake: 3-D Deformation Along the Fault and Within the Damage Zone Constrained From Differential Lidar Topography. *Journal of Geophysical Research: Solid Earth*, 123(7), 6138–6155.

Teo, T.-A., Shih, T.-Y., 2013. Lidar-based change detection and change-type determination in urban areas. *International Journal of Remote Sensing*, 34(3), 968–981.

Tran, T., Ressl, C., Pfeifer, N., 2018. Integrated change detection and classification in urban areas based on airborne laser scanning point clouds. *Sensors*, 18(2), 448.

Williams, J., Anders, K., Winiwarter, L., Zahs, V., Höfle, B., 2021. Multi-directional change detection between point clouds. *ISPRS Journal of Photogrammetry and Remote Sensing*, 172, 95–113.

Winiwarter, L., Anders, K., Höfle, B., 2021. M3C2-EP: Pushing the limits of 3D topographic point cloud change detection by error propagation. *ISPRS Journal of Photogrammetry and Remote Sensing*, 178, 240-258.

Zhang, Z., Vosselman, G., Gerke, M., Persello, C., Tuia, D., Yang, M., 2019. Change Detection between Digital Surface Models from Airborne Laser Scanning and Dense Image Matching using Convolutional Neural Networks. *ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences*, IV-2/W5, 453–460.

Zhou, K., Lindenbergh, R., Gorte, B., Zlatanova, S., 2020. LiDAR-guided dense matching for detecting changes and updating of buildings in Airborne LiDAR data. *ISPRS Journal of Photogrammetry and Remote Sensing*, 162, 200–213.