3D point clouds change detection:

From supervised to unsupervised deep learning methods

Iris de Gélis[1,2]

Webinare "Autour de la 3D" MAGIS/IG-RV – 01/12/2022

[1]Magellium, F-31000, Toulouse, France; iris.de-gelis@irisa.fr
[2]IRISA UMR 6074, Université Bretagne Sud, F-56000 Vannes, France

This work is part of a PhD funded by the CNES and Magellium

Change detection in urban areas

Google Earth Timelapse (Google, Landsat, Copernicus)

Wikipedia

Change detection in geosciences

NEWS

Cliff collapse on Greece's 'shipwreck beach' injures tourists

NEWS

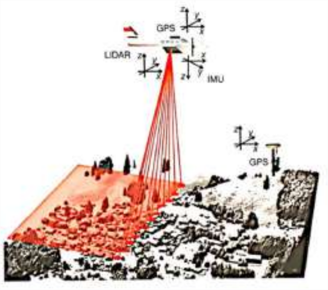
Thorpeness cliff erosion: Warning to avoid part of beach

- → Coastal cliffs ∼ **52% of the global shoreline** *(Young and Carilli 2019)*
- → Cliff erosion likely to **increase** with sea level rise *(Allan et al. 2021)*
- → **Endangering** population and infrastructure

3D point clouds

Data description

Laser Scanning: LiDAR



Lebègue et al. 2020

Photogrammetry

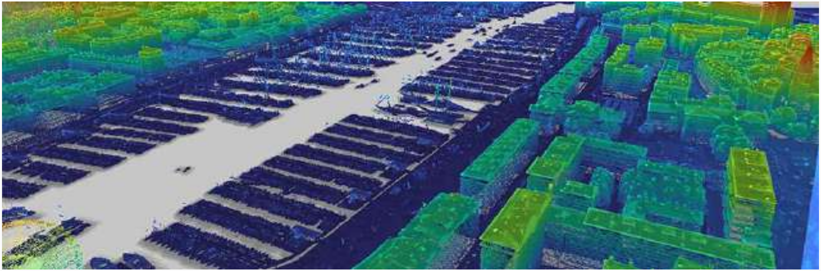
J. Vallet

Terrestrial, aerial or satellites acquisitions.

More and more 3D data available...

Lebègue et al. 2020

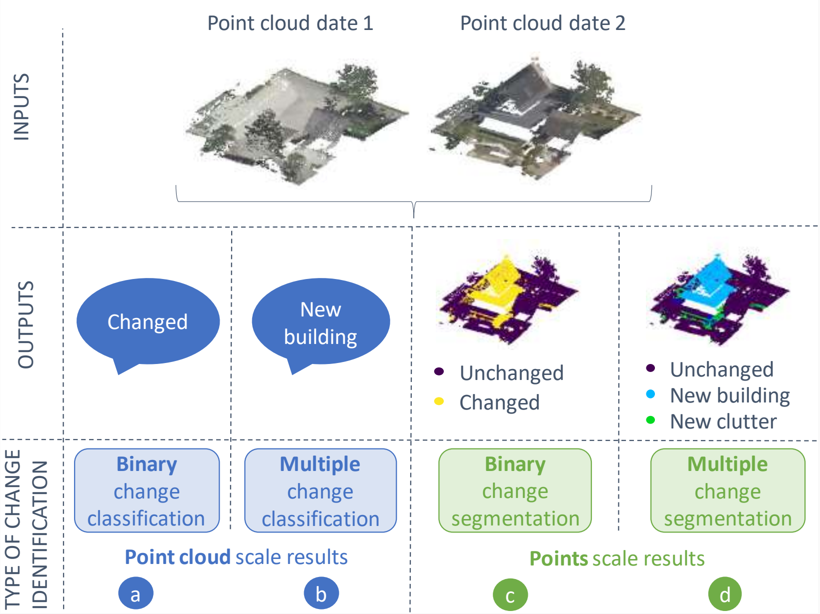
https://www.intelligence-airbusds.com/



https://www.ign.fr/institut/lidar-hd-vers-une-nouvelle-cartographie-3d-du-territoire

3D change detection

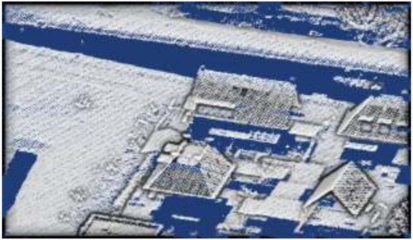
Change detection into 3D point clouds?



3D point clouds for change detection

Date 1

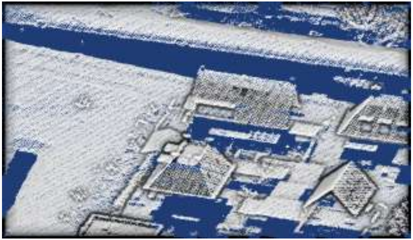
Date 2



3D point clouds for change detection

Date 1

Date 2



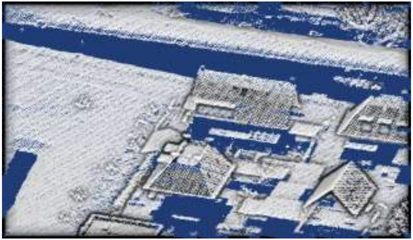
- Sparse
- Unordered

Unlike 2D images:

3D point clouds for change detection

Date 1

Date 2



- Sparse
- Unordered

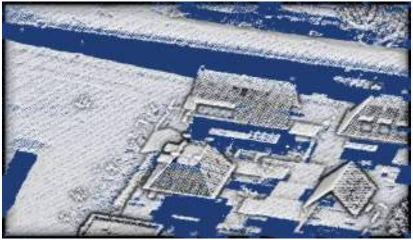
$\Big\}$ Raw PCs $\neq$ matrices

Unlike 2D images:

3D point clouds for change detection

Date 1

Date 2



$\left.\begin{array}{l}\bullet\ \text{Sparse}\\ \bullet\ \text{Unordered}\end{array}\right\}$ Raw PCs $\neq$ matrices

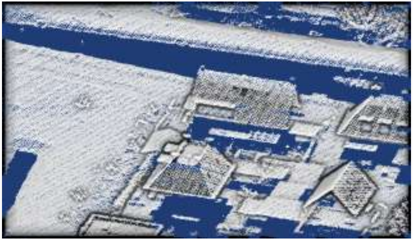
Unlike 2D images:

- No direct comparison possible

3D point clouds for change detection

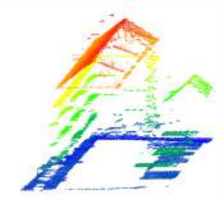
Date 1

Date 2



- Sparse
- Unordered

$\Big\}$ Raw PCs $\neq$ matrices

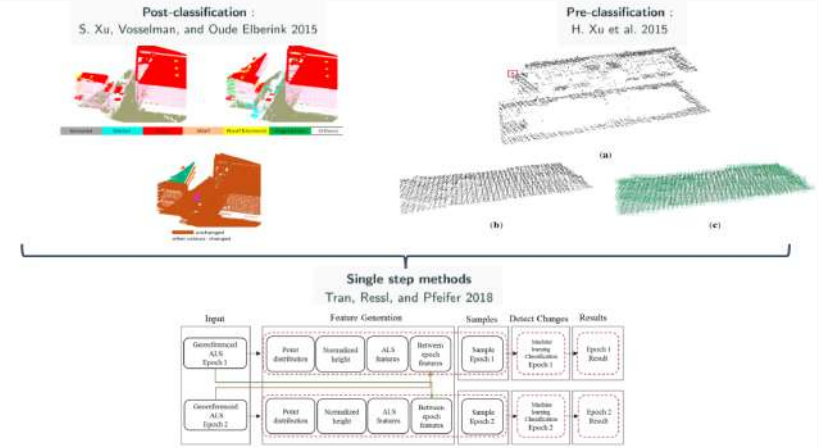


Unlike 2D images:

- No direct comparison possible
- Different hidden parts in each point cloud

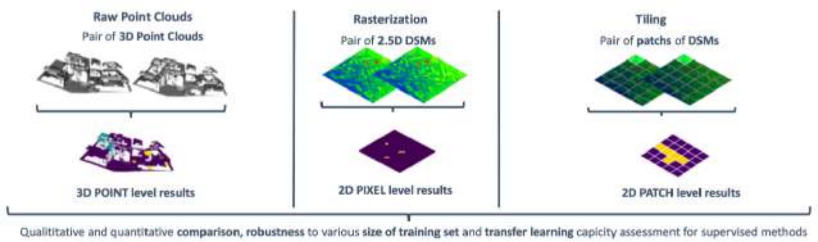
Related works

Related Works – Using raw 3D point clouds



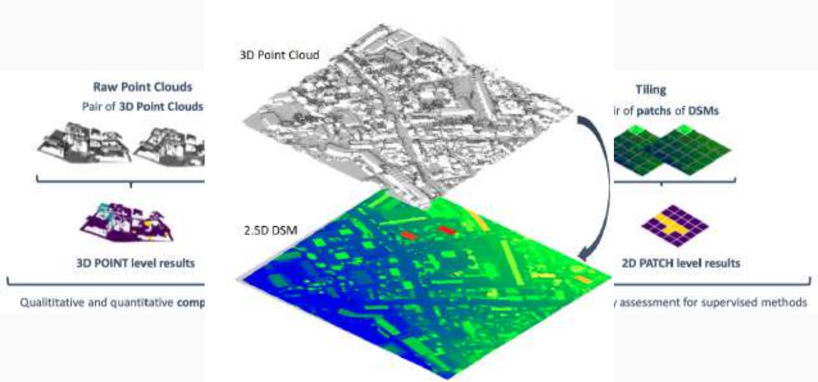
⇒ **No deep learning** based method for **3D point clouds change detection and categorization**

Benchmark of methods for change detection



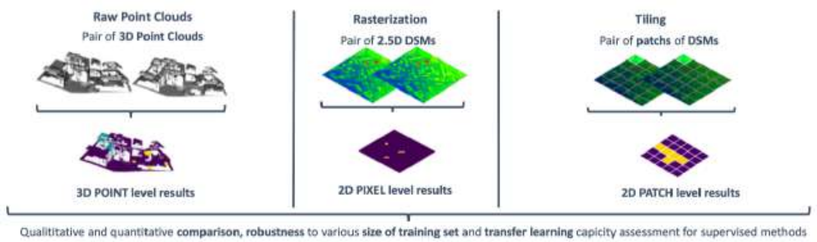
Iris de Gélis et al. (2021b). "Change Detection in Urban Point Clouds: An Experimental Comparison with Simulated 3D Datasets". In: *Remote Sensing* 13.13, p. 2629

Benchmark of methods for change detection



Iris de Gélis et al. (2021b). "Change Detection in Urban Point Clouds: An Experimental Comparison with Simulated 3D Datasets". In: *Remote Sensing* 13.13, p. 2629

Benchmark of methods for change detection



⇒ Majority of methods only focus on **DSMs** : loss of information

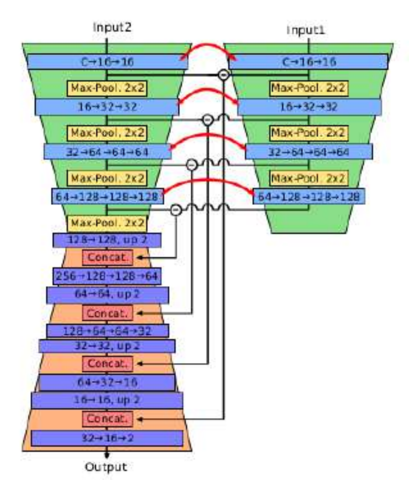
⇒ **Deep learning** method on produce a binary per 2D patch results

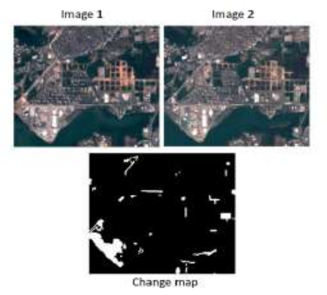
⇒ Existing **traditional methods** scores can be **largely improved**

Iris de Gélis et al. (2021b). "Change Detection in Urban Point Clouds: An Experimental Comparison with Simulated 3D Datasets". In: *Remote Sensing* 13.13, p. 2629

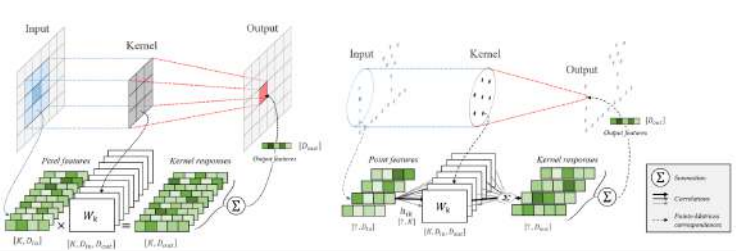
Our proposition using deep learning

Fully Convolutional Siamese Network with difference





2D Convolution **3D Kernel Point Convolution**



Thomas et al. 2019

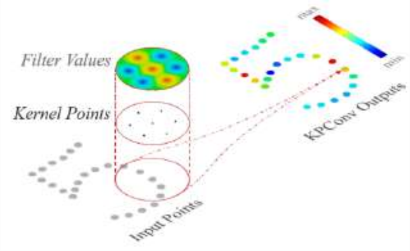
Convolution by a kernel function g at a point $x \in \mathbb{R}^3$:

$$(\mathcal{F} * g)(x) = \sum_{x_i \in \mathcal{N}_i} g(\underbrace{x_i - x}_{y_i}) f_i$$

- x_i points from $\mathcal{P} \in \mathbb{R}^{N \times 3}$
- f_i corresponding features from $\mathcal{F} \in \mathbb{R}^{N \times D}$
- $\mathcal{N}_i = \{x_i \in \mathcal{P} \mid \|x_i - x\| \leq R\}$ with $R \in \mathbb{R}$
- g: kernel function defined inside $\mathcal{B}_R^3 = \{y \in \mathbb{R}^3 \mid \|y\| \leq R\}$

3D point clouds Semantic Segmentation : KPConv

2D change detection | 3D segmentation | Our contribution



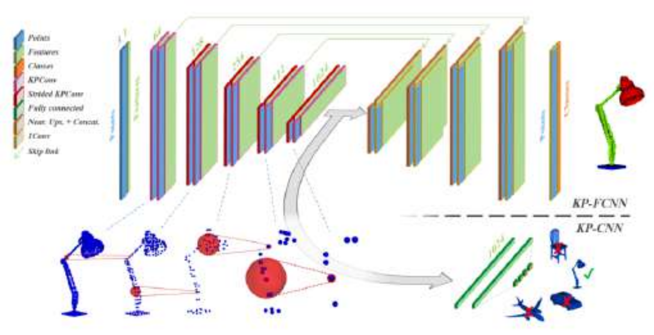
Thomas et al. 2019

Kernel function g applies weights to different areas :

$$g(y_i) = \sum_{k<K} h(y_i, \widetilde{x}_k) W_k$$

- $\widetilde{x}_k$: Kernel Point (k<K)
- W_k : Weight matrices $\{W_k | k < K\} \subset \mathbb{R}^{D_{in} \times D_{out}}$
- h : Correlation function: $h(y_i, \widetilde{x}_k) = max(0, 1 - \frac{\|y_i - \widetilde{x}_k\|}{\sigma})$
- σ : influence distance of kernel points

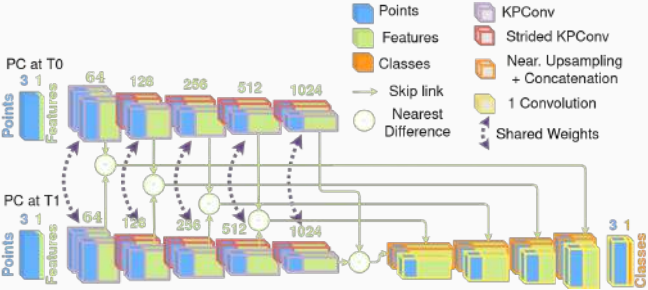
3D point clouds Semantic Segmentation : KPConv



Thomas et al. 2019

⇒ Network that looks like traditional 2D images CNN

Siamese Kernel Point Convolution Network



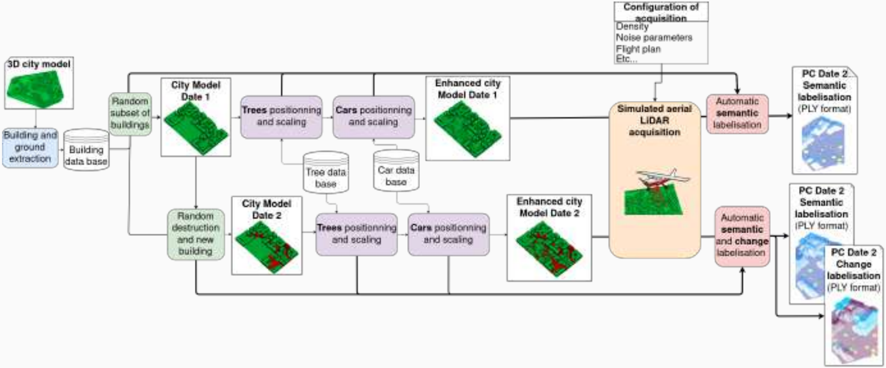
de Gélis et al. 2021a

→ Nearest point feature difference: $\ominus$ between the older PC $\mathcal{P}_0$ and its corresponding features in $\mathcal{F}_0$ and the newer PC $\mathcal{P}_1$ and its features $\mathcal{F}_1$

$$(\mathcal{P}_0, \mathcal{F}_0) \ominus (\mathcal{P}_1, \mathcal{F}_1) = f_{1i} - f_{0j|j=\arg\min(\|x_{\mathbf{1}i} - x_{\mathbf{0}j}\|)}$$

Application to urban environment

Urb3DCD - Simulator for 3D PCs in urban environment



→ **Automatic annotation** of PCs

→ Configuration of acquisition given by the user

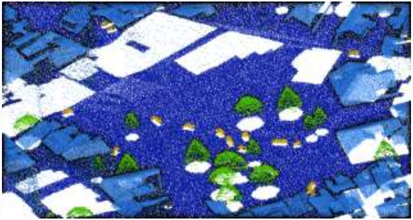
→ **8 different classes**: unchanged, new building, demolition, new vegetation, vegetation loss, vegetation growth, mobile objects

→ Mono-date semantic labels also available

Simulated pair of point clouds

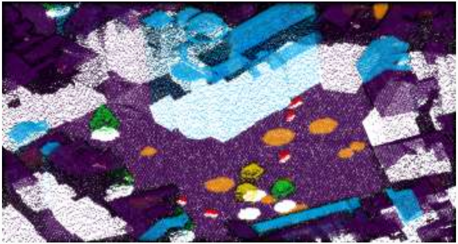
● Ground ● Building ● Vegetation ● Mobile Objects

Date 1



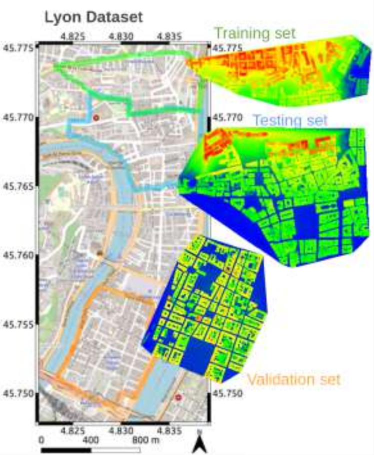
Date 2

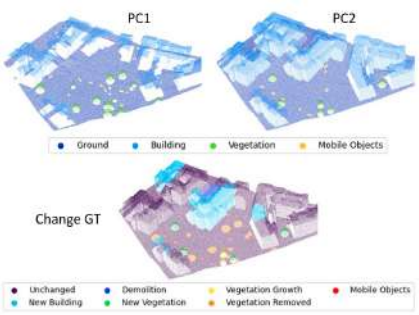
Labelling according to the change



● Unchanged ● New Building ● Demolition ● New Vegetation
● Vegetation Growth ● Vegetation Removed ● Mobile Objects

Urban 3D Point Clouds Changes Dataset





Parameters	Sub-datasets LiDAR low res. Urb3DCD-1	MS Urb3DCD-2
Amount of training pairs	10	10
Density (points/m²)	0.5	0.5 / 10
Noise range across track (°)	0.01	0.2 / 0.01
Noise range along track (°)	0	0.2 / 0
Noise scan direction (m)	0.05	1 / 0.05
Scan angle (°)	-20 to 20	-20 to 20
Overlapping (%)	10	10

⇒ This dataset is publicly available:
https://ieee-dataport.org/open-access/urb3dcd-urban-point-clouds-simulated-dataset-3d-change-detection

Learning strategies

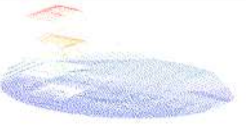
→ Cylindrical inputs for remote sensing large point clouds: $R = 25 \times dl_0$ (dl_0 input subsampling cell size)

→ Unbalanced classes: Input cylinders chosen thanks to a weighted random drawing

→ Loss: SGD with momentum (0.98) to minimize a point-wise weighted negative log likelihood loss

→ Data augmentation:

- Random rotation around vertical axis
- Point scale random Gaussian noise

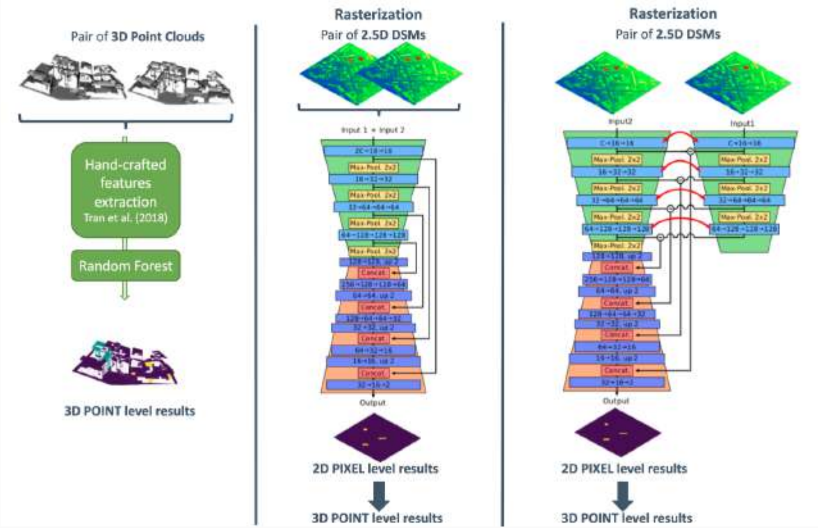


First cylinder

Second cylinder

Experimental Protocol

RF | DSM-FC-EF | DSM-Siamese



Tran et al. 2018 | Daudt et al. 2018 Zhang et al. 2019 | Daudt et al. 2018 Zhang et al. 2019

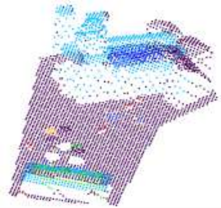
Urb3DCD – LiDAR low density – Qualitative results

(a) PC1

(b) PC2

(c) GT

- Ground
- Building
- Vegetation
- Mobile Objects



(d) RF

(e) DSM-FC-EF

(f) Pseudo-Siamese KPConv

- Unchanged
- New Building
- Demolition
- New Vegetation
- Vegetation Growth
- Missing Vegetation
- Mobile Objects

Urb3DCD – LiDAR low density – Quantitative results

Method	mAcc	$mIoU_{ch}$
Siamese KPConv	90.03 ± 0.69	81.54 ± 1.00
Pseudo-Siamese KPConv	$\mathbf{93.98} \pm 1.26$	$\mathbf{83.77} \pm 1.20$
DSM-Siamese	80.91 ± 5.29	57.41 ± 3.77
DSM-Pseudo-Siamese	75.17 ± 10.03	55.30 ± 8.17
DSM-FC-EF	81.47 ± 0.55	56.98 ± 0.79
RF	65.82 ± 0.05	52.37 ± 0.10

$\Rightarrow$ Large increase of performance with our (Pseudo-)Siamese KPConv

$\Rightarrow$ High results on harder classes (vegetation growth, mobile object)

Urb3DCD – Multi-Sensor – Qualitative results

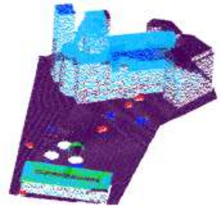


● Ground ● Building ● Vegetation ● Mobile Objects

(a) PC1

(b) PC2

(c) GT



(d) RF

(e) DSM-FC-EF

(f) Pseudo-Siamese KPConv

● Unchanged ● New Building ● Demolition ● New Vegetation
● Vegetation Growth ● Missing Vegetation ● Mobile Objects

Urb3DCD – Multi-Sensor – Quantitative results

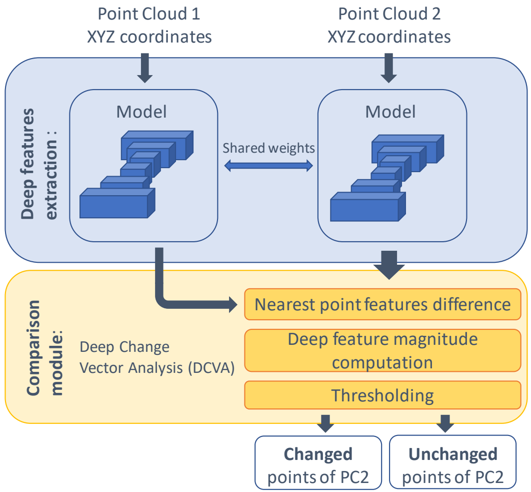
Method	mAcc	mIoU$_{ch}$
Siamese KPConv	58.19 $\pm$ 8.51	36.75 $\pm$ 5.46
Pseudo-Siamese KPConv	**89.74** $\pm$ 1.19	**75.59** $\pm$ 0.67
DSM-Siamese	69.91 $\pm$ 6.18	49.14 $\pm$ 4.92
DSM-Pseudo-Siamese	66.50 $\pm$ 10.82	46.60 $\pm$ 10.13
DSM-FC-EF	81.25 $\pm$ 1.86	55.59 $\pm$ 1.84
RF Tran et al. 2018	62.20 $\pm$ 0.02	46.81 $\pm$ 0.01

$\Rightarrow$ Unshared weights version of our network is better for multi-sensor dataset

What if no annotations are available?

Unsupervised learning and self-supervised learning

Idea: Find indirect strategies for the network to learn discriminating features



Inspired from unsupervised 2D change detection: Saha, Bovolo, et al. 2019

Supervised Semantic Segmentation Training

⇒ Use annex dataset: Supervised Semantic Segmentation Training (SSST)

Hessigheim 3D (H3D) ALS dataset (Kölle et al. 2021)

Self-Supervised Learning

⇒ Use different sub-tasks: Self-Supervised Learning (SSL)

Inspired from unsupervised 2D change detection between SAR and optical data:

Saha, Ebel, et al. 2021

Self-Supervised Learning

$\Rightarrow$ Use different sub-tasks: Self-Supervised Learning (SSL)

Based on two assumptions:

- Changes occur very few

Self-Supervised Learning

$\Rightarrow$ Use different sub-tasks: Self-Supervised Learning (SSL)

Based on two assumptions:

- Changes occur very few
- PCs properties : point distribution never similar

Self-Supervised Learning

$\Rightarrow$ Use different sub-tasks: Self-Supervised Learning (SSL)

Based on two assumptions:

- Changes occur very few
- PCs properties : point distribution never similar

Training alternatively with the following losses:

- $\mathcal{L}_{1,2}$: **Temporal consistency** loss: enforce similar predictions for tiles at similar places

Self-Supervised Learning

$\Rightarrow$ Use different sub-tasks: Self-Supervised Learning (SSL)

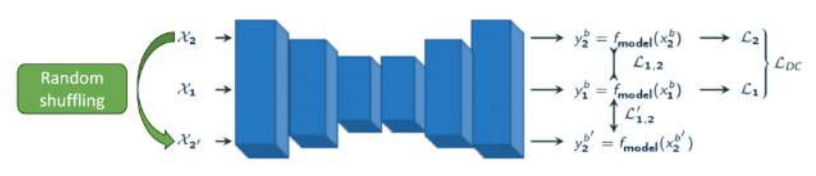
Based on two assumptions:

- Changes occur very few
- PCs properties : point distribution never similar

Training alternatively with the following losses:

- $\mathcal{L}_{1,2}$: **Temporal consistency** loss: enforce similar predictions for tiles at similar places
- $\mathcal{L}'_{1,2}$: **Contrastive** loss: enforce dissimilar features for tiles different places

$\Rightarrow$ Use different sub-tasks: Self-Supervised Learning (SSL)



Based on two assumptions:

- Changes occur very few
- PCs properties : point distribution never similar

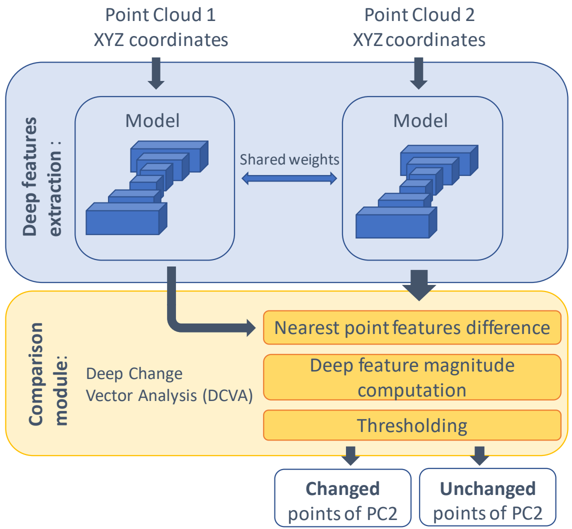
Training alternatively with the following losses:

- $\mathcal{L}_{1,2}$: **Temporal consistency** loss: enforce similar predictions for tiles at similar places
- $\mathcal{L}'_{1,2}$: **Contrastive** loss: enforce dissimilar features for tiles different places
- $\mathcal{L}_{DC}$: **Deep Clustering** loss: force the network to learn discriminative features

Self-Supervised learning

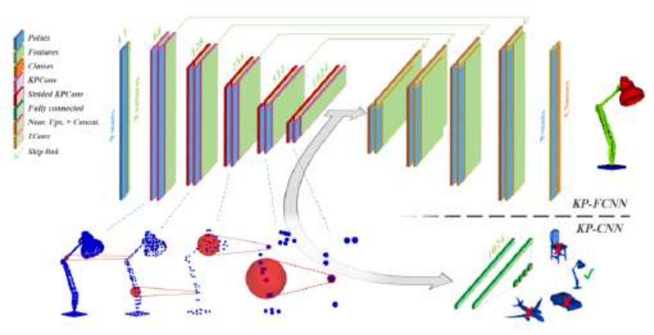
Algorithm 1 Self-supervised training of the backbone

Initialize KP-FCNN weights
for $e \leftarrow 1$ to $\mathcal{E}$ **do**
  Sample $\mathcal{B}$ tiles from $\mathcal{P}_{\mathbf{1}}$, denoted as $\mathcal{X}_{\mathbf{1}}$
  Obtain corresponding $\mathcal{B}$ tiles from $\mathcal{P}_{\mathbf{2}}$, denoted as $\mathcal{X}_{\mathbf{2}}$
  Obtain $\mathcal{X}'_{\mathbf{2}}$ as random shuffling of $\mathcal{X}_{\mathbf{2}}$
  for $i \leftarrow 0$ to $\mathcal{I} - 1$ **do**
    for $b \in \mathcal{B}$ **do**
      $y^b_{\mathbf{1}} = f_{\textbf{KP-FCNN}}(x^b_{\mathbf{1}})$
      $y^b_{\mathbf{2}} = f_{\textbf{KP-FCNN}}(x^b_{\mathbf{2}})$
      $y^{b'}_{\mathbf{2}} = f_{\textbf{KP-FCNN}}(x^{b'}_{\mathbf{2}})$
    Compute the weights W_k considering $y^b_{\mathbf{1}}$ and $y^b_{\mathbf{2}}$
    Calculate weighted deep clustering losses $\mathcal{L}_{\mathbf{1}}$ and $\mathcal{L}_{\mathbf{2}}$
    Calculate temporal consistency loss $\mathcal{L}_{\mathbf{1},\mathbf{2}}$
    Calculate contrastive loss $\mathcal{L}'_{\mathbf{1},\mathbf{2}}$
    if i mod $3 = 0$ **then**
      Use $\mathcal{L}_{DC} = \frac{\mathcal{L}_{\mathbf{1}} + \mathcal{L}_{\mathbf{2}}}{\mathbf{2}}$ to modulate KP-FCNN weights
    else if i mod $3 = 1$ **then**
      Use $\mathcal{L}_{\mathbf{1},\mathbf{2}}$ to modulate KP-FCNN weights
    else
      Use $\mathcal{L}'_{\mathbf{1},\mathbf{2}}$ to modulate KP-FCNN weights


Point Cloud 1
XYZ coordinates
Point Cloud 2
XYZ coordinates
Deep features extraction :
Model
Shared weights
Model
Comparison module:
Deep Change Vector Analysis (DCVA)
Nearest point features difference
Deep feature magnitude computation
Thresholding
Changed points of PC2
Unchanged points of PC2

DCVA - Experimental settings

- Backbone model: KP-FCNN (Thomas et al. 2019)



DCVA - Experimental settings

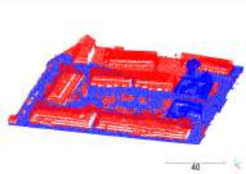
- Backbone model: KP-FCNN (Thomas et al. 2019)
- Dataset for the experimentation: Actueel Hoogtebestand Nederland (AHN)

- Backbone model: KP-FCNN (Thomas et al. 2019)
- Dataset for the experimentation: Actueel Hoogtebestand Nederland (AHN)
  - → Manual annotation of the test set.

AHN 3

AHN 4

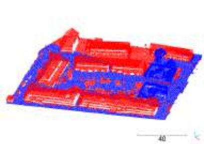
Ground Truth



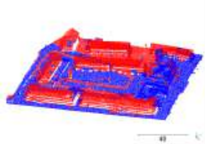
DCVA - Results

a) AHN3 data (time 1)

b) AHN4 data (time 2)



c) Ground truth



d) SSL-DCVA results

		IoU	
	mAcc	Unchange	Change
SSL-DCVA	**85.20**	**78.91**	**69.38**
SSST-DCVA	81.88	70.02	63.85
Siam. KPConv transfer **(de Gélis et al. 2021a)**	79.29	77.89	58.89
M3C2 **(Lague et al. 2013)**	51.77	3.66	39.90
Siam. KPConv (supervised) **(de Gélis et al. 2021a)**	97.08	95.39	92.95

Conclusion

Conclusions

- End-to-end **deep learning** method for **change detection and categorization on raw 3D point clouds**
- Application to **urban** and **geoscience** environment
  - *Urban application*: **Simulator** of **multi-temporal** urban **3D PCs** with **automatic annotation**
  - *Urban application*: IoU on classes of change: $\sim$ **+ 30 %** compared to RF with hand-crafted features (machine learning method)
- **Unsupervised binary change detection**: self-supervised learning
- **Nowadays works**:

  $\rightarrow$ Unsupervised change detection with multi-class results

Bibliographie i

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

Xu, Hao et al. (2015). "Using octrees to detect changes to buildings and trees in the urban environment from airborne LiDAR data". In: *Remote Sensing* 7.8, pp. 9682–9704.

Xu, Sudan, George Vosselman, and Sander Oude Elberink (2015). "Detection and classification of changes in buildings from airborne laser scanning data". In: *Remote sensing* 7.12, pp. 17051–17076.

Daudt, Rodrigo Caye, Bertrand Le Saux, and Alexandre Boulch (2018). "Fully convolutional siamese networks for change detection". In: *2018 25th IEEE International Conference on Image Processing (ICIP)*. IEEE, pp. 4063–4067.

- Tran, T.H.G., C. Ressl, and N. Pfeifer (2018). "Integrated change detection and classification in urban areas based on airborne laser scanning point clouds". In: *Sensors* 18.2, p. 448.
- Saha, Sudipan, Francesca Bovolo, and Lorenzo Bruzzone (2019). "Unsupervised deep change vector analysis for multiple-change detection in VHR images". In: *IEEE TGRS* 57.6, pp. 3677–3693.
- Thomas, Hugues et al. (2019). "Kpconv: Flexible and deformable convolution for point clouds". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6411–6420.
- Young, Adam P and Jessica E Carilli (2019). "Global distribution of coastal cliffs". In: *Earth Surface Processes and Landforms* 44.6, pp. 1309–1316.
- Zhang, Z. et al. (2019). "Detecting building changes between airborne laser scanning and photogrammetric data". In: *Remote sensing* 11.20, p. 2417.

Lebègue, L. et al. (2020). "CO3D, a Worldwide One One-Meter Accuracy dem for 2025". In: *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences* 43, pp. 299–304.

Allan, Richard P et al. (2021). *IPCC, 2021: Summary for Policymakers*.

de Gélis, Iris, Sébastien Lefèvre, and Thomas Corpetti (2021a). "3D Urban Change Detection with Point Cloud Siamese Networks". In: *ISPRS-International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* 43, pp. 879–886.

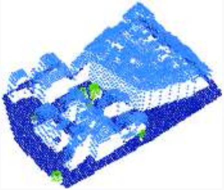
— (2021b). "Change Detection in Urban Point Clouds: An Experimental Comparison with Simulated 3D Datasets". In: *Remote Sensing* 13.13, p. 2629.

Bibliographie iv

- Kölle, Michael et al. (2021). "The Hessigheim 3D (H3D) benchmark on semantic segmentation of high-resolution 3D point clouds and textured meshes from UAV LiDAR and Multi-View-Stereo". In: *ISPRS Open Journal of Photogrammetry and Remote Sensing* 1, p. 100001.
- Saha, Sudipan, Patrick Ebel, and Xiao Xiang Zhu (2021). "Self-supervised multisensor change detection". In: *IEEE TGRS* 60, pp. 1–10.

Thank you for your attention

Urb3DCD – LiDAR low density – Qualitative results : Occlusions

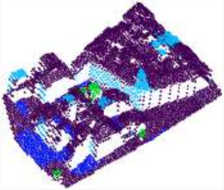


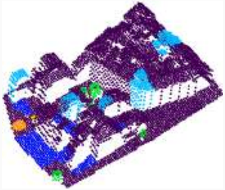
● Ground ● Building ● Vegetation ● Mobile Objects

(a) PC1

(b) PC2

(c) GT





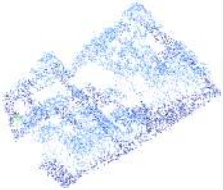
(d) RF

(e) DSM-FC-EF

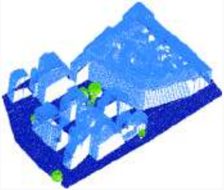
(f) Pseudo-Siamese KPConv

● Unchanged ● New Building ● Demolition ● New Vegetation
● Vegetation Growth ● Missing Vegetation ● Mobile Objects

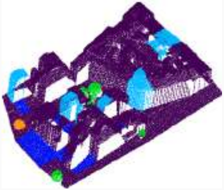
Urb3DCD – Multi-Sensor – Qualitative results : Occlusions



(a) PC1

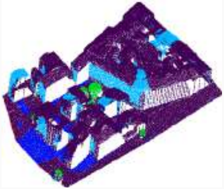


(b) PC2

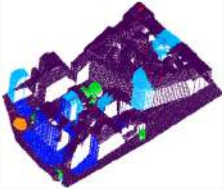


(c) GT

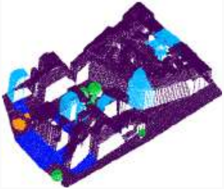
Ground · Building · Vegetation · Mobile Objects



(d) RF



(e) DSM-FC-EF



(f) Pseudo-Siamese KPConv

Unchanged · New Building · Demolition · New Vegetation
Vegetation Growth · Missing Vegetation · Mobile Objects