Change detection needs change information: improving deep 3D point cloud change detection

Iris de Gélis, Thomas Corpetti, Sébastien Lefèvre

Change detection needs change information: improving deep 3D point cloud change detection

Iris de Gélis, Thomas Corpetti, Sébastien Lefèvre

***Abstract*—Change detection is an important task to rapidly identify modified areas, in particular when multi-temporal data are concerned. In landscapes with complex geometry such as urban environment, vertical information turn out to be a very useful knowledge not only to highlight changes but also to classify them into different categories. In this paper, we focus on change segmentation directly using raw 3D point clouds (PCs), to avoid any loss of information due to rasterization processes. While deep learning has recently proved its effectiveness for this particular task by encoding the information through Siamese networks, we investigate here the idea of also using change information in early steps of deep networks. To do this, we first propose to provide the Siamese KPConv State-of-The-Art (SoTA) network with hand-crafted features and especially a change-related one. This improves the mean of Intersection over Union (IoU) over classes of change by 4.70%. Considering that the major improvement was obtained thanks to the change-related feature, we propose three new architectures to address 3D PCs change segmentation: OneConvFusion, Triplet KPConv, and Encoder Fusion SiamKPConv. All the three networks take into account change information in early steps and outperform SoTA methods. In particular, the last network, entitled Encoder Fusion SiamKPConv, overtakes SoTA with more than 5% of mean of IoU over classes of change emphasizing the value of having the network focus on change information for change detection task. The code will be made available at *https://github.com/IdeGelis/torch-points3d-SiamKPConvVariants*.**

***Index Terms*—Change detection, deep learning, 3D point clouds**

I. Introduction

In an ever evolving world, it is of prime importance to be able to sense landscape transformations. The change detection task aims at highlighting these modifications from two or several successive observations. Either in urban or geosciences domains, application are numerous. Change detection helps for example to easily updates maps [1], to identify damaged areas in case of natural disaster [2], [3], to help city managers [4], [5], to highlight coastal modifications [6]–[8], to identify glacier melting [9] or even to detect landslides [10].

Whether for urban application [11] or geosciences [12], three-dimensional (3D) data such as Point Cloud (PC) appear interesting since they provide additional vertical information, not available in two-dimensional (2D) images, and allowing to better characterize the geometry of complex landscapes. In practice, a PC is an unordered and sparse set of points represented by their 3D coordinates in a frame of reference (i.e. Cartesian coordinate system). In Earth observation purpose, 3D PCs whether acquired via photogrammetric process or Light Detection And Ranging (LiDAR) sensors (through Aerial Laser Scanning (ALS) for example) are generalizing. Because of their specific characteristics, PCs are often rasterized into 2.5D Digital Surface Model (DSM) to be easily handled using traditional image processing methods. However, the rasterization implies a significant loss of information that can be prejudicial, thus more and more studies encourage designing methods able to deal with raw 3D PCs [13], [14].

In a recent paper [14], we showed the possibilities brought by deep learning network to perform change detection and characterization into raw PCs. In particular, built upon 2D image change detection deep learning methods [15] and 3D PCs convolutions [16], the Siamese Kernel Point Convolution (KPConv) network outperforms traditional distance-based, DSM-based or machine learning-based methods on both real and simulated datasets for multiple change segmentation in urban areas. Aside from this contribution, the literature is still low in deep learning methods tackling change detection task at point level, i.e., segmentation of each point of one PC according to the different type of changes compared to the other PC. Indeed, one can cite the work of [17], [18] which apply deep models to retrieve changes, but they only focus on 2.5D DSM. On the contrary, [19] propose to process the raw 3D PCs thanks to graph convolutions [20]. However, their method is designed for change classification task, i.e., retrieving changes at scene level as proposed by the Change3D [19] or Urban 3D Change Detection Classification (Urb3DCD-Cls) [14] datasets. This task is less precise than the change segmentation one, since it allows identifying only the main changed object in a scene without precisely localizing it. Hence, the design of new methods able to enhance change segmentation in raw 3D PCs constitute open perspectives.

Considering the limited literature of deep learning methods for 3D PCs change detection, all the three studies [14], [18], [19] rely on Siamese architecture given its ability to detect changes in 2D images remote sensing field [15], [21]–[24]. Recent studies in 2D change detection have also shown that data fusion is a crucial step in change detection. Indeed, paying more attention on how to fuse information coming from the two network inputs, and on how to incorporate this fused information (i.e., change information), can improve change detection results. It has for example been demonstrated that multi-temporal fusion leads to better results when it is performed at multiple scales [15], [25]–[27]. While [15] propose

Iris de Gélis and Sébastien Lefèvre are with the Institut de Recherche en Informatique et Systèmes Aléatoires (IRISA), UMR 6074, Université Bretagne Sud, Vannes, France (e-mail: iris.de-gelis@irisa.fr,sebastien.lefevre@irisa.fr). Iris de Gélis is also with Magellium, Toulouse, France. Her work is partly funded by the CNES, Toulouse, France. Thomas Corpetti is with the Littoral – Environnement – Télédétection – Géomatique (LETG), UMR 6554, Université Rennes 2, Rennes, France. (e-mail: thomas.corpetti@cnrs.fr).

to merge information from both branches either by concatenation or differentiation of features, other studies propose more advanced fusion modules. For example, [28] propose a network based on the three results of addition, subtraction, and concatenation of features at multiple scales. The study in [29] takes a step aside from the traditional Siamese network with one input for each branch, by proposing to take in one branch the concatenation of the images, and in the other the difference, forming two sub-networks with different properties. At the output of each layer, the features of the two branches are summed and then concatenated at the corresponding scale in the decoder thanks to skip connections. In [30], authors propose to embed some fusion modules relying on multi-scale features difference aggregation and attention on concatenation of bi-temporal features. Note that according to [31], taking into account both concatenation and difference of input images is more efficient even in single-stream methods. As with multi-scale consideration, another category of methods uses attention mechanism to help the network focus on the most important features for multi-temporal information fusion [28], [32]–[34]. When going back to 3D PCs data, the data fusion from both input data is not immediate since no point-to-point correspondence exists between two PCs even in unchanged areas. To deal with this issue, Siamese KPConv include a nearest point difference of features of the encoder at multiple-scale. These feature differences are directly integrated in the decoder part. Thereby, encoders are not processing change information but only mono-date PC without any information on the other PC. We believe that processing change information earlier in the network will boost deep networks towards final change detection and categorization at point level.

Based on the consideration that a particular attention should be given to change information, i.e., both date data fusion, we propose in this paper to enhance 3D PCs change segmentation results. To do so, we i) experiment to provide additional hand-crafted features and in particular a change-related feature as input along with 3D point coordinates to the existing Siamese KPConv network; ii) design three new deep learning architectures for 3D change segmentation; iii) prove the effectiveness of the two last items on the public dataset Urban 3D Change Detection (Urb3DCD) [14].

The description of hand-crafted features and all the three new architectures are presented in Section II. Then, assessment of the methods is given in Section III and these results are further discussed in Section IV. The conclusions of the paper are given in Section V.

II. Incorporating change information in deep models

To incorporate change information in deep networks, we first propose to add as input to the current state-of-the-art methods for 3D PCs change detection some hand-crafted features, and in particular one change-related feature (Sec. II-A). Then we propose three new deep architectures (Sec. II-B) integrating change information directly in the encoder, conversely to Siamese KPConv [14].

A. Considering hand-crafted features

Even if not usual in deep learning, some studies showed that combining deep and hand-crafted features improves final results in computer vision [35] or even in remote sensing [36]. The study in [37] also showed that incorporating hand-crafted features into a deep learning framework allows improving PCs semantic segmentation. In particular, they evaluate the benefit of giving some different types of features in addition to 3D points coordinates for PointNet [38] and PointNet++ [39] 3D deep frameworks. It is shown that depending on the dataset (Mobile Laser Scanning (MLS) or ALS), PointNet basic architecture can equalize or even outperform more complex architectures such as PointNet++ or Kernel Point – Fully Convolutional Neural Network (KP-FCNN) [16] when input embeds hand-crafted features.

Therefore, we propose to study whether adding hand-crafted features, and in particular a change-related feature, in Siamese KPConv [14] deep network influences the change segmentation results. As for hand-crafted features, we used the following ones related to:

- Point normals (N_x, N_y, N_z)
- Point distribution represented by their organization in their neighborhood (named L_λ, P_λ, O_λ);
- Height information (named Z_{range}, Z_{rank} and nH);
- Change information (named $Stability$).

Information on the distribution of points contained in the neighborhood are given by the three variables: linearity L_λ, planarity P_λ and omnivariance O_λ. These variables represent respectively the likelihood of a point to belong to a linear (1D), planar (smooth surface) (2D) or volumetric (3D) neighborhood. These three attributes are common to extract information into 3D PCs. They are computed from the three eigenvalues ($\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq 0$) obtained after applying a Principal Component Analysis (PCA) to a matrix containing 3D coordinates of points contained in the neighborhood. Formulas of L_λ, P_λ and O_λ are given in Equations 1 to 3:

$$L_\lambda = \frac{\lambda_1 - \lambda_2}{\lambda_1} \tag{1}$$

$$P_\lambda = \frac{\lambda_2 - \lambda_3}{\lambda_1} \tag{2}$$

$$O_\lambda = \sqrt[3]{\lambda_1 \lambda_2 \lambda_3} \tag{3}$$

In practice, if λ_1 is large compared to λ_2 and λ_3, L_λ is near to 1. In this situation, only one eigenvalue is meaningful, i.e., only one principal axis results from the PCA and points are mainly distributed along a single axis. If λ_1 and λ_2 are large regarding λ_3, implying P_λ near to 1, points are spread in a plan defined by eigenvectors corresponding to λ_1 and λ_2. Lastly, O_λ is high if each of the three eigenvalues are of equal importance. This implies the points are scattered along the three axis, i.e., in a 3D volumetric space.

Z_{range} and Z_{rank} give information on the height by providing the maximum height (Z coordinate) difference between points in the neighborhood and the rank of the height of the considered points within the neighborhood. The normalized

height nH also completes height information by providing the difference between the height of the considered points and the local Digital Terrain Model (DTM) (rasterization of the PC at the ground level).

Lastly, the *Stability* [40] feature is given to provide a bi-temporal information on the considered point. It is defined as the ratio of the number of points in the spherical neighborhood to the number of points in the vertical cylindrical (oriented along the vertical axis) neighborhood in the other PC. Thus, in each point of the current PC, *Stability* is the ratio between the 3D (n_{3D}) and 2D (n_{2D}) neighborhood in the other PC:

$$Stability = \frac{n_{3D}}{n_{2D}} \times 100 \tag{4}$$

Notice that looking only at the number of points in the 3D neighborhood of each point of both PCs is enough to retrieve changes on isolated buildings and trees. However, in dense tree areas or when different objects are closed to each other, the 3D spherical neighborhood may still contain points coming from some other unchanged entity. Thus, taking the ratio with the 2D neighborhood is a way to take into account unchanged points and to obtain an indicator of change and the instability of the object. Thereby, the ratio will be near 100% if there is no change, and it tends to 0% if changes occur. On vegetation, we expect the *Stability* value to be lower. Thereby, most of hand-crafted features presented in [40] are used except those using LiDAR's multi-target capability, because our dataset does not contain such information. We recall that Siamese KPConv architecture takes as many input features as desired, by simply modifying the number of inputs of the first layer of the encoders.

B. New models for 3D point clouds change detection

We now explore how to learn this change information through novel deep networks. To do so, we build upon the Siamese KPConv model and propose three original architectures that emphasize change related features. All the three presented architectures are based on Kernel Point Convolution (KPConv) [16] as it has been proved efficient for change detection task in 3D PCs. Then, in order to fuse features coming from both PCs, the nearest point difference strategy is used as in Siamese KPConv:

$$(\mathcal{P}_1, \mathcal{F}_1) \ominus (\mathcal{P}_2, \mathcal{F}_2) = f_{2i} - f_{1j|j=\arg\min(\|x_{2i}-x_{1j}\|)} \tag{5}$$

Thus, for two PCs $\mathcal{P}_1$ and $\mathcal{P}_2$, with their corresponding features $\mathcal{F}_1$ and $\mathcal{F}_2$, the feature difference $\ominus$ is computed between features $f_{2i} \in \mathcal{F}_2$ of each point $x_{2i} \in \mathcal{P}_2$ of the second PC and features $f_{1j} \in \mathcal{F}_1$ of the nearest point $x_{1j} \in \mathcal{P}_1$.

A first option is to lighten the network by fusing both PCs information just after the first layer, as illustrated in Figure 1. The following layer of the encoder takes as input only the nearest point features difference (noted $\ominus$). Then, for the following layers of the encoder and the decoder, they take as input the output of the previous layer as in a classical fully convolutional network (FCN). Here, the idea is to evaluate the benefits of dealing with differences earlier in the process. This architecture is named OneConvFusion.

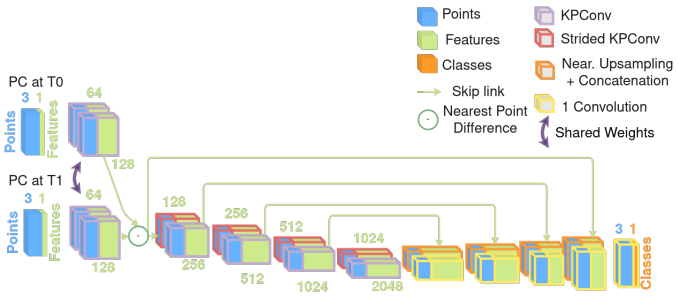


Fig. 1. OneConvFusion architecture for 3D PCs change segmentation. Links between successive layers are omitted for the sake of concision.

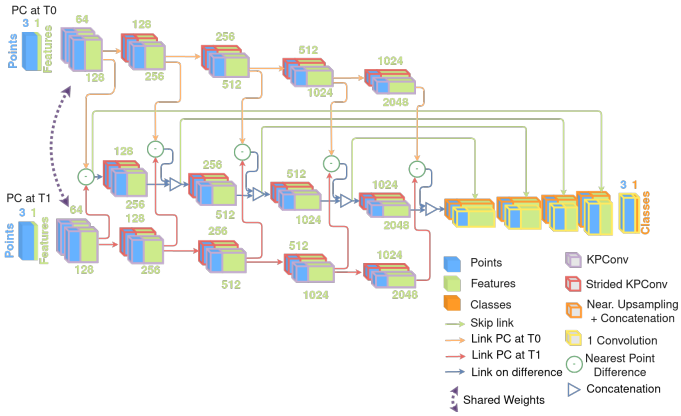


Fig. 2. Triplet KPConv architecture for 3D PCs change segmentation.

However, mono-date features of the first layer might not be sufficient for accurate change identification. Therefore, we designed the Triplet KPConv network. It contains two encoders to extract mono-date information (as in the Siamese KPConv network) and an additional encoder whose goal is to extract change related features. The "change encoder" takes as input the nearest point difference computed after the first layer of mono-date encoders. Then, the following layers of the change encoder take as input the concatenation of the output features of the previous layer and the result of the nearest point features (from mono-date encoder) difference of the corresponding scale. Thereby, multi-scale mono-date information is taken into account as well as multi-scale change information. The decoder uses features extracted by the change encoder as input. Notice that mono-date encoders can share weights or not (leading to pseudo-Triplet KPConv), as for Siamese KPConv and Pseudo-Siamese KPConv. This network is shown in Figure 2.

The third version of the architecture is designed to directly fuse mono-date and change features in a same encoder. This network is called Encoder Fusion SiamKPConv. A first encoder extracts mono-date features of the older PC using convolution layers (top of Figure 3), as in all previous architectures. Then, as illustrated in the bottom of Figure 3, the second encoder is more specific to combine output features from the newer PC and the nearest point difference of features. In particular, each layer of this second encoder takes as input the concatenation of output features of the previous layer and the difference of features from this encoder and the mono-

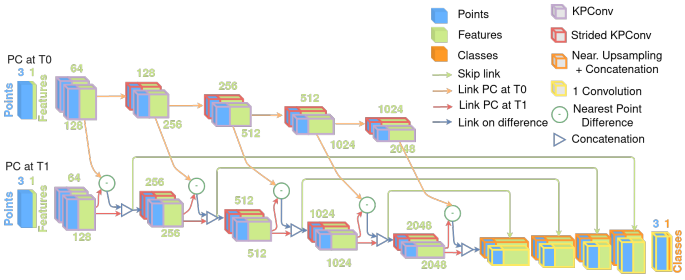


Fig. 3. Encoder Fusion SiamKPConv architecture for 3D PCs change segmentation.

date encoder of the older PC. Thereby, both mono-date and change features can be combined in convolutional layers. As with Triplet KPConv and OneConvFusion architectures, the idea is to encode the differences earlier in the process, but here they are combined with features of the second PC.

These proposed new architectures are in line with recent developments for 2D image change detection concerning the importance of data fusion [25], [29], [30] for change detection task. Indeed, by convolving change features in the encoder, we expect that the network will put more attention on changes and also better combine multi-scale change features.

In the following section, we illustrate the performances of the three proposed networks.

III. Results

In the following section, we present the experimental results of our methods on a public simulated in order to quantitatively evaluate our networks. Before describing them into detail, let us first introduce the experimental settings.

A. Dataset

In order to conduct our experiments, the public dataset Urb3DCD [41] is used. This dataset is composed of various situation of semantic changes inside cities, based on real information (related to the organization of streets, areas, ...) on which buildings, vegetation, or cars have been added. It then simulates point clouds derived from laser pulses issued from airborne with real flight plans.

Among the different versions, the second one, with a LiDAR low density (around 0.5 points/m^2), is assessed here as it contains more classes of changes, and it relies on PCs more realistic than the first version of the dataset [14].

B. Experimental settings

Concerning experimental settings, the same hyperparameters as those used by [14] for Siamese KPConv are used. In particular, some pairs of cylinders of a radius of 50 m are extracted from both PCs. The first sub-sampling rate dl_0 is set to 1 m. For training purpose, we minimize a negative log-likelihood (NLL) loss using a Stochastic Gradient Descent (SGD) with a momentum of 0.98. A batch size of 10 is used. The initial learning rate is set to 10^{-2} and scheduled to decrease exponentially. Still for the training, as change detection dataset are generally largely imbalanced, we rely on a random drawing of training cylinders as function of the class distribution as in [14]. For each training epoch, 6,000 training pairs of cylinders are seen by the network. During the validation, 3,000 pairs from the validation set are used. The loss is also weighted according to class distribution to be sure to learn also less-represented classes. Data augmentation is performed during the training: random rotation of cylinders around the vertical axis (both cylinders of a pair are rotated according to the same angle to keep coherence inside the pair) and addition of a Gaussian noise at point level.

The whole development of these architectures are implemented in PyTorch and relies on KPConv implementation available in Torch-Points3D [42]. Concerning the nearest point feature difference (Equation 5), the nearest point is determined thanks to the k-Nearest Neighbors (kNN) implementation available in PyTorch Geometric, which is graphics processing unit (GPU) compliant for faster computation.

For the hand-crafted feature extraction, the computation is made before the cylinder extraction to limit border effects. Neighborhood sizes are set at 5 m for the $Stability$. Concerning other neighborhoods, they are based on the 10 nearest neighbor points. Point normal and DTM computations are performed using Point Data Abstraction Library (PDAL)[1].

Since in change detection and categorization datasets are in general largely imbalanced (i.e., most data belong to the unchanged class despite this class not being the most interesting one), we prefer to discard the overall accuracy or precision scores that are not very indicative of method performance in such settings. We therefore select the mean of accuracy (mAcc) and the mean of Intersection over Union (IoU) over classes of changes (mIoU$_{ch}$) for reliable quantitative assessment of the different methods. IoU formula is indicated by:

$$IoU = \frac{TP}{TP + FP + FN} \tag{6}$$

where TP, TN, FP and FN respectively stand for True Positive, True Negative, False Positive and False Negative. The accuracy is given in the following formula:

$$Acc = \frac{TP + TN}{TP + TN + FP + FN} \tag{7}$$

C. Experimental results

1) Results on the addition of hand-crafted features to Siamese KPConv network: Quantitative results are given in Table I. Note that results given with zero input features corresponds to results reported in the original publication of Siamese KPConv [14]. First, we can observe that providing as input hand-crafted features in addition to point coordinates considerably improves both Siamese KPConv and Pseudo-Siamese KPConv. Then, we assessed the importance of the unique change-related hand-crafted feature ($Stability$). As visible, it seems that point distribution and height hand-crafted features have only a slight beneficial impact (+0.37% of mIoU$_{ch}$) on change segmentation results. On the opposite, the $Stability$ feature seems to have a major impact (+3.67% of mIoU$_{ch}$) on both metrics mAcc and mIoU$_{ch}$.

[1]https://pdal.io/en/2.5-maintenance/index.html, accessed on 27/02/2023.

TABLE I
COMPARISON OF OUR ARCHITECTURES WITH DIFFERENT INPUT FEATURES ON URB3DCD-V2 LOW DENSITY LIDAR DATASET. RESULTS ARE GIVEN IN %. THE TEN INPUT FEATURES ARE: N_x, N_y, N_z, L_T, P_T, O_T, Z_{range}, Z_{rank}, nH AND $Stability$.

Method	# of input features		mAcc	mIoU_{ch}
Siamese KPConv	0		91.21 ± 0.68	80.12 ± 0.02
	10		**93.65** ± 0.16	**84.82** ± 0.58
	9	w/o $Stability$	91.44 ± 0.47	80.49 ± 0.64
	1	$Stability$ only	92.92 ± 0.47	83.80 ± 0.89

TABLE II
GENERAL RESULTS IN % OF THE THREE SIAMESE KPCONV EVOLUTIONS ON URB3DCD-V2 LOW DENSITY LIDAR DATASET.

Method	mAcc (%)	mIoU_{ch} (%)
Siamese KPConv [14]	91.21 ± 0.68	80.12 ± 0.02
Siamese KPConv (+10 input features)	93.65 ± 0.16	84.82 ± 0.58
OneConvFusion	92.62 ± 1.10	81.74 ± 1.45
Triplet KPConv	92.94 ± 0.53	84.08 ± 1.20
Encoder Fusion SiamKPConv	**94.23** ± 0.88	**85.19** ± 0.24

More specifically, when looking at the per class gain in IoU, the $Stability$ feature on its own principally helps for 'new building', 'demolition' and 'missing vegetation' classes (see Figure 4).

2) Results on the evolution of Siamese KPConv: Quantitative results of the evaluation of the three architectures are presented in Tables II and III. It is worth noting that each of the three architectures outperforms Siamese KPConv network. In particular, the best architecture is Encoder Fusion SiamKPConv nearly followed by Triplet KPConv, while OneConvFusion is only slightly better (1.5% of mIoU_{ch}) than Siamese KPConv. When looking at per class results (Table III and Figure 5), Encoder Fusion SiamKPConv network provides a significant improvement for 'new building', 'demolition', 'new vegetation', 'missing vegetation' and 'vegetation growth' classes. Qualitative results are shown in Figures 6 and 7. As can be seen, the three architectures provide very similar results to the ground truth. In Figure 7, each of the three Siamese KPConv evolution show results more accurate than Siamese KPConv in the new building facades. These facades are particularly hard to correctly detect, because in the first PC (Figure 7a), the neighbor facade was not visible. Thereby, identifying the new facade in the class 'new building' while neighboring facades are unchanged is not obvious. In this situation, the network should understand that if the roof is new, the facade is probably new also. In the same way, if the roof has not changed, the facade also should be identical. Another difference with Siamese KPConv results is visible in Figure 6, where a part of the church roof is identified as new vegetation for Siamese KPConv while not for the other architectures. The misclassification is probably due to the shape of the dome roof that looks like a tree in simulated data. Indeed, even if tree models are not totally spherical (in particular the Arbraro software [43] was used to obtain OBJ models of trees, see [14]), LiDAR simulation on these models render a quite spherical object with only a few points inside the foliage of the tree unlike real LiDAR acquisition. Therefore, aside from the shape, the main way to distinguish between the vegetation and the dome is that trees are generally on the ground. These examples, highlight the fact that the network should be able to understand the PC at multiple scales and predict changes with regard to surrounding objects.

IV. DISCUSSION

On the importance of learning change information

An immediate observation from our experiments is that the addition of hand-crafted features related to both input data as input to Siamese KPConv network does not bring any significant change on the results (see Table II, third line – addition of features– compared to the first one).

However, even though looking at the results presented in [14], Siamese KPConv architecture is able to recover the change on its own, it seems that giving a hand-crafted feature related to the change specifically as input helps the network to focus on the change with a significant improvement (see Table II, fourth line –addition of a change feature– compared to the first one).

This strengthens the fact that encoding change information is important. On this basis, the proposed evolutions of Siamese KPConv show the relevance of applying convolution also on the nearest point features difference at multiple scale to obtain change-related features, as illustrated in Table III (three last lines, without the introduction of change features but by encoding them directly). Lower results of OneConvFusion network exhibit that it is important to keep multi-scale mono-date features in the architecture (note that it is in line with deep learning for change detection literature in 2D [15], [25]–[27]). Then, the fact that Encoder Fusion SiamKPConv provides better results than the Triplet network shows that combining both mono-date semantic features and change features as input to convolutional layers can extract useful discriminative features for the change segmentation task. Both Triplet KPConv and Encoder Fusion SiamKPConv are closer to results on the benefit of adding hand-crafted features as input to Siamese KPConv network. More specifically, Encoder fusion SiamKPConv gets better results than the Siamese KPConv with the 10 hand-crafted features.

Finally, we tried to add hand-crafted features as input to Encoder Fusion SiamKPConv network, results are only very slightly improved (less than 1% of mIoU_{ch},). This shows that an architecture more specifically designed for change detection is capable of extracting discriminative features on its own. This is especially true for change-related features such as the $Stability$ which is no longer required in Encoder Fusion SiamKPConv architecture.

V. CONCLUSION

In this paper, we proposed to enhance change detection into raw 3D PCs using deep networks. To do so, we suggest introducing change information earlier in the network in order to better detect and categorized changes into 3D PCs. A first proposition to enhance the existing method is to provide some hand-crafted features as input along with 3D point coordinates. In particular, we demonstrated that a single addition of a change-related feature input to Siamese KPConv existing

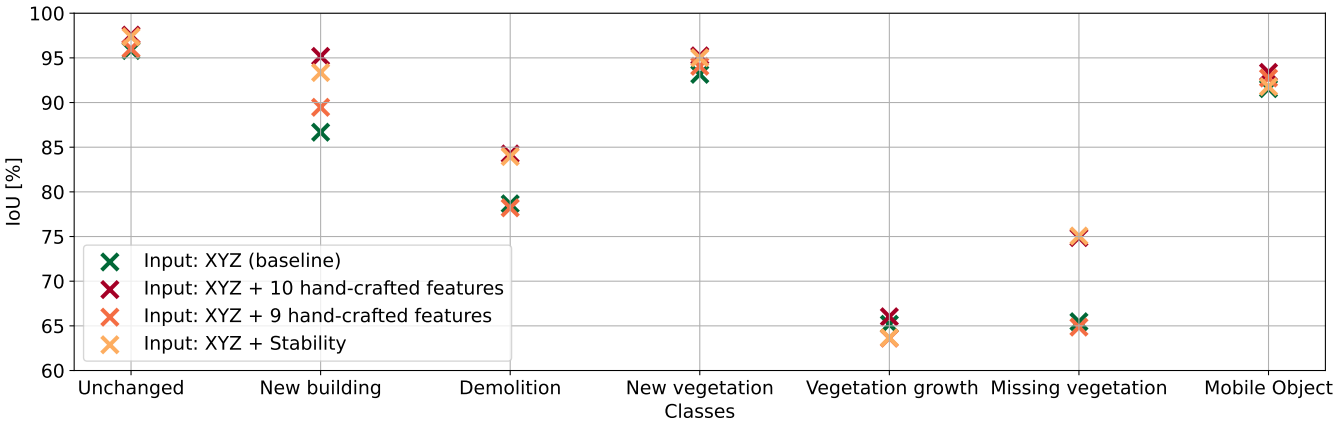


Fig. 4. **Influence on per class IoU of adding hand-crafted features along with 3D point coordinates as input to Siamese KPConv.** For classes 'new building', 'demolition' and 'missing vegetation', the high disparity in IoU shows that adding hand-crafted features to the input has a greater influence than on classes where results are grouped around a same value.

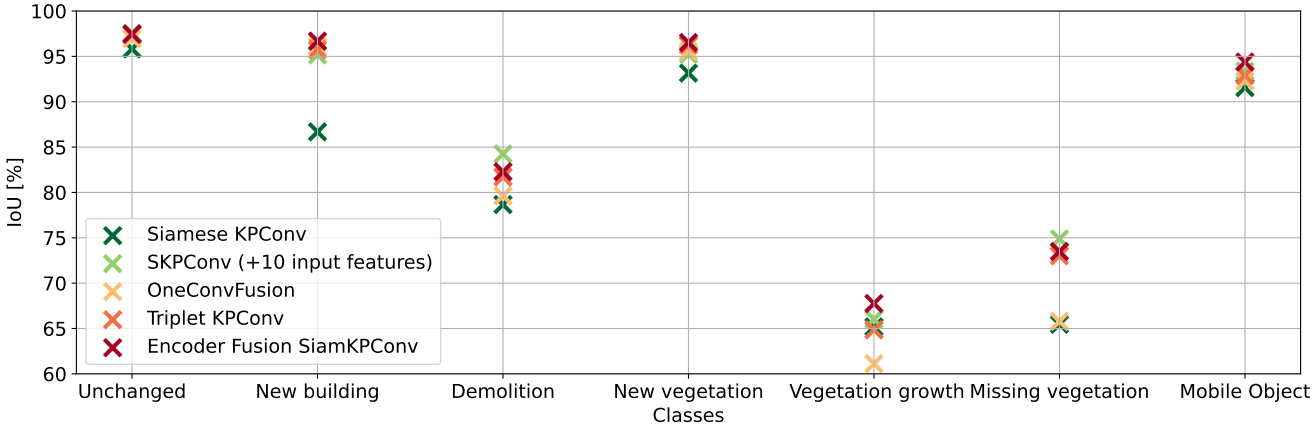


Fig. 5. **Influence on per class IoU of the three Siamese KPConv evolutions**, namely OneConvFusion, Triplet KPConv and Encoder Fusion SiamKPConv. For comparison purpose, results of Siamese KPConv with 10 hand-crafted input features are also shown.

method allows enhancing of about 3.70% of mean of IoU over classes change. Then, we propose three new architecture for change segmentation into raw 3D PCs that encode also change information conversely to the current state-of-the-art that was only incorporating change information in the decoder step. All these three architectures out-perform the current state-of-the-art methods up to 5.07% of mean of IoU over classes of change. Thereby, in this paper, we showed the importance of encoding change information.

Acknowledgment

This research was funded by Magellium and the CNES, Toulouse, with access to the HPC resources of IDRIS under the allocation 2022-AD011011754R2 made by GENCI.

References

[1] N. Champion, D. Boldo, M. Pierrot-Deseilligny, and G. Stamon, "2d building change detection from high resolution satelliteimagery: A two-step hierarchical method based on 3d invariant primitives," *Pattern Recognition Letters*, vol. 31, no. 10, pp. 1138–1147, 2010.

[2] J. Sublime and E. Kalinicheva, "Automatic post-disaster damage mapping using deep-learning techniques for change detection: Case study of the tohoku tsunami," *Remote Sensing*, vol. 11, no. 9, p. 1123, 2019.

[3] V. Zahs, K. Anders, J. Kohns, A. Stark, and B. Höfle, "Classification of structural building damage grades from multi-temporal photogrammetric point clouds using a machine learning model trained on virtual laser scanning data," *arXiv preprint arXiv:2302.12591*, 2023.

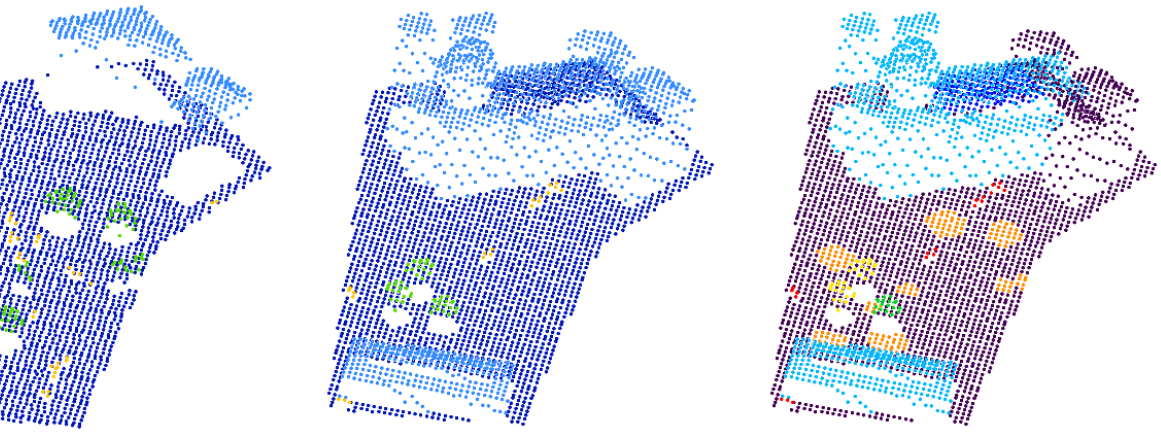
[4] I. Sandric, B. Mihai, I. Savulescu, B. Suditu, and Z. Chitu, "Change detection analysis for urban development in bucharest-romania, using high resolution satellite imagery," in *2007 Urban Remote Sensing Joint Event*. IEEE, 2007, pp. 1–8.

[5] J. Feranec, G. Hazeu, S. Christensen, and G. Jaffrain, "Corine land cover change detection in europe (case studies of the netherlands and slovakia)," *Land use policy*, vol. 24, no. 1, pp. 234–247, 2007.

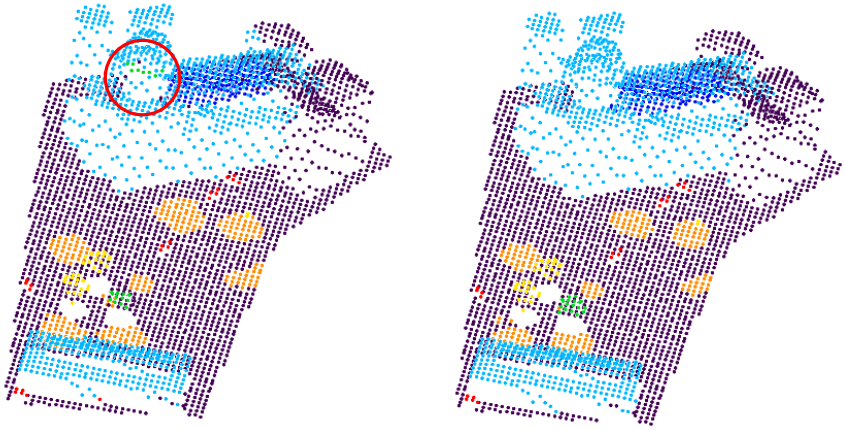
[6] P. Letortu, S. Costa, O. Maquaire, C. Delacourt, E. Augereau, R. Davidson, S. Suanez, and J. Nabucet, "Retreat rates, modalities and agents responsible for erosion along the coastal chalk cliffs of Upper Normandy: The contribution of terrestrial laser scanning," *Geomorphology*, vol. 245, pp. 3–14, Sep. 2015.

[7] A. R. Enríquez, M. Marcos, A. Falqués, and D. Roelvink, "Assessing beach and dune erosion and vulnerability under sea level rise: a case study in the mediterranean sea," *Frontiers in Marine Science*, vol. 6, p. 4, 2019.

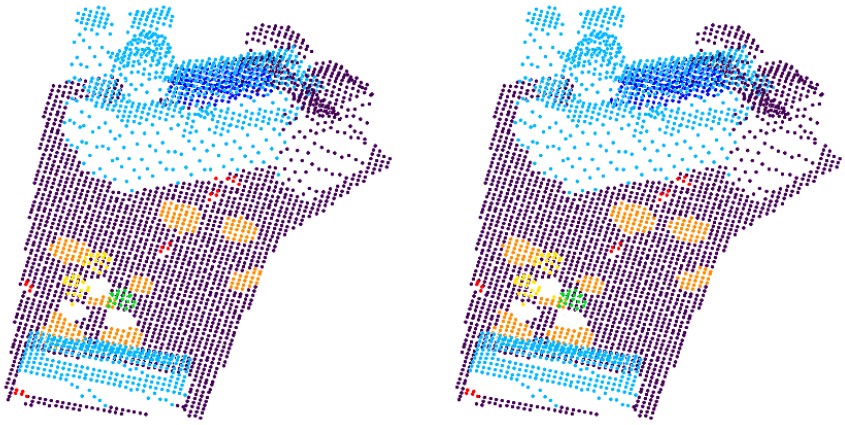
[8] I. de Gélis, Z. Bessin, P. Letortu, M. Jaud, C. Delacourt, S. Costa, O. Maquaire, R. Davidson, T. Corpetti, and S. Lefèvre, "Cliff change detection using siamese kpconv deep network on 3d point clouds," *ISPRS*



(**a**) PC 1 (**b**) PC 2 (**c**) GT

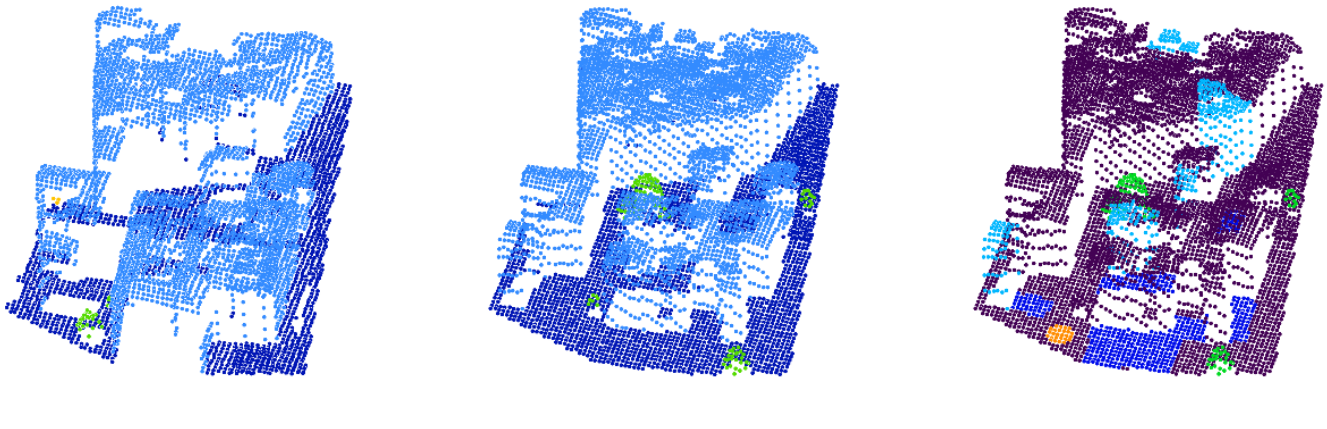


(**d**) Siamese KPConv [14] (**e**) OneConvFusion

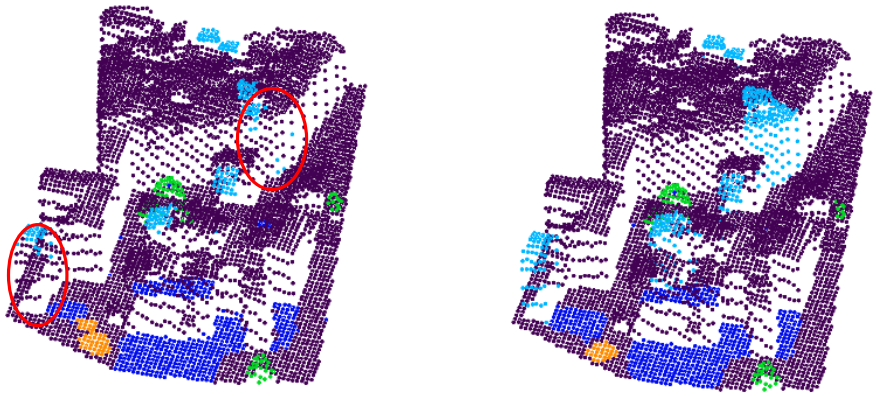


(**f**) Triplet KPConv (**g**) Encoder Fusion SiamKPConv

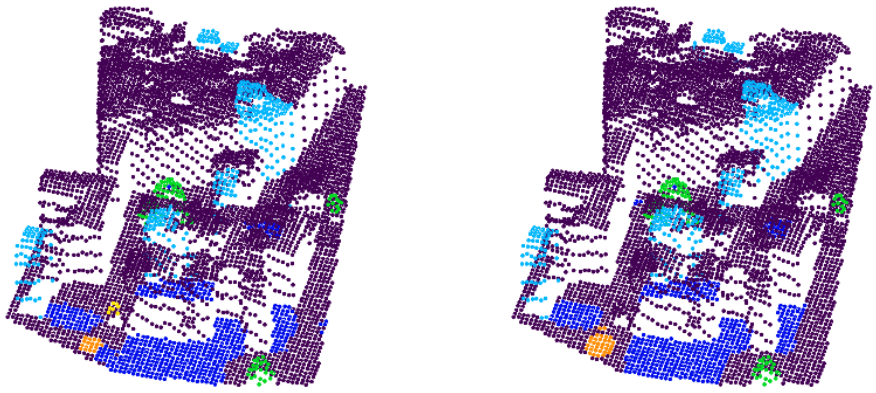
Fig. 6. **Visual change detection results on Urb3DCD-V2 low density LiDAR sub-dataset:** (**a**-**b**) the two input point clouds; (**c**) ground truth (GT): simulated changes; (**d**) Siamese KPConv results; (**e**) OneConvFusion results; (**f**) Triplet KPConv results; (**g**) Encoder Fusion SiamKPConv results. Region of interest specifically discussed in the text is highlighted with an ellipse.



(**a**) PC 1 (**b**) PC 2 (**c**) GT



(**d**) Siamese KPConv [14] (**e**) OneConvFusion



(**f**) Triplet KPConv (**g**) Encoder Fusion SiamKPConv

Fig. 7. **Visual change detection results on Urb3DCD-V2 low density LiDAR sub-dataset in an area containing occlusions:** (**a**-**b**) the two input point clouds; (**c**) ground truth (GT): simulated changes; (**d**) Siamese KPConv results; (**e**) OneConvFusion results; (**f**) Triplet KPConv results; (**g**) Encoder Fusion SiamKPConv results. Regions of interest specifically discussed in the text are highlighted with ellipses.

TABLE III
PER-CLASS IOU SCORES OF THE THREE SIAMESE KPCONV EVOLUTIONS ON URB3DCD-V2 LOW DENSITY LIDAR DATASET. RESULTS ARE GIVEN IN %. VEG. STANDS FOR VEGETATION; INPUT FEAT. FOR INPUT FEATURES; SKPCONV FOR SIAMESE KPCONV.

Method	Per class IoU (%)						
	Unchanged	New building	Demolition	New veg.	Veg. growth	Missing veg.	Mobile Object
Siamese KPConv [14]	95.82 ± 0.48	86.67 ± 0.47	78.66 ± 0.47	93.16 ± 0.27	65.18 ± 1.37	65.46 ± 0.93	91.55 ± 0.60
SKPConv (+10 input feat.)	**97.55** ± 0.11	95.17 ± 0.21	**84.25** ± 0.59	95.23 ± 0.21	66.02 ± 1.33	**74.88** ± 1.03	93.38 ± 0.74
OneConvFusion	96.95 ± 0.34	96.06 ± 0.27	79.63 ± 1.48	95.53 ± 0.77	61.12 ± 2.13	65.79 ± 2.61	92.89 ± 1.95
Triplet KPConv	97.41 ± 0.24	95.73 ± 0.67	81.71 ± 1.47	96.24 ± 0.37	64.85 ± 1.46	73.02 ± 1.18	92.90 ± 2.47
Encoder Fusion SKPConv	97.47 ± 0.04	**96.68** ± 0.30	82.29 ± 0.16	**96.52** ± 0.03	**67.76** ± 1.51	73.50 ± 0.81	**94.37** ± 0.54

Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. V-3-2022, pp. 649–656, 2022.

[9] R. Hock, "Glacier melt: a review of processes and their modelling," *Progress in physical geography*, vol. 29, no. 3, pp. 362–391, 2005.

[10] B. D. Malamud, D. L. Turcotte, F. Guzzetti, and P. Reichenbach, "Landslide inventories and their statistical properties," *Earth Surface Processes and Landforms*, vol. 29, no. 6, pp. 687–711, 2004.

[11] U. Stilla and Y. Xu, "Change detection of urban objects using 3d point clouds: A review," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 197, pp. 228–255, 2023.

[12] R. Qin, J. Tian, and P. Reinartz, "3D change detection – approaches and applications," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 122, pp. 41–56, 2016.

[13] T. G. Bernard, D. Lague, and P. Steer, "Beyond 2d landslide inventories and their rollover: synoptic 3d inventories and volume from repeat lidar data," *Earth Surface Dynamics*, vol. 9, no. 4, pp. 1013–1044, 2021.

[14] I. de Gélis, S. Lefèvre, and T. Corpetti, "Siamese kpconv: 3d multiple change detection from raw point clouds using deep learning," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 197, pp. 274–291, 2023.

[15] R. C. Daudt, B. Le Saux, and A. Boulch, "Fully convolutional siamese networks for change detection," in *2018 25th IEEE International Conference on Image Processing (ICIP)*. IEEE, 2018, pp. 4063–4067.

[16] H. Thomas, C. R. Qi, J.-E. Deschaud, B. Marcotegui, F. Goulette, and L. J. Guibas, "Kpconv: Flexible and deformable convolution for point clouds," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 6411–6420.

[17] Z. Zhang, G. Vosselman, M. Gerke, D. Tuia, and M. Y. Yang, "Change detection between multimodal remote sensing data using siamese cnn," *arXiv preprint arXiv:1807.09562*, 2018.

[18] Z. Zhang, G. Vosselman, M. Gerke, C. Persello, D. Tuia, and M. Yang, "Detecting building changes between airborne laser scanning and photogrammetric data," *Remote sensing*, vol. 11, no. 20, p. 2417, 2019.

[19] T. Ku, S. Galanakis, B. Boom, R. C. Veltkamp, D. Bangera, S. Gangisetty, N. Stagakis, G. Arvanitis, and K. Moustakas, "Shrec 2021: 3d point cloud change detection for street scenes," *Computers & Graphics*, vol. 99, pp. 192–200, 2021.

[20] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic graph cnn for learning on point clouds," *ACM Transactions On Graphics (TOG)*, vol. 38, no. 5, pp. 1–12, 2019.

[21] Y. Zhan, K. Fu, M. Yan, X. Sun, H. Wang, and X. Qiu, "Change detection based on deep siamese convolutional network for optical aerial images," *IEEE Geoscience and Remote Sensing Letters*, vol. 14, no. 10, pp. 1845–1849, 2017.

[22] S. Lefèvre, D. Tuia, J. D. Wegner, T. Produit, and A. S. Nassar, "Toward seamless multiview scene analysis from satellite to street level," *Proceedings of the IEEE*, vol. 105, no. 10, pp. 1884–1899, 2017.

[23] H. He, M. Chen, T. Chen, and D. Li, "Matching of remote sensing images with complex background variations via siamese convolutional neural network," *Remote Sensing*, vol. 10, no. 2, p. 355, 2018.

[24] W. Shi, M. Zhang, R. Zhang, S. Chen, and Z. Zhan, "Change detection based on artificial intelligence: State-of-the-art and challenges," *Remote Sensing*, vol. 12, no. 10, p. 1688, 2020.

[25] H. Chen, C. Wu, B. Du, and L. Zhang, "Deep siamese multi-scale convolutional network for change detection in multi-temporal vhr images," in *2019 10th International Workshop on the Analysis of Multitemporal Remote Sensing Images (MultiTemp)*. IEEE, 2019, pp. 1–4.

[26] M. Zhang and W. Shi, "A feature difference convolutional neural network-based change detection method," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 58, no. 10, pp. 7232–7246, 2020.

[27] X. Zheng, D. Guan, B. Li, Z. Chen, and L. Pan, "Global and local graph-based difference image enhancement for change detection," *Remote Sensing*, vol. 15, no. 5, p. 1194, 2023.

[28] L. Song, M. Xia, J. Jin, M. Qian, and Y. Zhang, "Suacdnet: Attentional change detection network based on siamese u-shaped structure," *International Journal of Applied Earth Observation and Geoinformation*, vol. 105, p. 102597, 2021.

[29] K. Jiang, W. Zhang, J. Liu, F. Liu, and L. Xiao, "Joint variation learning of fusion and difference features for change detection in remote sensing images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–18, 2022.

[30] H. Yin, L. Weng, Y. Li, M. Xia, K. Hu, H. Lin, and M. Qian, "Attention-guided siamese networks for change detection in high resolution remote sensing images," *International Journal of Applied Earth Observation and Geoinformation*, vol. 117, p. 103206, 2023.

[31] X. Peng, R. Zhong, Z. Li, and Q. Li, "Optical remote sensing image change detection based on attention mechanism and image difference," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59, no. 9, pp. 7296–7307, 2020.

[32] H. Jiang, X. Hu, K. Li, J. Zhang, J. Gong, and M. Zhang, "Pga-siamnet: Pyramid feature-based attention-guided siamese network for remote sensing orthoimagery building change detection," *Remote Sensing*, vol. 12, no. 3, p. 484, 2020.

[33] P. Chen, L. Guo, X. Zhang, K. Qin, W. Ma, and L. Jiao, "Attention-guided siamese fusion network for change detection of remote sensing images," *Remote Sensing*, vol. 13, no. 22, p. 4597, 2021.

[34] J. Chen, J. Fan, M. Zhang, Y. Zhou, and C. Shen, "Msf-net: A multiscale supervised fusion network for building change detection in high-resolution remote sensing images," *IEEE Access*, vol. 10, pp. 30 925–30 938, 2022.

[35] L. Nanni, S. Ghidoni, and S. Brahnam, "Handcrafted vs. non-handcrafted features for computer vision classification," *Pattern Recognition*, vol. 71, pp. 158–172, 2017.

[36] R. Nijhawan, J. Das, and B. Raman, "A hybrid of deep learning and hand-crafted features based approach for snow cover mapping," *International journal of remote sensing*, vol. 40, no. 2, pp. 759–773, 2019.

[37] P.-H. Hsu and Z.-Y. Zhuang, "Incorporating handcrafted features into deep learning for point cloud classification," *Remote Sensing*, vol. 12, no. 22, p. 3713, 2020.

[38] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 652–660.

[39] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," *Advances in neural information processing systems*, vol. 30, 2017.

[40] T. Tran, C. Ressl, and N. Pfeifer, "Integrated change detection and classification in urban areas based on airborne laser scanning point clouds," *Sensors*, vol. 18, no. 2, p. 448, 2018.

[41] I. de Gélis, S. Lefèvre, and T. Corpetti, "Change detection in urban point clouds: An experimental comparison with simulated 3d datasets," *Remote Sensing*, vol. 13, no. 13, 2021.

[42] T. Chaton, N. Chaulet, S. Horache, and L. Landrieu, "Torch-points3d: A modular multi-task frameworkfor reproducible deep learning on 3d point clouds," *arXiv preprint arXiv:2010.04642*, 2020.

[43] W. Diestel, "Arbaro—tree generation for povray," 2003.