From Points to Prints

Generating Building Roofprints and Footprints from Airborne Lidar Data and Inaccurate Outlines

A3 Presentation

Alexandre Bry, supervised by Hugo Ledoux, Ravi Peters and Bruno Vallet

May 18, 2026

Title
—

Context

Structure similar to the paper

Roofprint vs. footprint

	Roofprint	Footprint
Defined by	Roof	Façades
Comes from	Aerial data (images, ALS)	Terrain measurements (TLS, MLS)
Used for	3D modelling, solar potential	Tax estimation, energy consumption

Table 1: Comparison between roofprints and footprints

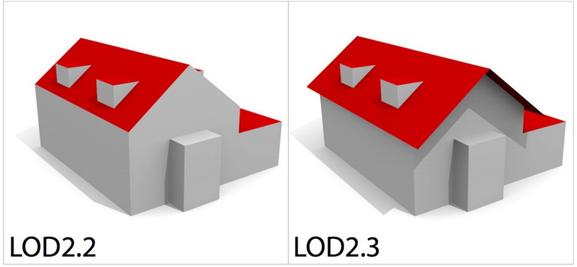


Figure 1: Part of the visual definition of buildings Level of Detail (LoD) containg LoD 2.2 and 2.3 (Biljecki et al., 2016) .

- Roofprints and footprints can serve different purposes, and knowing the difference between them is important when the roof overhangs are significant.
- Since this difference is usually not available, it is unclear what the use cases could be, but we can think about:
  - More accurate visualisation and texturing of 3D models
  - More accurate simulations (wind, luminosity)

Strengths and weaknesses of BD TOPO

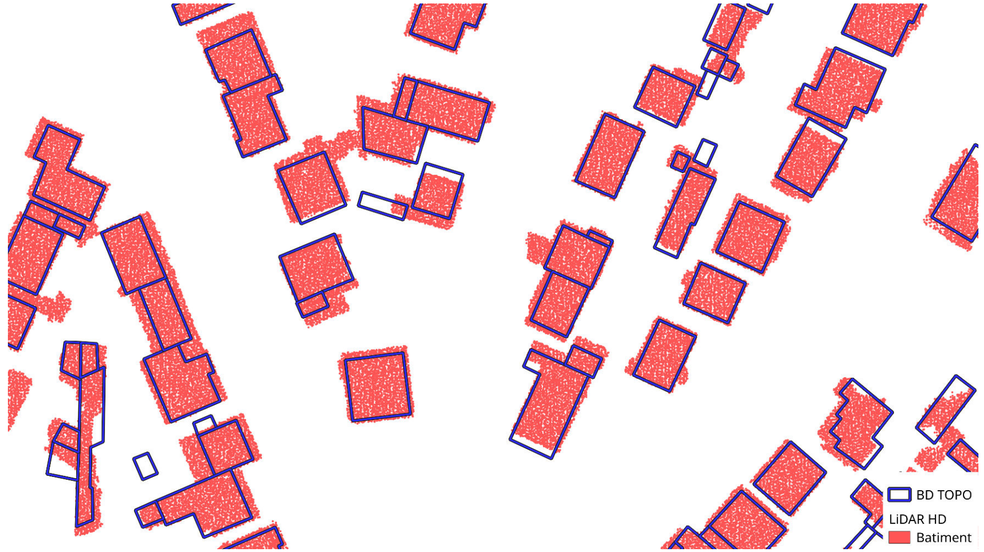


Figure 2: BD TOPO buildings are most of the time footprints, and often shifted.

- Strengths:
  - The BD TOPO contains most of the buildings in France
  - Buildings were manually delineated by humans, so their shape is generally of good quality
- Weaknesses:
  - It is a mix of footprints and roofprints, coming from different sources
  - Georeferencing is not perfect because it was not crucial for the cadastre initially

Research questions

*How to generate coherent building **roofprints** and **footprints** from high-density **ALS** point clouds and existing **imprecise outlines**?*

- How to identify and use the **points on roof edges** in ALS point clouds?
- How to identify and use the points in an ALS point cloud that contain **information about the façades** despite their sparsity?
- How to **deform an imprecise outline** with global and local transformations while preserving the angles of the edges?

Overview of the methodology

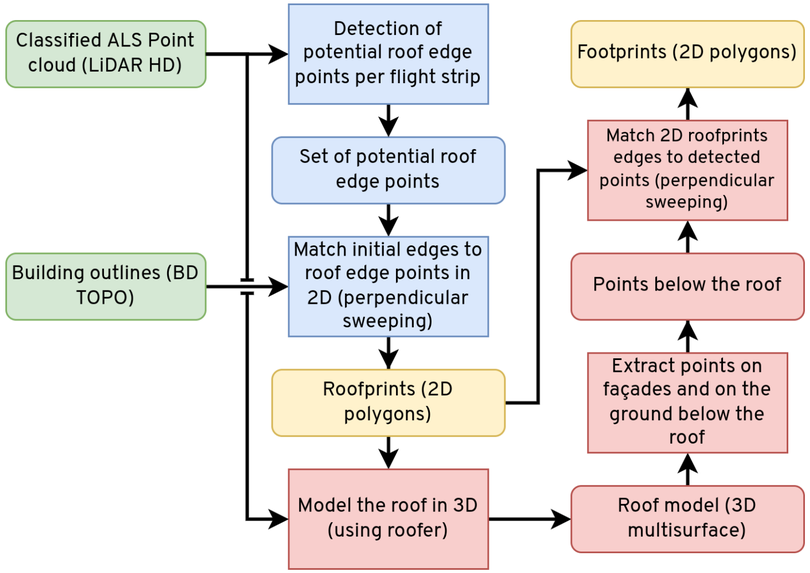


Figure 3: Overview of the pipeline.

- We start with the roofprint because aerial LiDAR data gives a much higher density of points on the roof than on the façades, making the roof much easier to identify.
- Then, once the roofprint is accurately registered on the point cloud, we can use it to generate an accurate 3D roof model that allows to select all the points below the roof (hopefully façade and ground points), which are all the points that provide information about the façades and the roof overhangs.

Polygon deformation

Constraints over the movements

We modify the polygons by applying the following rules:

- **never rotate an edge**,
- **never flip an edge**,
- **move shared edges together**.

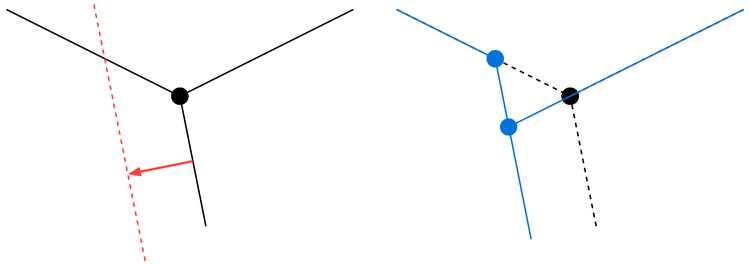
- The constraints ensure that we **preserve some topology** while keeping a **lot of freedom** for the edges to move.
- They could be **further improved** to preserve **shared vertices**, which would be possible but a bit more complex to implement and at the cost of some freedom of motion.

Constraints over the movements

We modify the polygons by applying the following rules:

- **never rotate an edge**,
- **never flip an edge**,
- **move shared edges together**.

However it is not perfect as it can **separate two points** that were initially at the same position.

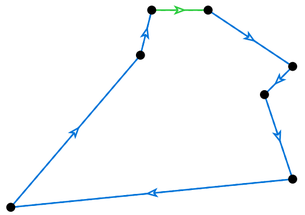


(a) Initial situation with one vertex. (b) Result with two vertices.

Figure 4: A vertex shared by two polygons can be split.

- The constraints ensure that we **preserve some topology** while keeping a **lot of freedom** for the edges to move.
- They could be **further improved** to preserve **shared vertices**, which would be possible but a bit more complex to implement and at the cost of some freedom of motion.

Our simple approach

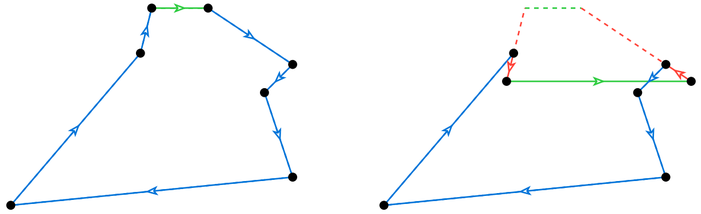


(a) Initial state.

Figure 5: Illustration of the polygon deformation algorithm on an isolated polygon.

This illustrates the behavior of the polygon deformation algorithm, which is designed to **propagate the shift** in the polygon as long as there are **self-intersections** of **flipped edges**.

Our simple approach



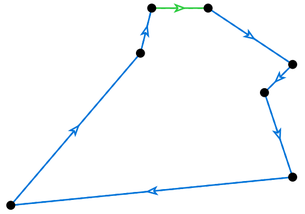
(a) Initial state.

(b) After shifting the focus edge.

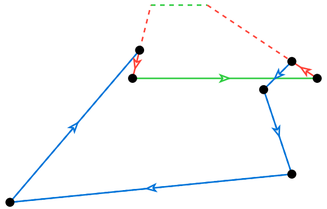
Figure 5: Illustration of the polygon deformation algorithm on an isolated polygon.

This illustrates the behavior of the polygon deformation algorithm, which is designed to **propagate the shift** in the polygon as long as there are **self-intersections** of **flipped edges**.

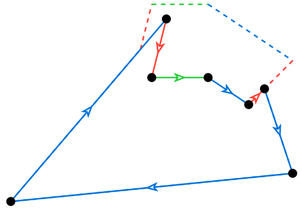
Our simple approach



(a) Initial state.



(b) After shifting the focus edge.

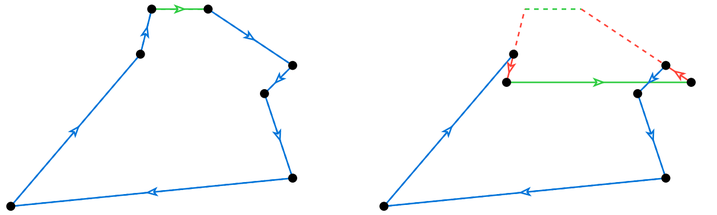


(c) First step of the resolution.

Figure 5: Illustration of the polygon deformation algorithm on an isolated polygon.

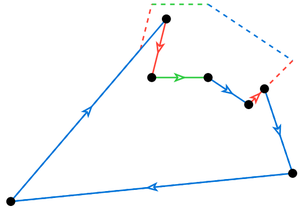
This illustrates the behavior of the polygon deformation algorithm, which is designed to **propagate the shift** in the polygon as long as there are **self-intersections** of **flipped edges**.

Our simple approach

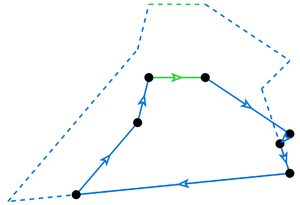


(a) Initial state.

(b) After shifting the focus edge.



(c) First step of the resolution.



(d) Second step of the resolution.

Figure 5: Illustration of the polygon deformation algorithm on an isolated polygon.

This illustrates the behavior of the polygon deformation algorithm, which is designed to **propagate the shift** in the polygon as long as there are **self-intersections** of **flipped edges**.

Roofprints

Point cloud topology

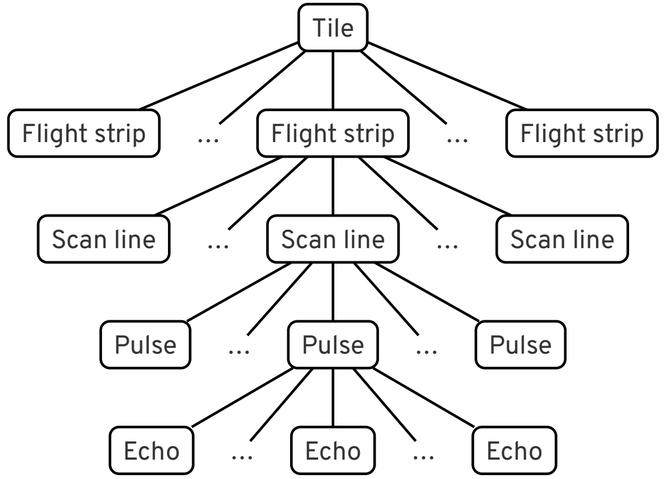


Figure 6: Illustration of the structure of an ALS point cloud.

- This creates a **hierarchy** in the point cloud, with points belonging to pulses, pulses belonging to scan lines, and scan lines belonging to flight strips.
- With multiple flight strips in the same area, this creates a sort of **irregular 3D structure**:
  - 1st dimension: the flight strip
  - 2nd dimension: the scan line
  - 3rd dimension: the pulse

  It is then possible to **navigate** in this structure along any of these dimensions.

Roofprints evidences

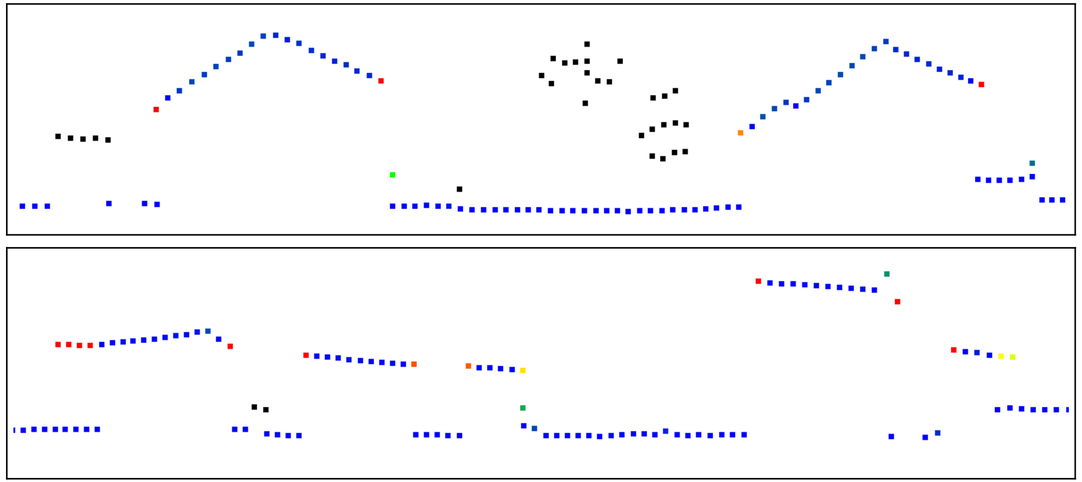


Figure 7: Two examples of the height differences obtained on a scan line. Black points are vegetation points, and for the other points the colour represents the value of Δh, with blue-green-yellow-red from low to high values.

We use the **height differences Δh between neighbor points** to identify points that are likely to be on the edges of roofs:

- We look at the neighbors in the previous, current and next scan lines
- Points at the edges of building roofs have **high values of Δh**
- Points from **high vegetation** also get high Δh and therefore need to be excluded
- With circular ALS sensors, the scan lines are **curved** when looked at from above, but this is not a problem as the **angle is very small** between consecutive pulses

Energy to minimise for roofprints

Total energy:

$$E = \underbrace{-\sum_{i \in \mathcal{P}} w_i \max_{j \in \mathcal{L}} \{\text{score}(p_i, l_j)\}}_{\text{proximity to the points}} + \alpha \underbrace{\sum_{j \in \mathcal{L}} \left(|l_j| - |l_j^0|\right)^2}_{\text{similarity to the initial edges}}$$

- The energy is divided in two terms:
  - A **proximity term** that counts the number of points close to the edges while prioritizing points with higher weights.
  - A **regularization term** that entices the edges to keep their initial length, to prioritize a shape closer to the initial one.
- We chose this regularization term to ensure that the edges will **keep their initial length** and otherwise **share the change equally** between them. This is desired if we assume that the **roof overhang is the same on both sides** along a given direction.
- Notations:
  - $\mathcal{P}$ is the set of points, and $\mathcal{L}$ is the set of edges (lines),
  - w_i is the weight of point p_i, which is independent of the lines,
  - $\text{score}(p_i, l_j)$ is the score of point p_i for edge l_j, which is defined in the next slide,
  - $|l_j|$ is the length of edge l_j in the current configuration, and $|l_j^0|$ is its initial length,
  - α is a parameter to adjust between the two terms.

Energy to minimise for roofprints

Total energy:

$$E = \underbrace{-\sum_{i\in\mathcal{P}} w_i \max_{j\in\mathcal{L}}\{\text{score}(p_i, l_j)\}}_{\text{proximity to the points}} + \alpha \underbrace{\sum_{j\in\mathcal{L}} \left(|l_j| - |l_j^0|\right)^2}_{\text{similarity to the initial edges}}$$

with the score of a pair of point p_i and line l_j:

$$\text{score}(p_i, l_j) = \underbrace{(v_i \cdot n_j)}_{\substack{\text{alignment of} \\ \text{point and edge} \\ \text{'normals'}}} \times \underbrace{\left(1 - \frac{|p_i - p_{i\perp j}|}{\varepsilon}\right)}_{\text{proximity to the edge}} \geq 0$$

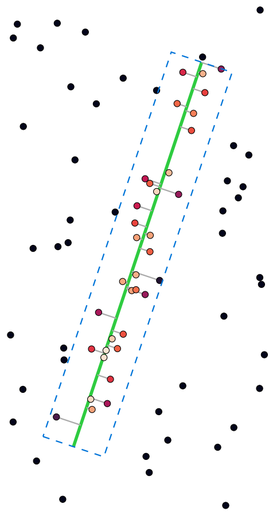


Figure 8: Illustration of the proximity score for an isolated edge.

- Any point **outside the rectangle** defined by extruding the edge along its normal by ε has a score of 0. This geometric ensures that only points that are close enough to the edge will count.
- Then, the dot product between the inward vector and the normal of the edge ensures that points that are part of a parallel but opposite edge will not count, **preventing matching to the neighbour building.**
- Notations:
  - v_i is the inward vector of point p_i, which is defined in the next slide,
  - n_j is the normal of the edge l_j, pointing inwards,
  - $p_{i\perp j}$ is the projection of p_i on the supporting line of l_j
  - ε is a parameter to adjust the size of the rectangle.

Energy to minimise for roofprints

Total energy:

$$E = \underbrace{-\sum_{i\in\mathcal{P}} w_i \max_{j\in\mathcal{L}}\{\text{score}(p_i, l_j)\}}_{\text{proximity to the points}} + \alpha \underbrace{\sum_{j\in\mathcal{L}} \left(|l_j| - |l_j^0|\right)^2}_{\text{similarity to the initial edges}}$$

with the score of a pair of point p_i and line l_j:

$$\text{score}(p_i, l_j) = \underbrace{(v_i \cdot n_j)}_{\substack{\text{alignment of} \\ \text{point and edge} \\ \text{'normals'}}} \times \underbrace{\left(1 - \frac{|p_i - p_{i\perp j}|}{\varepsilon}\right)}_{\text{proximity to the edge}} \geq 0$$

with the inward direction:

$$v_i = \frac{1}{|\mathcal{B}(p_i, \delta)|} \sum_{p_j \in \mathcal{B}(p_i,\delta)} \frac{p_j - p_i}{|p_j - p_i|}$$

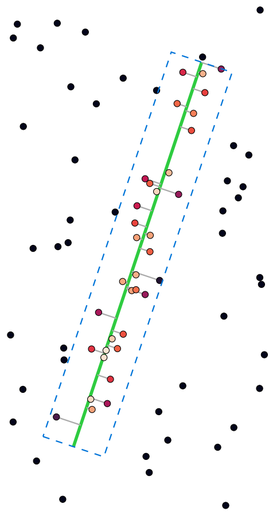
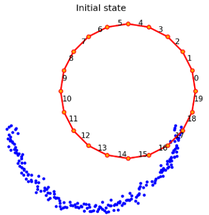


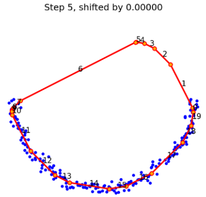
Figure 8: Illustration of the proximity score for an isolated edge.

- The idea behind this method is that points on the edge of the roof should have **many points towards the inside** of the building, and **few points towards the outside**.
- Using unit vectors and averaging them allows for all points in the neighbourhood to contribute equally, meaning that the result could be interpreted as proxy of the **direction of density imbalance**. Therefore, the **magnitude** of the inward vector is an indication of the confidence of the direction.
- Notations:
  - $\mathcal{B}(p_i, \delta)$ is the set of points with a distance less than δ from point p_i
  - δ is a parameter to adjust the size of the neighbourhood.

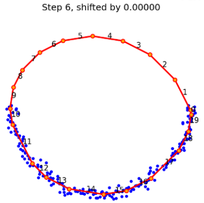
Importance of the two terms on toy examples



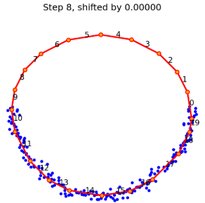
(a) Initial state



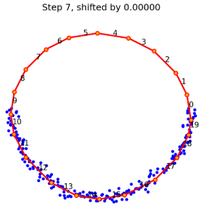
(b) $\alpha = 0.00$ (no regularisation)



(c) $\alpha = 0.05$



(d) $\alpha = 0.20$

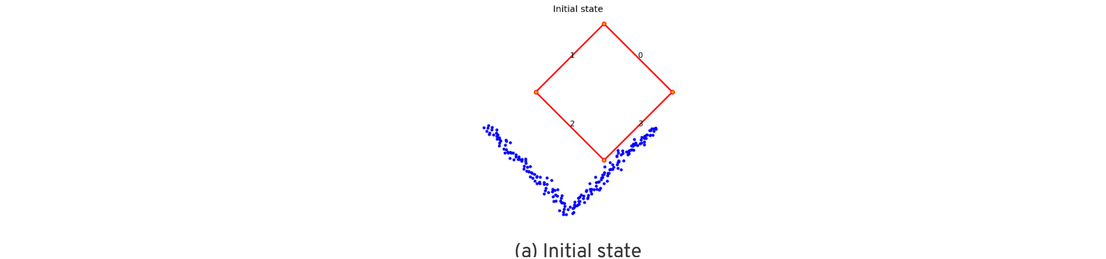


(e) $\alpha = 1.00$

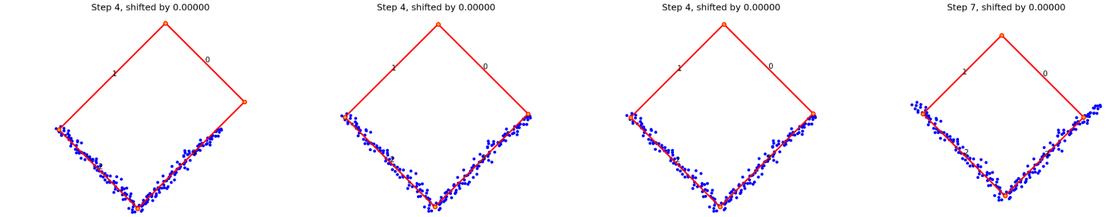
Figure 32: Matching a circle to a half-circle of points with different values of the regularization parameter α.

We can see how regularization helps to get a shape more similar to the target even with half of the points missing.

Importance of the two terms on toy examples



(a) Initial state



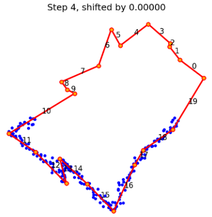
(b) $\alpha = 0.00$ (no regularization) (c) $\alpha = 0.05$ (d) $\alpha = 0.20$ (e) $\alpha = 1.00$

Figure 32: Matching a square to points along two sides of a larger square with different values of the regularization parameter α.

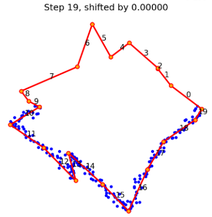
$\alpha > 0$ encourages the shape to be closer to the initial one, which is better than $\alpha = 0$, until a certain point where the regularization is too strong and prevents the shape from extending to capture all the points.

Importance of the two terms on toy examples

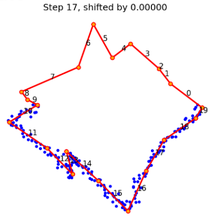
(a) Initial state



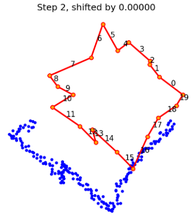
(b) $\alpha = 0.00$ (no regularization)



(c) $\alpha = 0.05$



(d) $\alpha = 0.20$



(e) $\alpha = 1.00$

Figure 32: Matching a polygon to half of its shape with points with different values of the regularization parameter α.

- $\alpha > 0$ forces the shape to be closer to the initial one, but it takes many more steps to converge, because the more edges that don't have points take longer to balance the regularization term between each other.
- If α is too high, it prevents the shape from matching the points, because the proximity score is not good enough to balance the regularization term, which is the case here for $\alpha \geq 0.50$.

Illustration with a real building

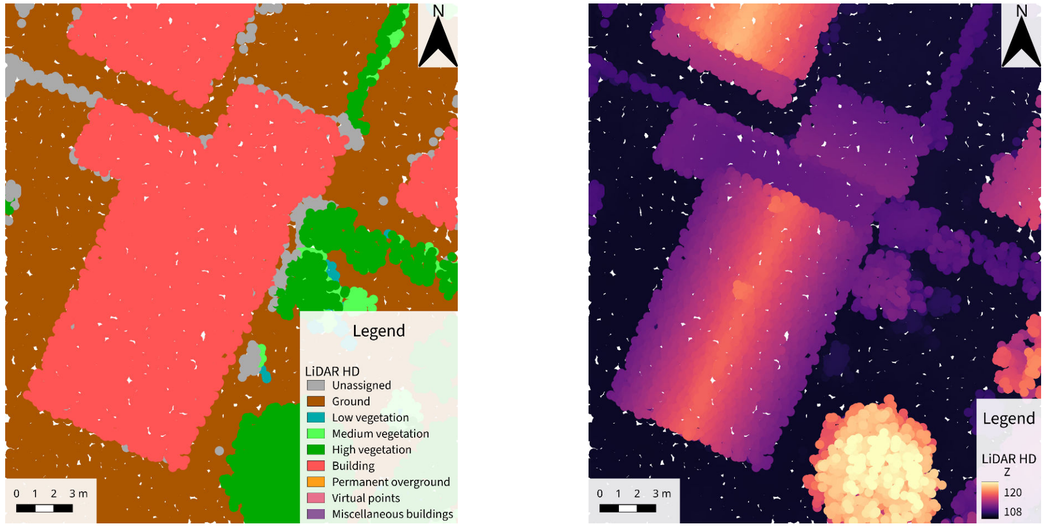


Figure 32: The LiDAR HD data.

- We can see **three different building parts** that are connected, one with a pitched roof and the two others with flat roofs.
- The points are well classified overall, but there are a number of points **classified as Unassigned (grey)**, especially at the boundaries between the building and the vegetation.

Illustration with a real building

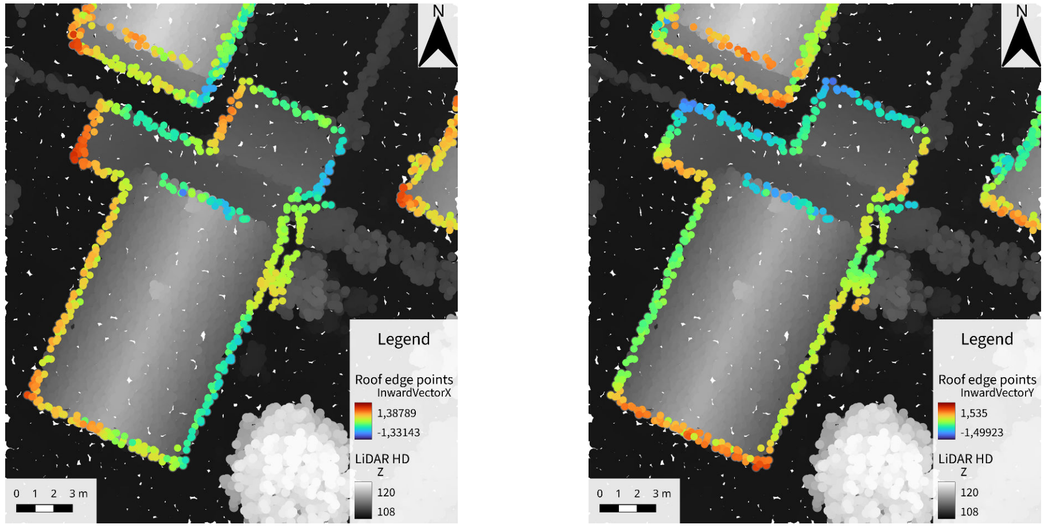


Figure 32: The detected roof edge points coloured by their inward vector.

- The detected roof edge points are shown here, coloured by their **inward vector** (the direction in which the building is located from the edge).
- We can see that the inward vectors are **generally well oriented** towards the building, except in areas where the building is **very close to vegetation** at the same height.

Illustration with a real building

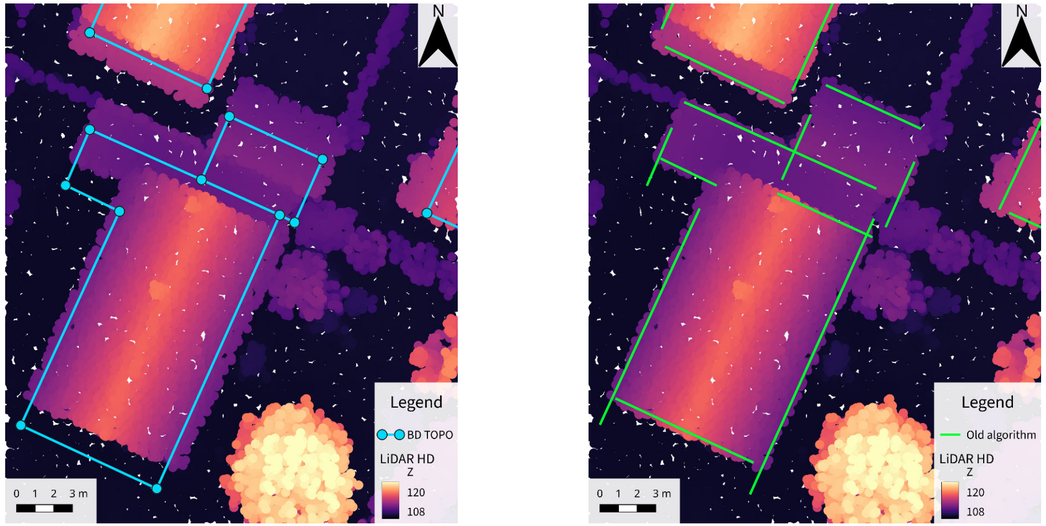


Figure 32: Comparison of BD TOPO outlines and roofprints computed with the old algorithm.

- Interestingly, this building unit is actually divided into **two parts in the BD TOPO**, even though we could expect 1 or 3 as well.
- In the old algorithm, we processed edges individually, resulting in the segments that are displayed on the right.
- The results give a better alignment, but there is still an issue to fix: the bottom left edge of the small top right building was **matched incorrectly**. This could be fixed by **matching the whole initial line once** instead of keeping it split between buildings.

Illustration with a real building

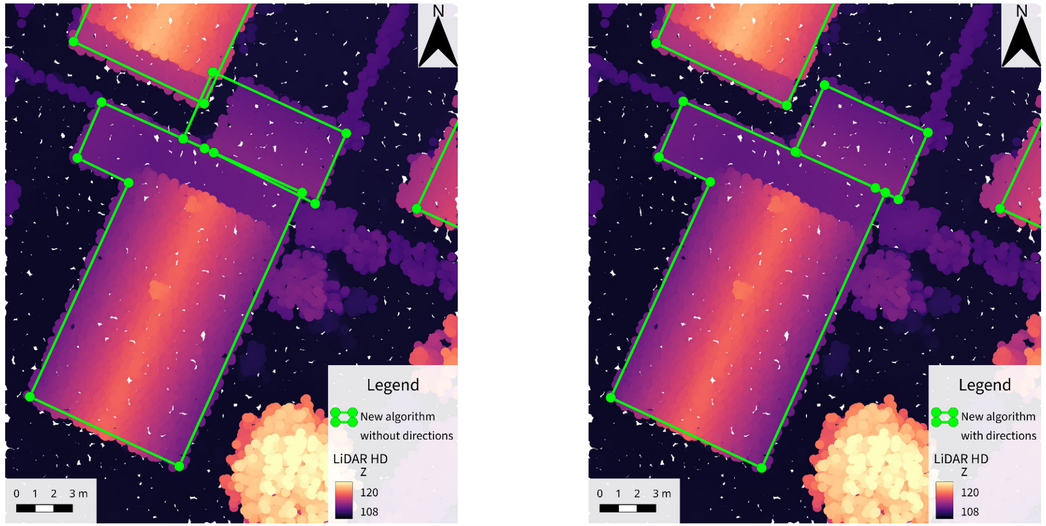


Figure 32: Comparison of the roofprints computed with the final algorithm without and with inward vectors.

- Grouping the shared edges and processing all edges together **fixes the issue** for the bottom left edge of the small top right building.
- However, for the same building without using the inward vectors, one edge ends up **matching the neighbouring building** instead. This is fixed with the inward vectors, which prevent the points from the other building from counting in the score of the edge.

Footprints

Footprints evidences

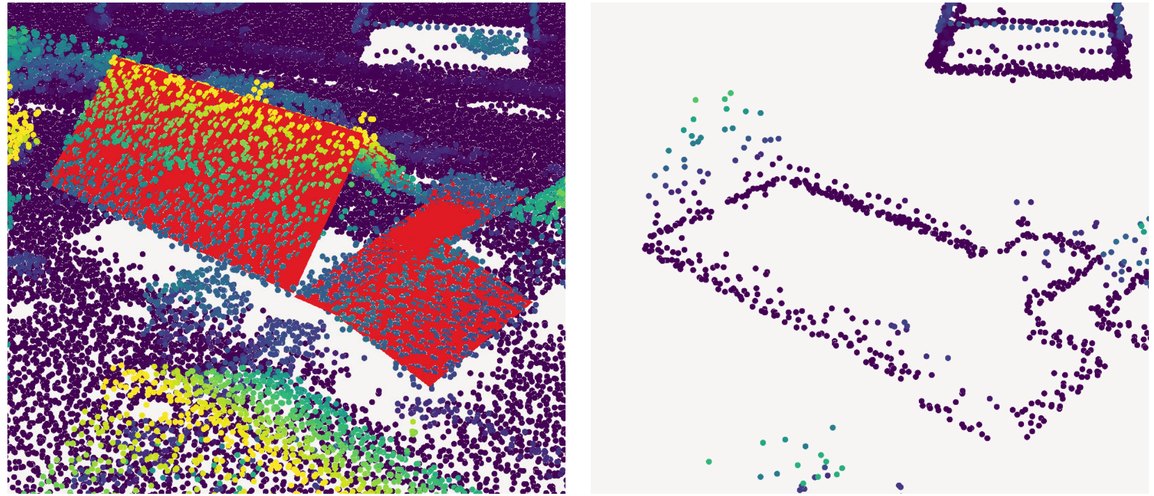


Figure 32: Illustration of the 3D roof structure and the selected points for the footprint computation.

- Process:
  1. Build the roof in 3D using a 3D roof constructor (such as roofer) with the roofprints as input
  2. Select all the points **under** the 3D roof with a small horizontal buffer (for the scoring function) and a small vertical buffer (for roof points slightly below the roof)
  - It seems very simple in practice but actually requires to use the complex roof reconstruction algorithms developed in the past few years.

Footprints evidences

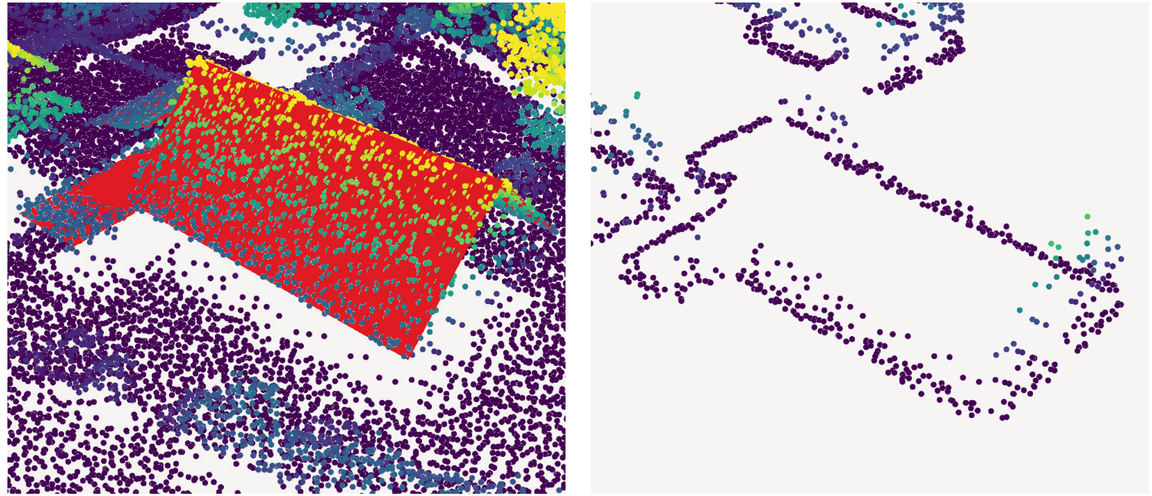


Figure 32: Illustration of the 3D roof structure and the selected points for the footprint computation.

- Process:
  1. Build the roof in 3D using a 3D roof constructor (such as roofer) with the roofprints as input
  2. Select all the points **under** the 3D roof with a small horizontal buffer (for the scoring function) and a small vertical buffer (for roof points slightly below the roof)
  - It seems very simple in practice but actually requires to use the complex roof reconstruction algorithms developed in the past few years.

Energy to minimise for footprints

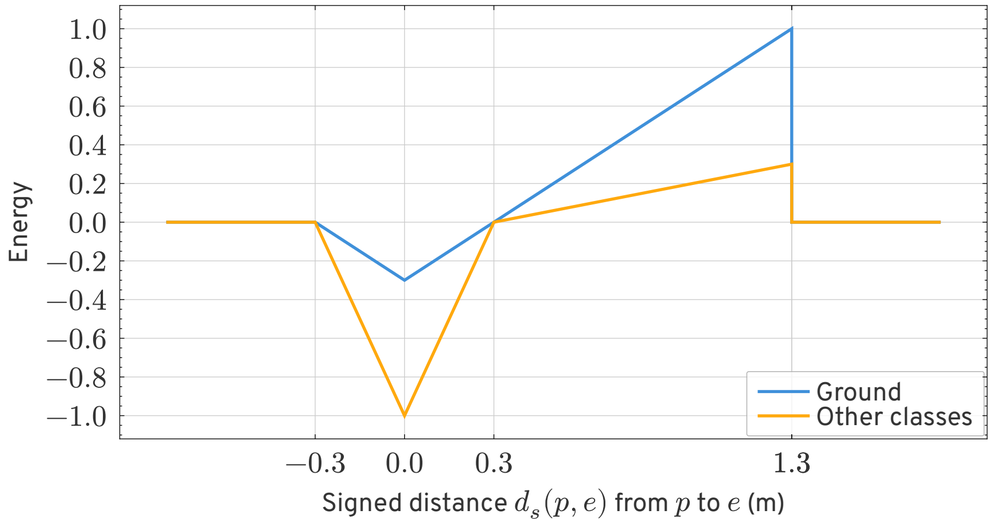


Figure 18: Energy of a point p for a footprint edge e.

- The ideas behind the two terms of the score are simple:
  - The **proximity term** encourages finding a position with a concentration of points (likely to be the façade in 3D)
  - The **penalty term** discourages positions that have points behind them (likely to be points on the façade or on the ground outside the building)
- The difference between ground points and other classes of points comes from the ground points being very good indicators for the penalty term but much less for the proximity, while the others are assumed to be potential facade points, which are good indicators for the proximity but not necessarily for the penalty term.

Illustration with a real building

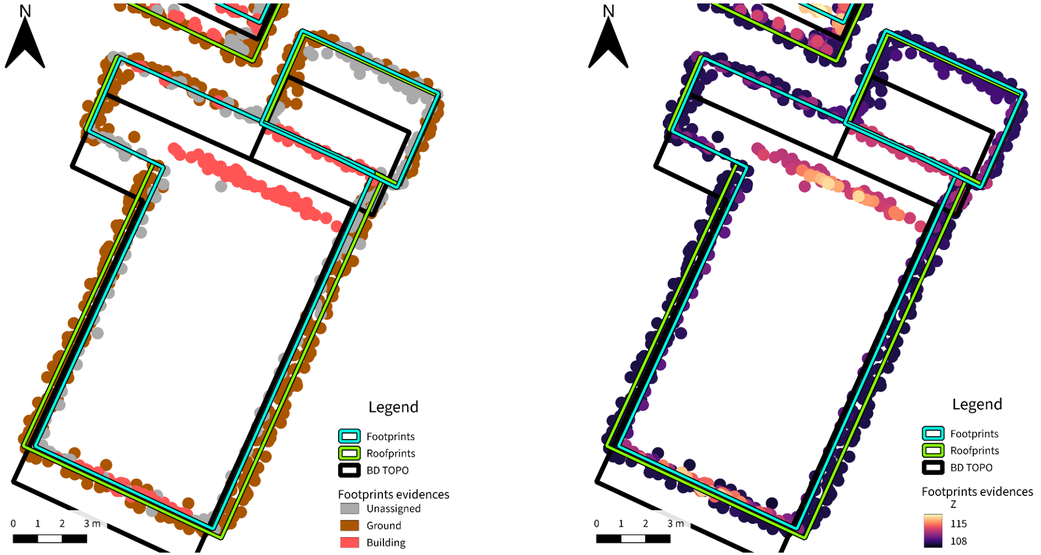


Figure 32: Illustration of the computed footprints.

Example where it works well:

- the footprint is **properly aligned** with the façade points
- we can see **significant roof overhangs for façades oriented west/east** and no significant roof overhangs for north/south, as expected from the orientation of the slanted roof
- the final footprint is **very close to the initial one from BD TOPO** in terms of shape, with a much better alignment on the point cloud

Illustration with a real building

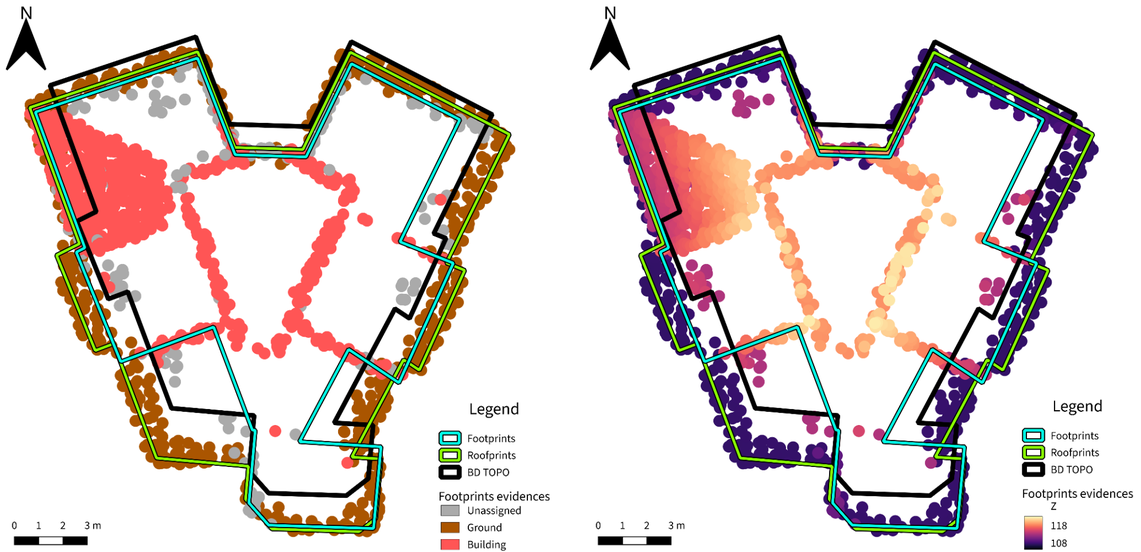


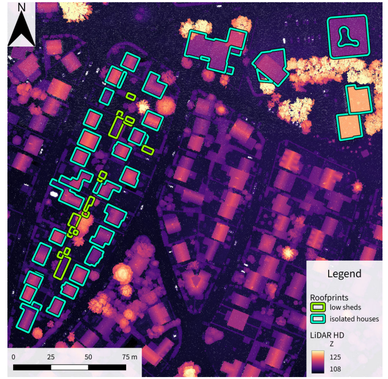
Figure 32: Illustration of the computed footprints.

Example where it doesn't work well:

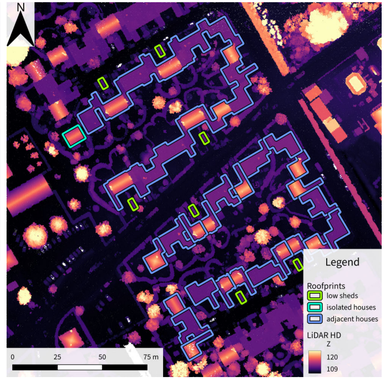
- many points were selected even though they are actually on the roof
- this resulted in some edges aligning on structures in the roof instead of on the façades

Validation

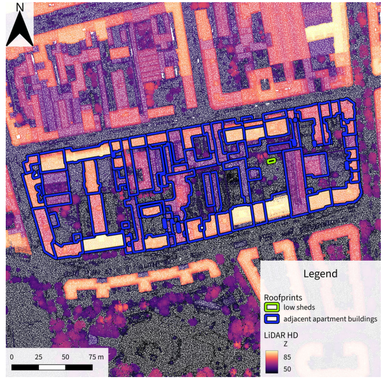
Validation dataset



(a) Area in Ozoir-la-Ferrière with isolated houses and small sheds.



(b) Area in Ozoir-la-Ferrière with mainly adjacent houses.



(c) Area in Paris with mainly adjacent blocks of flats.

Figure 32: Validation dataset.

We made a validation dataset for roofprints with buildings in four different categories (see Figure 21):

- 40 “isolated houses”: medium houses and buildings having up to a few storeys, isolated from the buildings around,
- 25 “adjacent houses”: blocks of adjacent and medium houses and buildings having up to a few storeys,
- 25 “low sheds”: low buildings in height, usually in gardens,
- 80 “adjacent blocks of flats”: blocks of adjacent and high buildings.

Validation results

Dataset	IoU	Chamfer (m)	Centroid (m)
BD TOPO	0.735	0.604	0.939
Rfpt 1 iter	0.842	**0.331**	**0.541**
Rfpt 2 iter	0.851	0.335	0.548
Rfpt 3 iter	**0.853**	0.342	0.566

(a) Whole dataset.

Dataset	IoU	Chamfer (m)	Centroid (m)
BD TOPO	0.776	0.557	0.846
Rfpt 1 iter	0.89	0.243	0.388
Rfpt 2 iter	0.902	0.226	0.362
Rfpt 3 iter	**0.905**	**0.22**	**0.354**

(b) Whole dataset except the “low sheds” category.

Dataset	IoU	Chamfer (m)	Centroid (m)
BD TOPO	0.497	0.876	1.477
Rfpt 1 iter	**0.562**	**0.842**	**1.426**
Rfpt 2 iter	0.557	0.97	1.631
Rfpt 3 iter	0.549	1.045	1.793

(c) Category “low sheds”.

Dataset	IoU	Chamfer (m)	Centroid (m)
BD TOPO	0.744	0.55	0.681
Rfpt 1 iter	0.888	0.165	0.215
Rfpt 2 iter	**0.935**	**0.114**	0.12
Rfpt 3 iter	**0.935**	**0.114**	**0.117**

(d) Category “isolated houses”.

Dataset	IoU	Chamfer (m)	Centroid (m)
BD TOPO	0.797	0.448	0.638
Rfpt 1 iter	0.901	0.196	0.309
Rfpt 2 iter	0.91	0.175	0.269
Rfpt 3 iter	**0.914**	**0.168**	**0.259**

(e) Category “adjacent houses”.

Dataset	IoU	Chamfer (m)	Centroid (m)
BD TOPO	0.785	0.595	0.994
Rfpt 1 iter	**0.888**	0.297	**0.499**
Rfpt 2 iter	0.882	0.298	0.512
Rfpt 3 iter	0.887	**0.289**	0.503

(f) Category “adjacent blocks of flats”.

Figure 32: Average metrics over different subsets of the validation dataset.

The different metrics on the validation dataset show that the do improve the outlines by a lot in all categories except “low sheds” where the results are bad for a few buildings and from correct to very good for the rest.

Conclusion

Answers to the research questions

- **Roofprints:**
  - High vertical gaps with neighbours
  - Complex energy combining points, edge lengths and inward directions

Answers to the research questions

- **Roofprints**:
  - High vertical gaps with neighbours
  - Complex energy combining points, edge lengths and inward directions
- **Footprints**:
  - Below the roof reconstructed with the roofprint
  - Energy combining façade and ground points

Answers to the research questions

- **Roofprints**:
  - High vertical gaps with neighbours
  - Complex energy combining points, edge lengths and inward directions
- **Footprints**:
  - Below the roof reconstructed with the roofprint
  - Energy combining façade and ground points
- **Polygon deformation**:
  - Incremental algorithm
  - Preserve the quality, validity and adjacencies of the polygons

Answers to the research questions

- **Roofprints**:
  - High vertical gaps with neighbours
  - Complex energy combining points, edge lengths and inward directions
- **Footprints**:
  - Below the roof reconstructed with the roofprint
  - Energy combining façade and ground points
- **Polygon deformation**:
  - Incremental algorithm
  - Preserve the quality, validity and adjacencies of the polygons
- All combined in a **single pipeline** generating **coherent roofprints and footprints** and allowing to estimate roof overhangs

Future work

- **Polygon deformation**:
  - Continuity instead of binary decision on shifting or not each edge
  - Preserve shared vertices

Future work

- **Polygon deformation**:
  - Continuity instead of binary decision on shifting or not each edge
  - Preserve shared vertices
  - More complex operations: rotations of edges, insertion/deletion of vertices

Future work

- **Polygon deformation**:
  - Continuity instead of binary decision on shifting or not each edge
  - Preserve shared vertices
  - More complex operations: rotations of edges, insertion/deletion of vertices
- **Roofprints** and **footprints**:
  - Classification of roofs and facades, but also balconies, chimneys, etc
  - More robust decision-making process for footprints due to the low amount of evidences

Future work

- **Polygon deformation**:
  - Continuity instead of binary decision on shifting or not each edge
  - Preserve shared vertices
  - More complex operations: rotations of edges, insertion/deletion of vertices
- **Roofprints** and **footprints**:
  - Classification of roofs and facades, but also balconies, chimneys, etc
  - More robust decision-making process for footprints due to the low amount of evidences
- **Other ideas**:
  - More complex representations than 2D partitions to combine roofprints, footprints, balconies, etc
  - Use aerial images to produce roofprints with clean angles

Thank you for your attention!

Link to the slides:

https://alexandre-bry.github.io/MSc_Thesis-Report/slides_pages/2026_06_12-A3.html

alexandre.bry@ign.fr, abry@tudelft.nl

The end

—

References

Biljecki, F., Ledoux, H., & Stoter, J. (2016). An improved LOD specification for 3D building models. *Computers, Environment and Urban Systems*, *59*, 25–37. https://doi.org/10.1016/j.compenvurbsys.2016.04.005

The end

—

Appendix

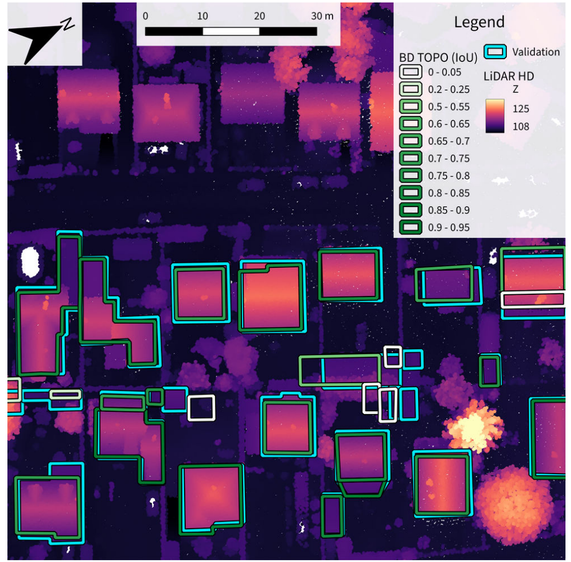


Figure 27: BD TOPO.

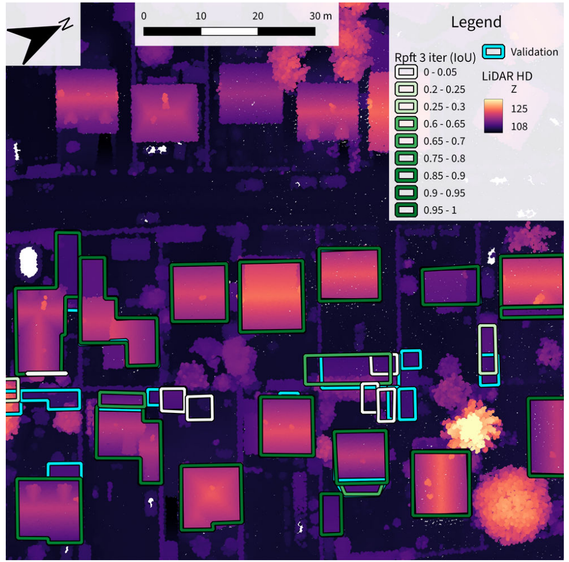


Figure 28: Roofprints after 3 iterations.

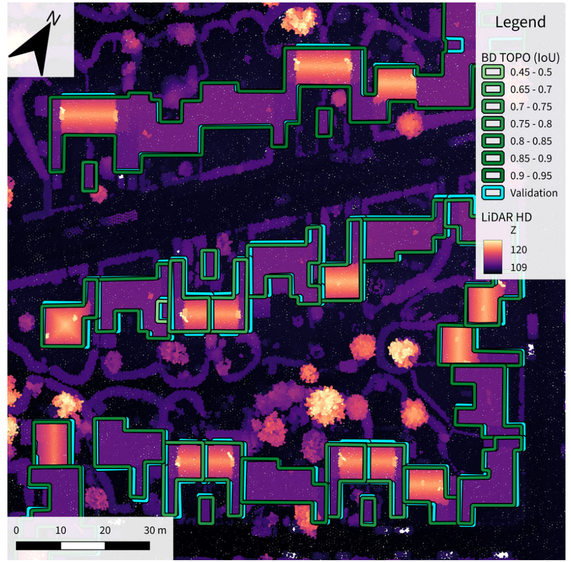


Figure 29: BD TOPO.

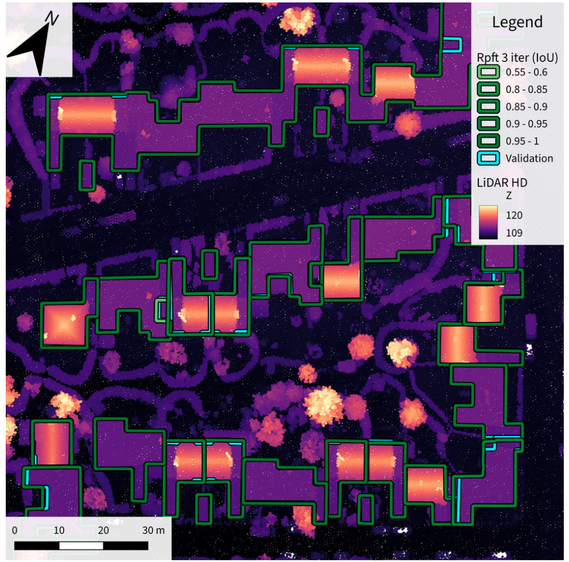


Figure 30: Roofprints after 3 iterations.

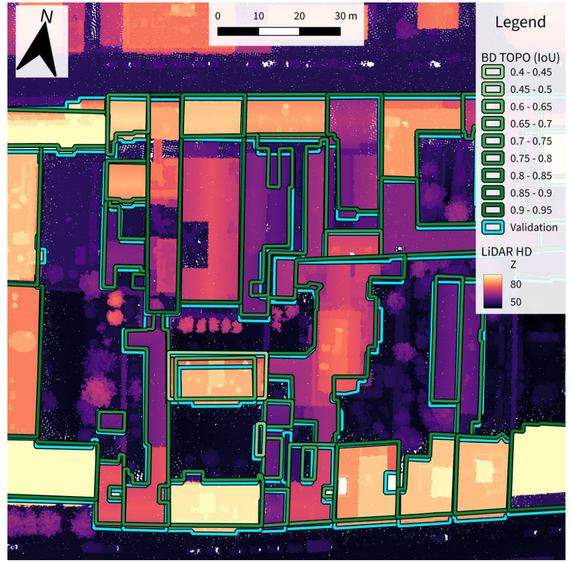


Figure 31: BD TOPO.

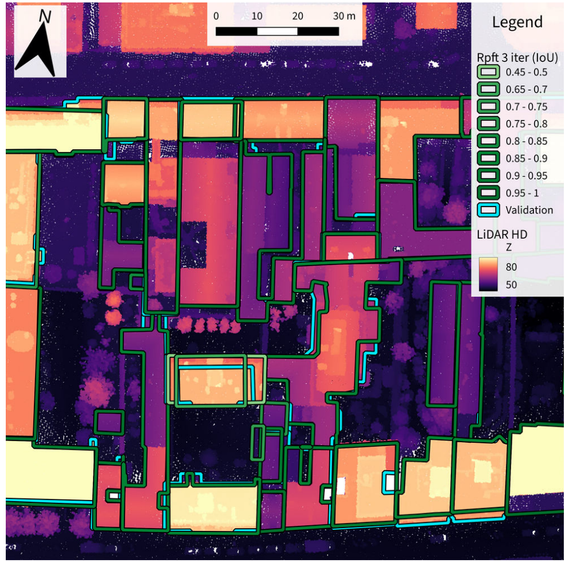


Figure 32: Roofprints after 3 iterations.