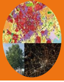


Détection de changement en milieu urbain au sein de nuages de points 3D par apprentissage profond

Iris de Gélis[1,2], Sébastien Lefèvre[2] and Thomas Corpetti[3]

[1]Magellium, F-31000, Toulouse, France; iris.de-gelis@irisa.fr
[2]IRISA UMR 6074, Université Bretagne Sud, F-56000 Vannes, France
[3]CNRS, LETG UMR 6554, F-35000 Rennes, France
This work is part of a PhD funded by the CNES and Magellium

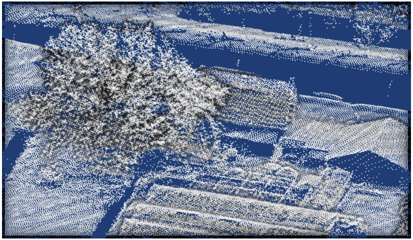
Change detection

Google Earth Timelapse (Google, Landsat, Copernicus)

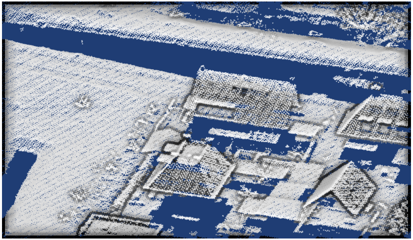
Wikipedia

3D point clouds for change detection

Date 1

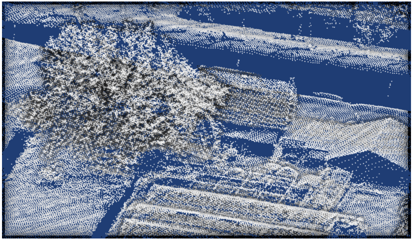


Date 2

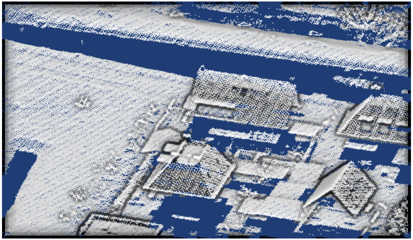


3D point clouds for change detection

Date 1



Date 2

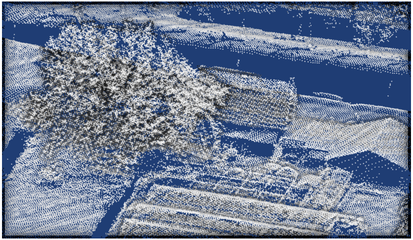


- Sparse
- Unordered

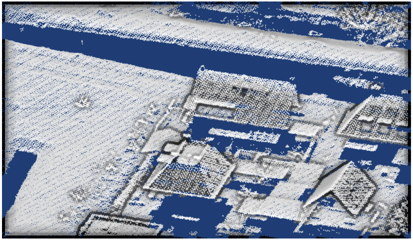
Unlike 2D images:

3D point clouds for change detection

Date 1



Date 2



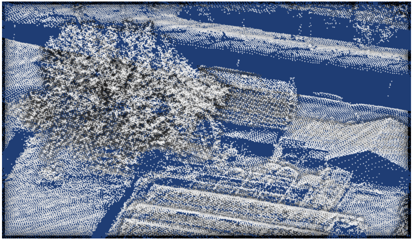
- Sparse
- Unordered

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

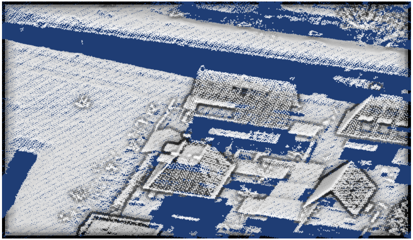
Unlike 2D images:

3D point clouds for change detection

Date 1



Date 2



- Sparse
- Unordered

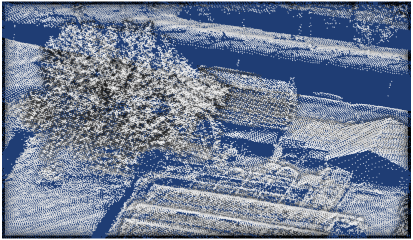
$\Big\}$ Raw PCs $\neq$ matrices

Unlike 2D images:

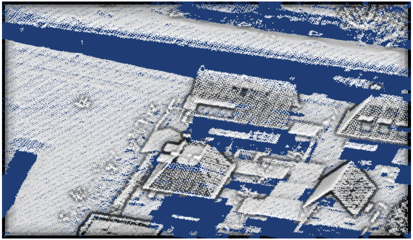
- No direct comparison possible

3D point clouds for change detection

Date 1



Date 2

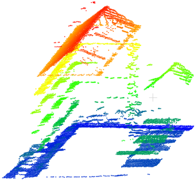


- Sparse
- Unordered

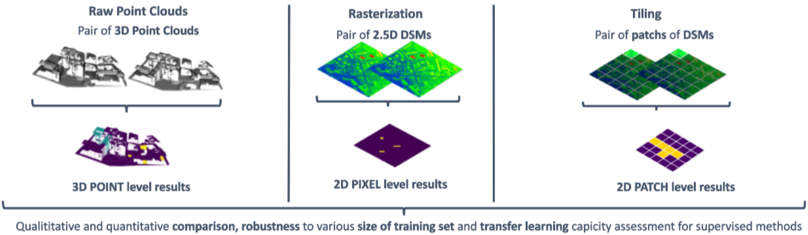
$\Big\}$ Raw PCs $\neq$ matrices

Unlike 2D images:

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



Benchmark of methods for change detection



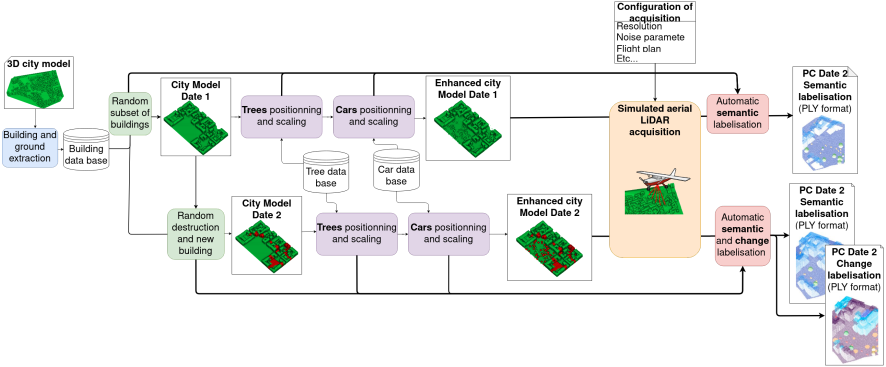
$\Rightarrow$ Majority of methods only focus on **DSMs** : loss of information

$\Rightarrow$ **Deep learning** method on produce a binary per 2D patch results

$\Rightarrow$ Existing **traditional methods** scores can be **largely improved**

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

Urb3DCD - Simulator for 3D PCs in urban environment



→ **Automatic annotation** of PCs

→ Configuration of acquisition given by the user

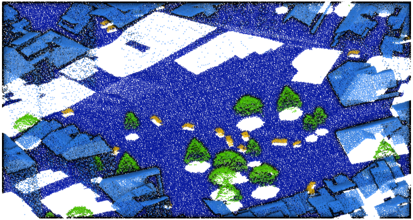
→ **8 different classes**: unchanged, new building, demolition, new vegetation, vegetation loss, vegetation growth, mobile objects

→ Mono-date semantic labels also available

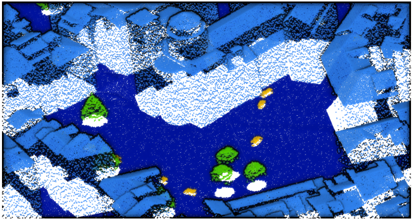
3D point clouds coming from our simulator V2

● Ground ● Building ● Vegetation ● Mobile Objects

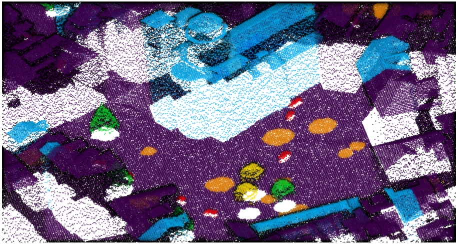
Date 1



Date 2

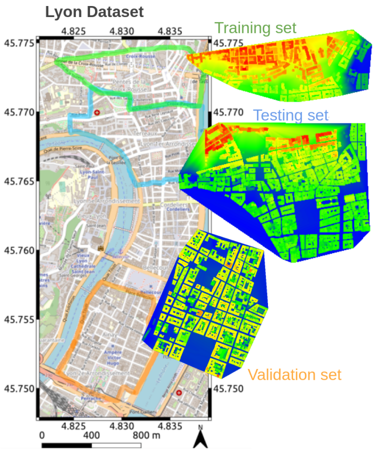


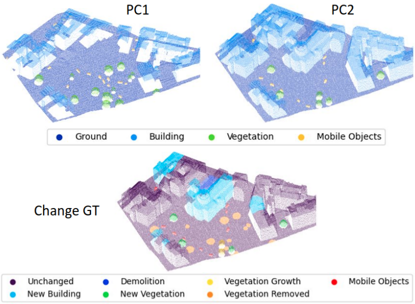
Labelling according to the change



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

Urban 3D Point Clouds Changes Dataset





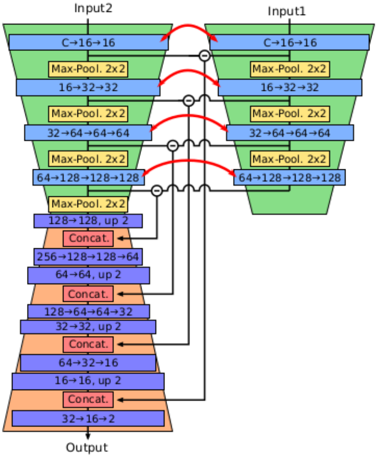
Parameters	Sub-datasets: LiDAR low res. Urb3DCD-1	Sub-datasets: MS Urb3DCD-2
Amount of training pairs	10	10
Density (points/m^2)	0.5	0.5 / 10
Noise range across track (°)	0.01	0.2 / 0.01
Noise range along track (°)	0	0.2 / 0
Noise scan direction (m)	0.05	1 / 0.05
Scan angle (°)	-20 to 20	-20 to 20
Overlapping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

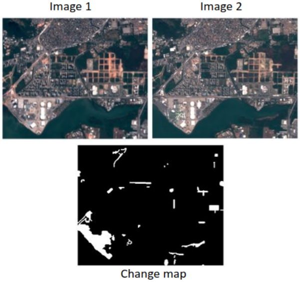
⇒ This dataset is publicly available:

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

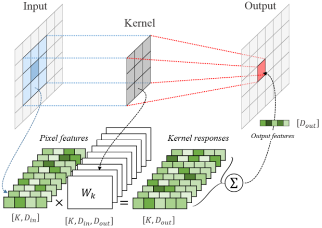
2D change detection and categorization : Siamese Networks

Fully Convolutional Siamese Network with difference

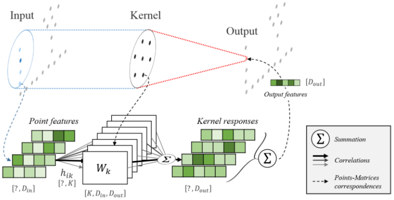




2D Convolution



3D Kernel Point Convolution

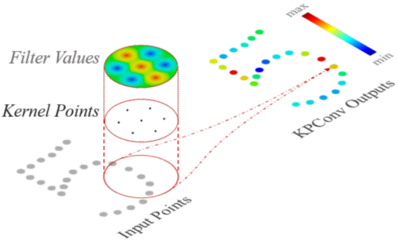


Thomas et al. 2019

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



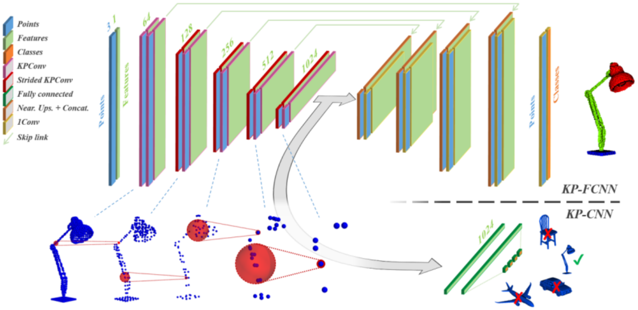
Thomas et al. 2019

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

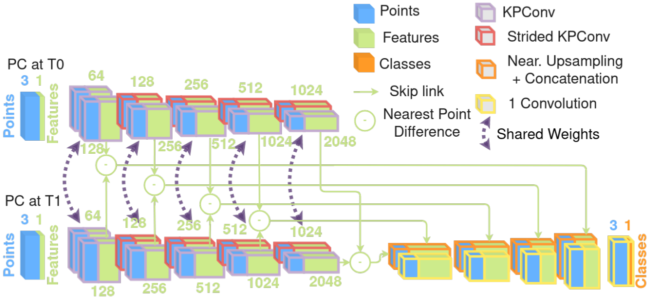


Thomas et al. 2019

⇒ Network that looks like traditional 2D images CNN

Siamese KPConv : deep network for 3D PCs change detection

Siamese Kernel Point Convolution Network



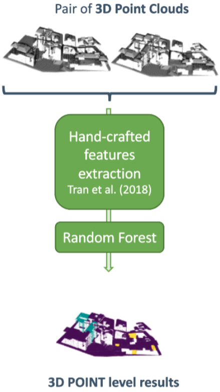
de Gélis, Lefèvre, and Corpetti 2021a

→ 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 $\mathcal{F}_1$

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

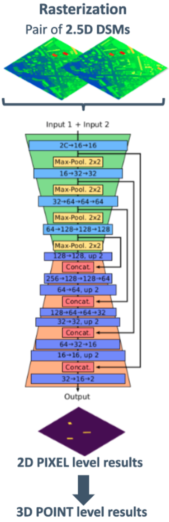
Experimental Protocol

RF



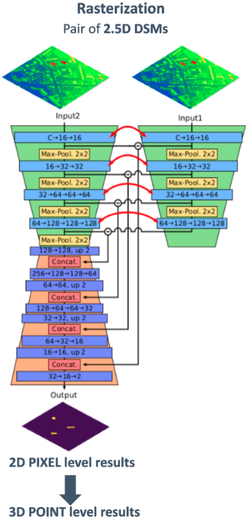
Tran, Ressl, and Pfeifer 2018

DSM-FC-EF



Daudt, Le Saux, and Boulch 2018
Zhang et al. 2019

DSM-Siamese



Daudt, Le Saux, and Boulch 2018
Zhang et al. 2019

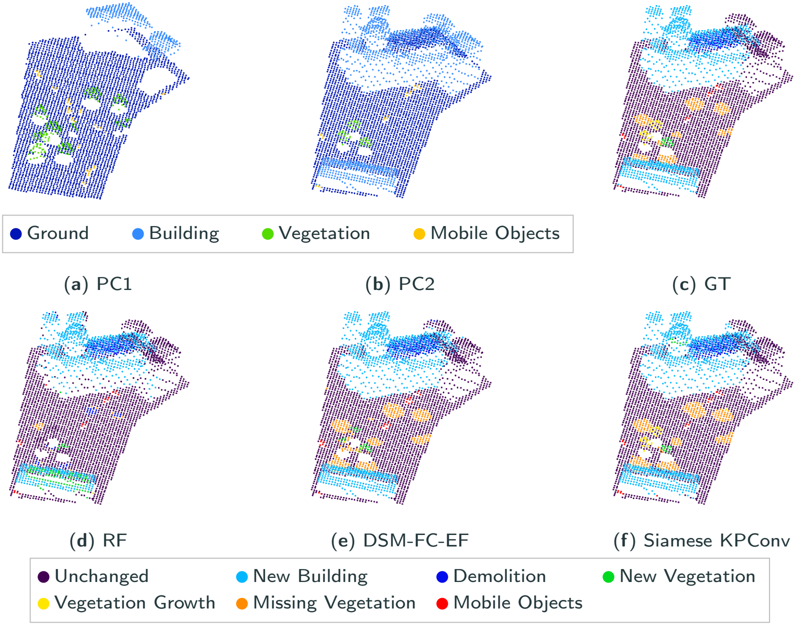
Urb3DCD – LiDAR low density – Quantitative results

Method	mAcc	mIoU$_{ch}$
Siamese KPConv (ours)	91.21 $\pm$ 0.68	**80.12** $\pm$ 0.02
Pseudo-Siamese KPConv (ours)	**91.31** $\pm$ 2.34	77.80 $\pm$ 1.69
DSM-Siamese	80.91 $\pm$ 5.29	57.41 $\pm$ 3.77
DSM-Pseudo-Siamese	75.17 $\pm$ 10.03	55.30 $\pm$ 8.17
DSM-FC-EF	81.47 $\pm$ 0.55	56.98 $\pm$ 0.79
RF tran2018integrated	65.82 $\pm$ 0.05	52.37 $\pm$ 0.10

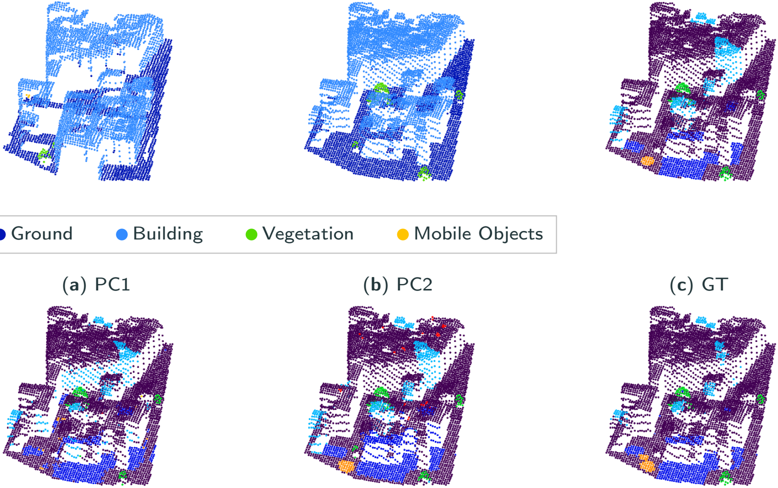
$\Rightarrow$ Large increase of performance with our Siamese KPConv

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

Urb3DCD – LiDAR low density – Qualitative results



Urb3DCD – LiDAR low density – Qualitative results



(a) PC1 (b) PC2 (c) GT

(d) RF (e) DSM-FC-EF (f) Siamese KPConv

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

Conclusions

- **Simulator** of **multi-temporal** urban **3D PCs** with **automatic annotation**
- End-to-end **deep learning** method for **change detection and categorization on raw 3D point clouds**
- IoU on classes of change: $\sim$ **+ 30 %** compared to RF with hand-crafted features
- **Nowadays works**:

  $\rightarrow$ Low supervised learning

    - Unsupervised learning for binary change extraction (SSST-DCVA, SSL-DCVA)
    - Low supervised learning multi-class change extraction (DC3DCD)

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.

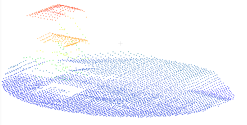
Tran, T.H.G., C. Ressl, and N. Pfeifer (2018). "Integrated change detection and classification in urban areas based on airborne laser scanning point clouds". In: *Sensors* 18.2, p. 448.

- 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.
- Zhang, Z. et al. (2019). "Detecting building changes between airborne laser scanning and photogrammetric data". In: *Remote sensing* 11.20, p. 2417.
- 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.

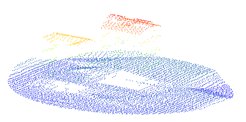
Merci de votre attention

Learning strategies

- → Cylindrical inputs for remote sensing large point clouds: $R = 50 \times dl_0$ (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

Frame Title

S. Xu, Vosselman, and Oude Elberink 2015 H. Xu et al. 2015