# A voxel-based methodology to detect (clustered) outliers in aerial LiDAR point clouds

**P5** **MSc thesis Geomatics for the built environment**

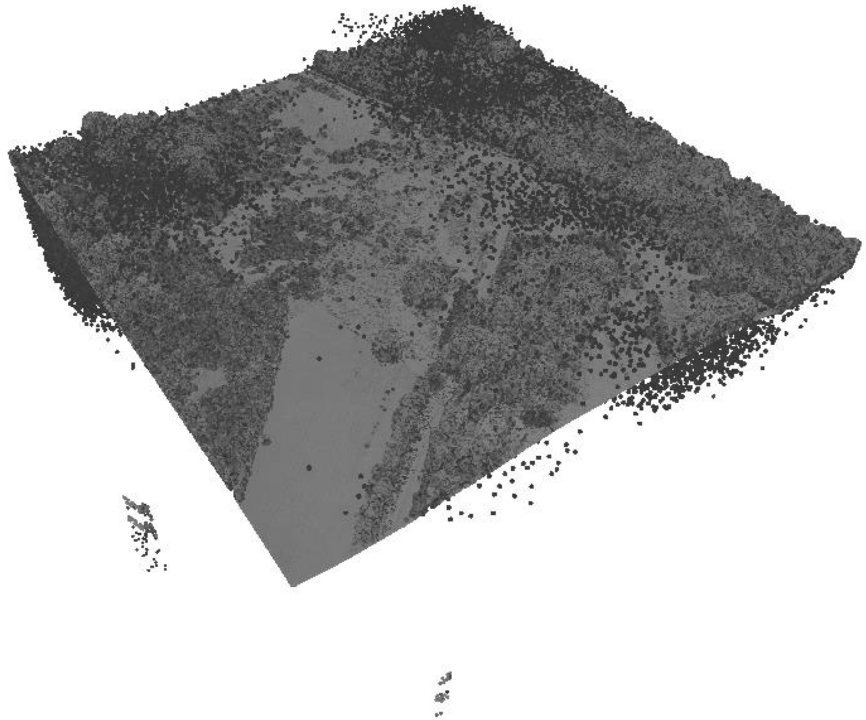
Simon Griffioen

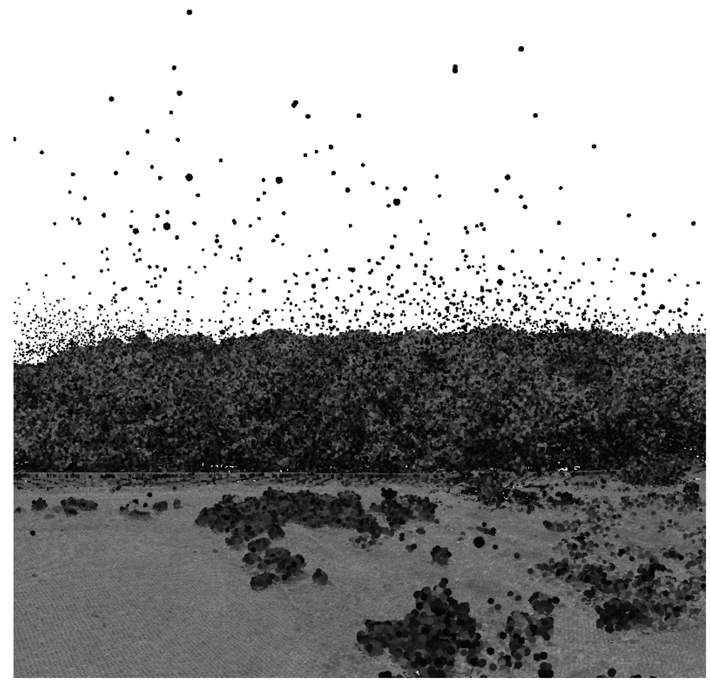
Mentors:
- Ravi Peters
- Hugo Ledoux
- Maarten Pronk (Deltares)

Co-reader:
- Martijn Meijers

TU Delft

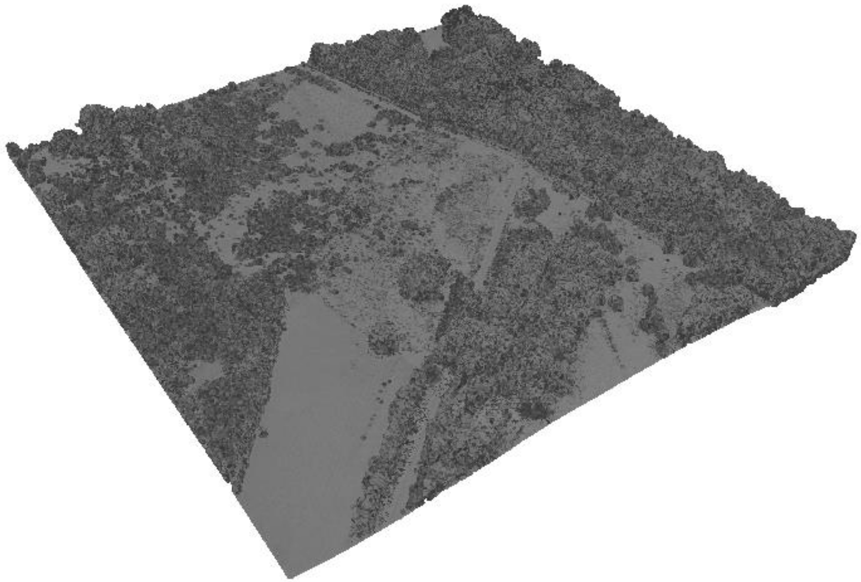
# Main goal of study:

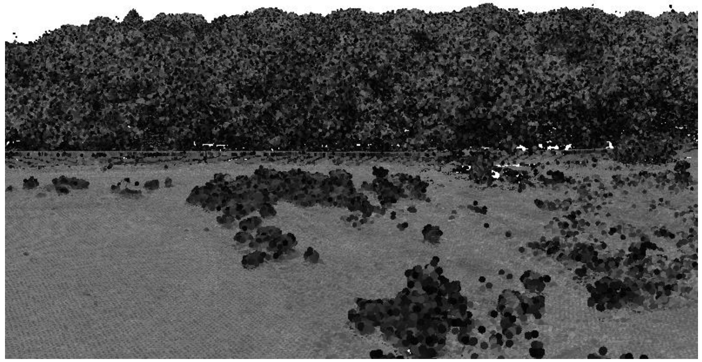




- From: raw aerial LiDAR point clouds

# Main goal of study:





- To: cleaned datasets without outliers

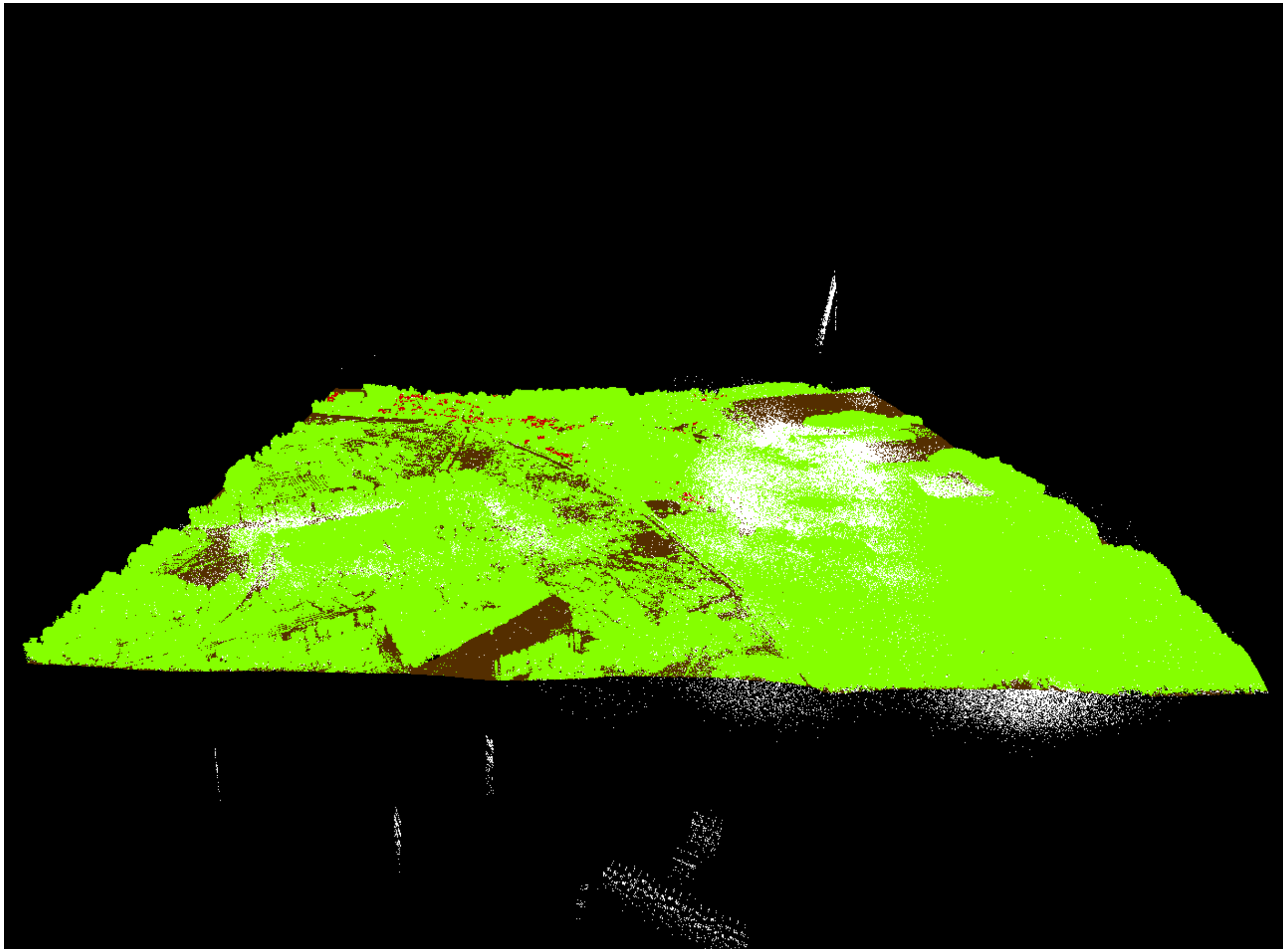
# How to do this?

- Research Motivation
- Related Work
- A Voxel-based Methodology
- Results & Quality Assessment
- Discussion & Future Work
- Conclusions

TU Delft

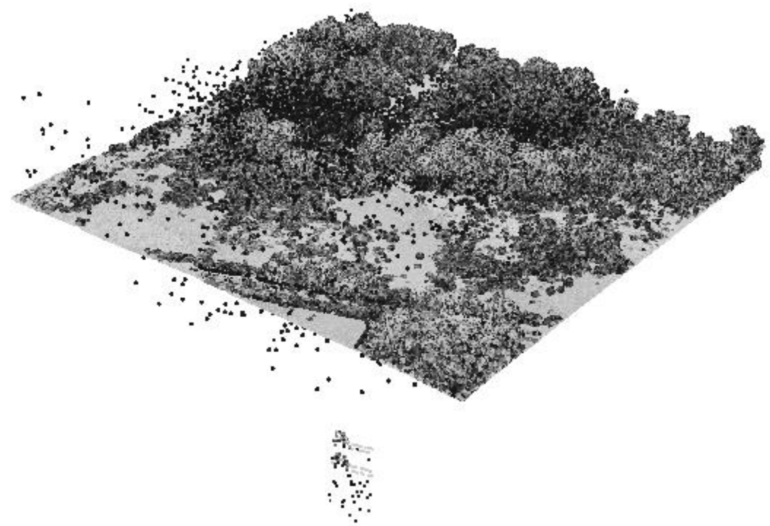
# Research motivation (1/2)

- Raw 3D point cloud data often includes errors (outliers);
- Outliers need to be removed to effectively analyze point cloud data;
- Deltares makes extensive use of point cloud data.

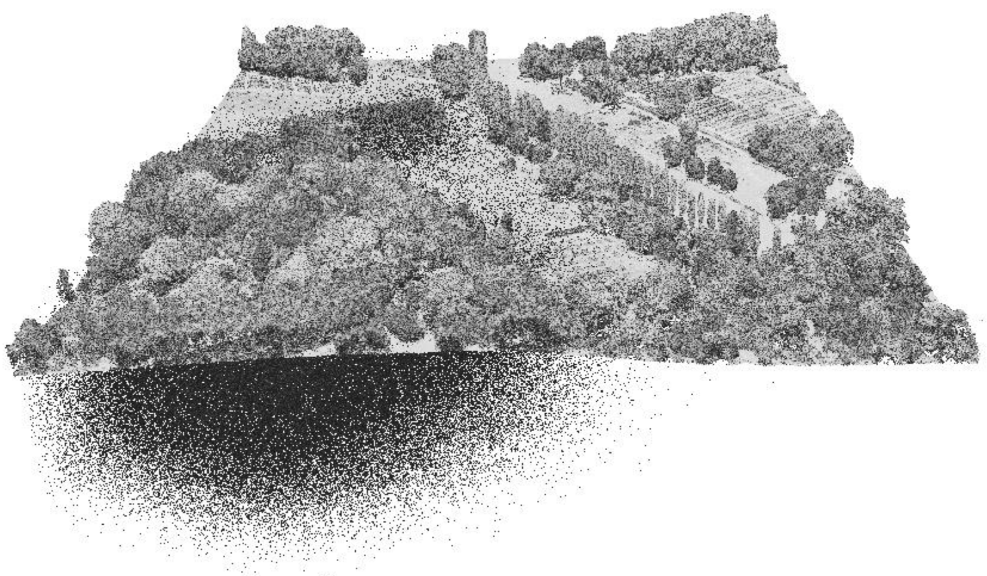


Data: Aerodata

**Type-1**: Isolated (high and low) outliers



**Type-2**: Clustered outliers



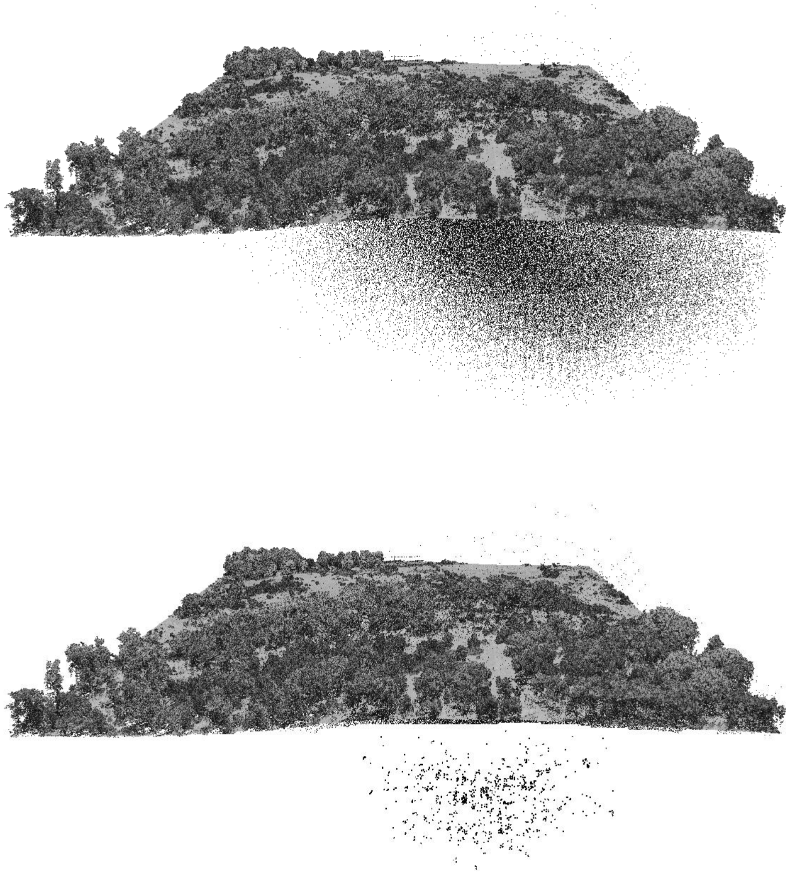
**Type-3**: Randomly scattered with high and low densities

Data: Aerodata

TU Delft

# Research motivation (2/2)
# Existing tools have limitations

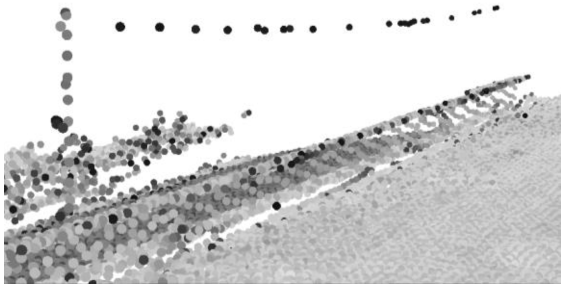
- Can only detect isolated points (type-1)
- Fail to detect clusters of outliers (type-2,  -3)
- Can remove features with low densities

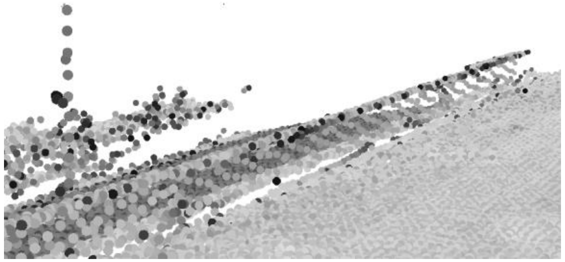


Example: LAStools/lasnoise

# Research motivation (2/2)
# Existing tools have limitations

- Can only detect isolated points (type-1)
- Fail to detect clusters of outliers (type-2,  -3)
- *Can remove features with low densities*





Example: LAStools/lasnoise

# Research scope and goals

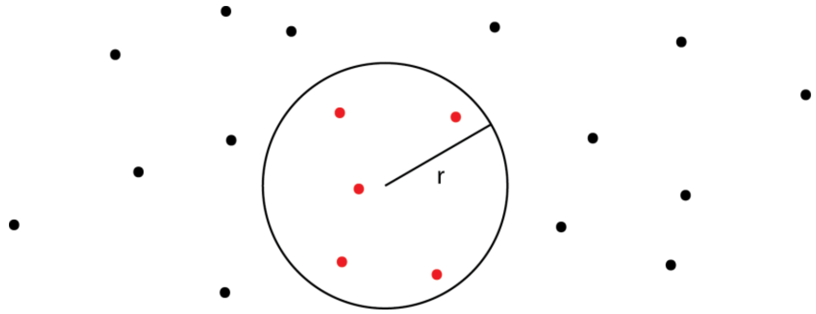
- Automatically detect outliers
  - Isolated, clustered and random
  - Using a voxel-based solution
  - Fully automatic

- In Aerial Laser Scanned (ALS) point clouds
  - Natural environments (vegetation, forest) & urban
  - Terrestrial/Mobile Laser Scanned data is not considered

- Scalability
  - Outperform existing tools in terms of accuracy, not speed
  - How to handle massive datasets (>100MM points)?

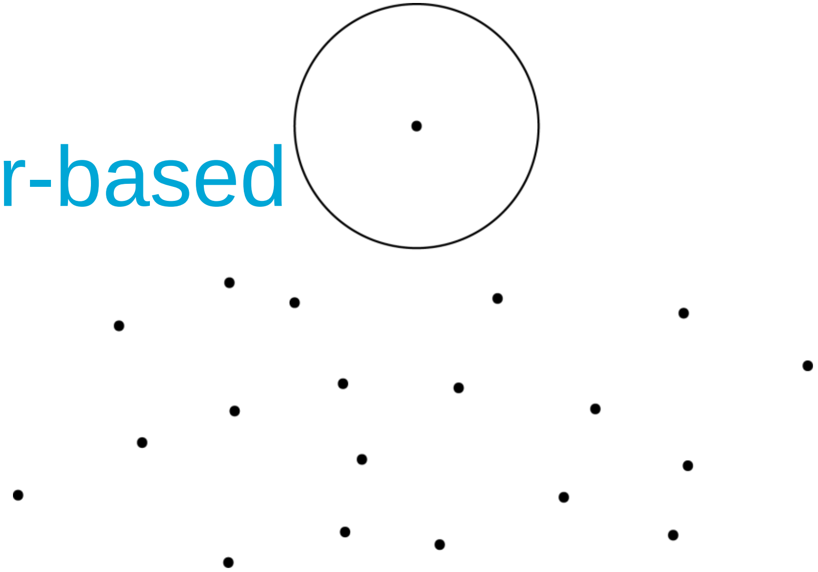
# Related work:
# Neighborhood- vs. Cluster-based

1. **Local Neighborhood-based**
   - Density-based
   - Distance-based
   - Mathematical morphology




- Works well for isolated outliers
- Trade-off between false positives and true positives
- Only considers geometric features

# Related work: Neighborhood- vs. Cluster-based



1. **Local Neighborhood-based**
   - Density-based
   - Distance-based
   - Mathematical morphology

- Works well for isolated outliers
- Trade-off between false positives and true positives
- Only considers geometric features

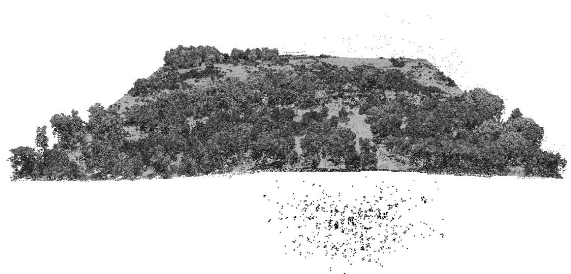
TUDelft

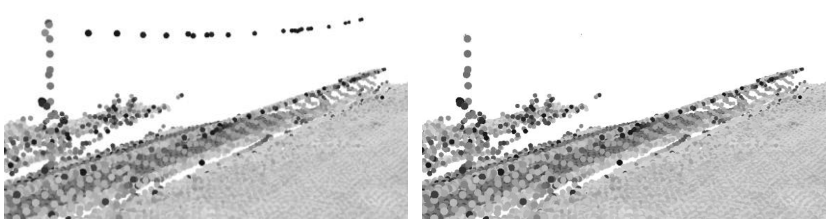
# Related work: Neighborhood- vs. Cluster-based

1. **Local Neighborhood-based**
   - Density-based
   - Distance-based
   - Mathematical morphology

- Works well for isolated outliers
- Trade-off between false positives and true positives
- Only considers geometric features

→ Problems handling clustered outliers (type-2 and -3) and
→ Features may be locally indistinguishable from outliers





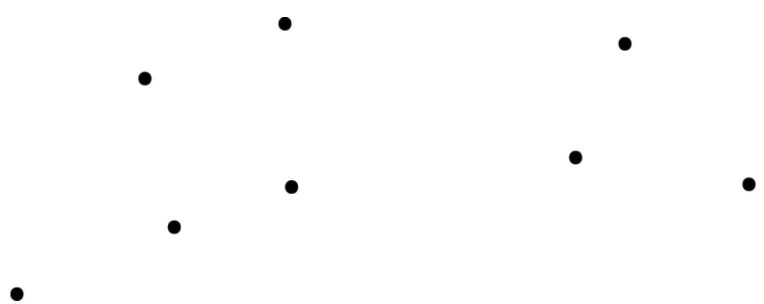
# Related work:
# Neighborhood- vs. Cluster-based

**2. Cluster/graph-based**

- Can detect clustered outliers

E.g.

- Delaunay Triangulation → Connected Components (Arge et al., 2010)
- Delaunay Triangulation → Edge pruning (Sotoodeh, 2007)



TU Delft

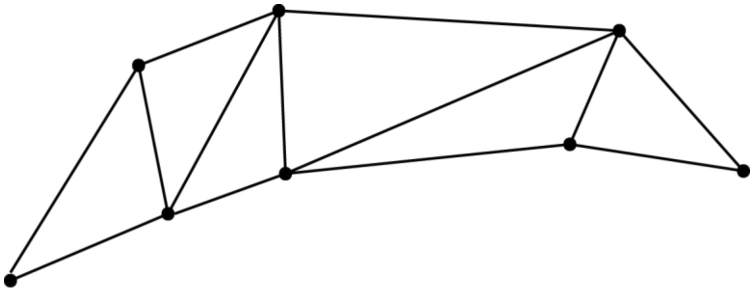
# Related work: Neighborhood- vs. Cluster-based

## 2. Cluster/graph-based

- Can detect clustered outliers

E.g.

- Delaunay Triangulation → Connected Components (Arge et al., 2010)
- Delaunay Triangulation → Edge pruning (Sotoodeh, 2007)



TU Delft

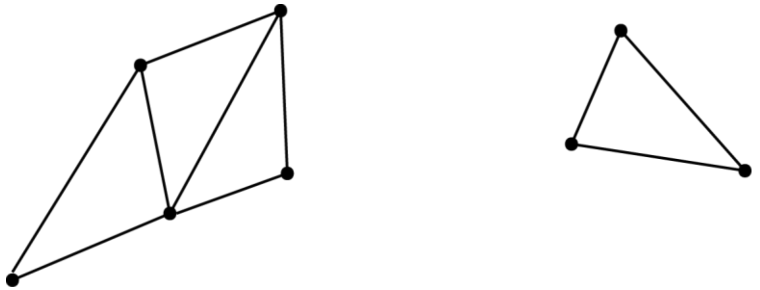
# Related work:
# Neighborhood- vs. Cluster-based

**2. Cluster/graph-based**

- Can detect clustered outliers

E.g.

- Delaunay Triangulation → Connected Components (Arge et al., 2010)
- Delaunay Triangulation → Edge pruning (Sotoodeh, 2007)



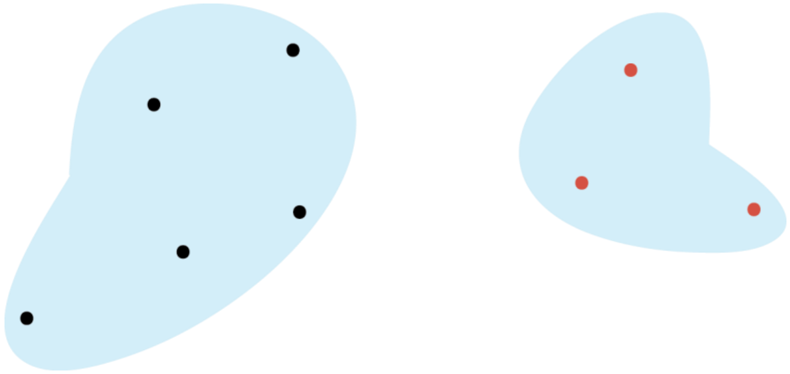
# Related work:
# Neighborhood- vs. Cluster-based

**2. Cluster/graph-based**

- Can detect clustered outliers

E.g.

- Delaunay Triangulation → Connected Components (Arge et al., 2010)
- Delaunay Triangulation → Edge pruning (Sotoodeh, 2007)

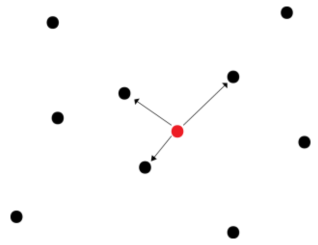


# Related work:
## Group-based vs. Point-based

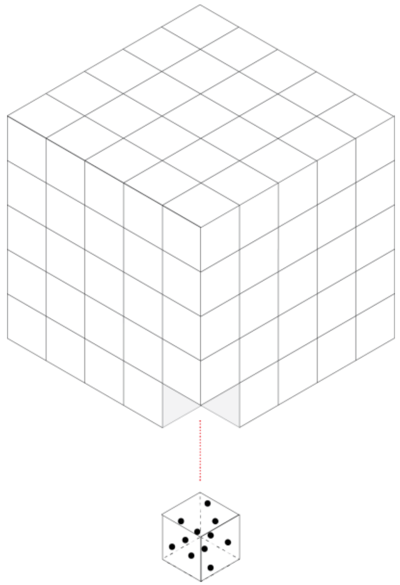
- Point-wise
  – Compute features for every point, e.g. k-nn

- Group-based
  – Segment points prior to feature extraction, e.g. voxels

K-nn:



Voxels:

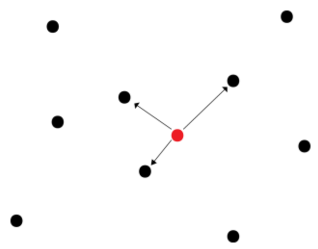


TU Delft

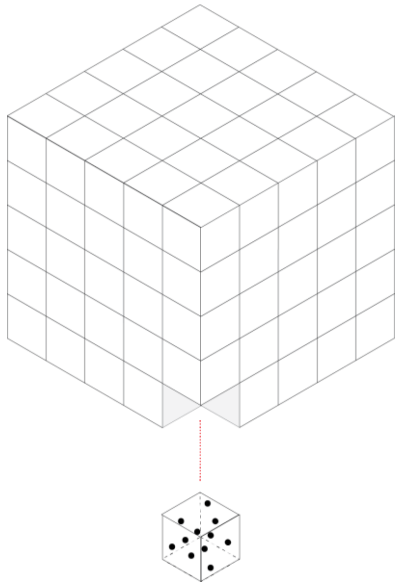
# Related work:
## Group-based vs. Point-based

- Point-wise
  - Compute features for every point, e.g. k-nn
  - Imply high computation load
- Group-based
  - Segment points prior to feature extraction, e.g. voxels
  - Speed up point cloud processing

K-nn:



Voxels:



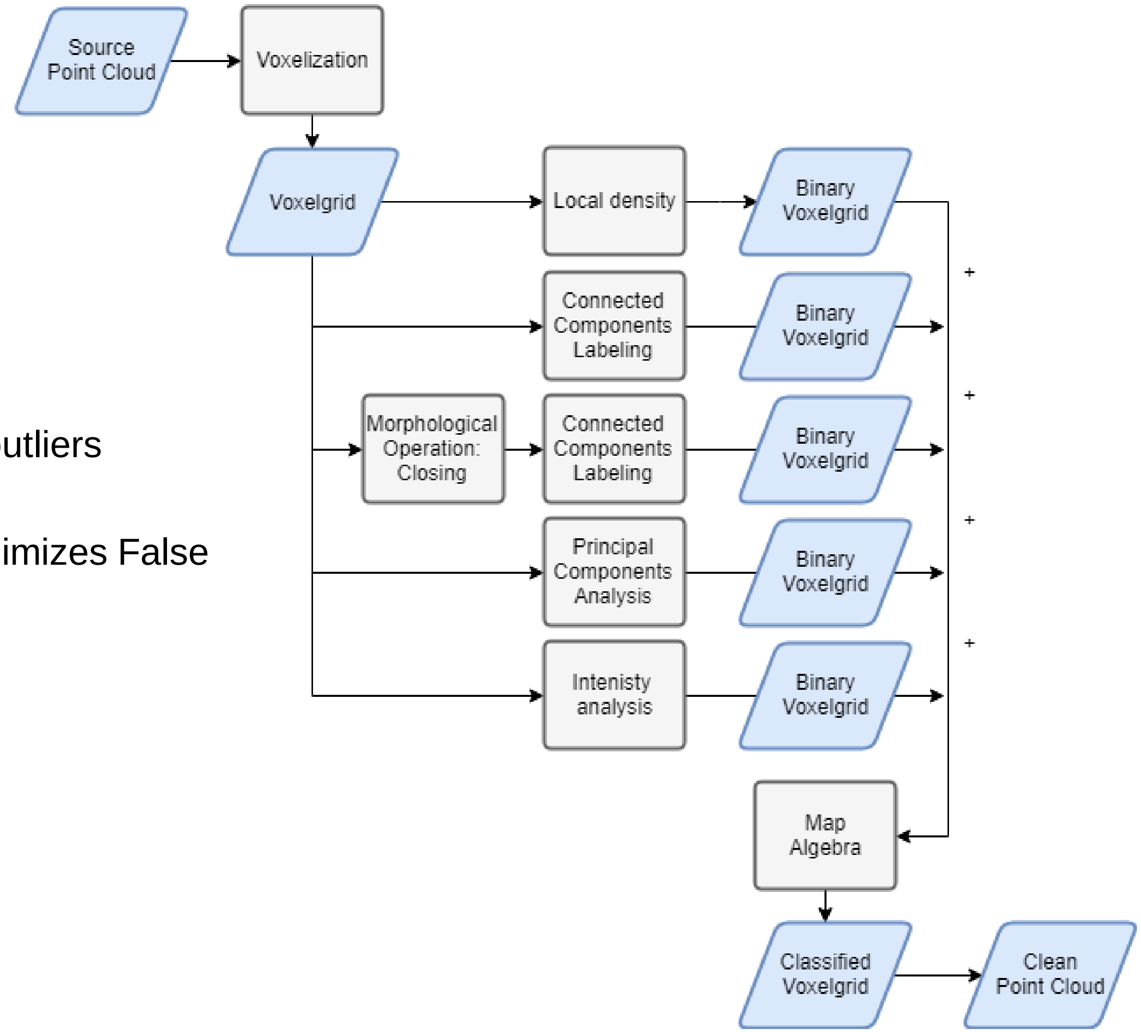
# Conclusions Literature Study

- Detect clustered outliers
- Keep features intact
- Group-based feature extraction
- Potential LiDAR attributes
- Trade-off between TP and FP

# Conclusions Literature Study

- Detect clustered outliers → **cluster-based approach**
- Keep features intact → **adjacency/connectivity**
- Group-based feature extraction → **voxels**
- Potential LiDAR attributes → **intensity analysis**
- Trade-off between TP and FP → **series of methods**

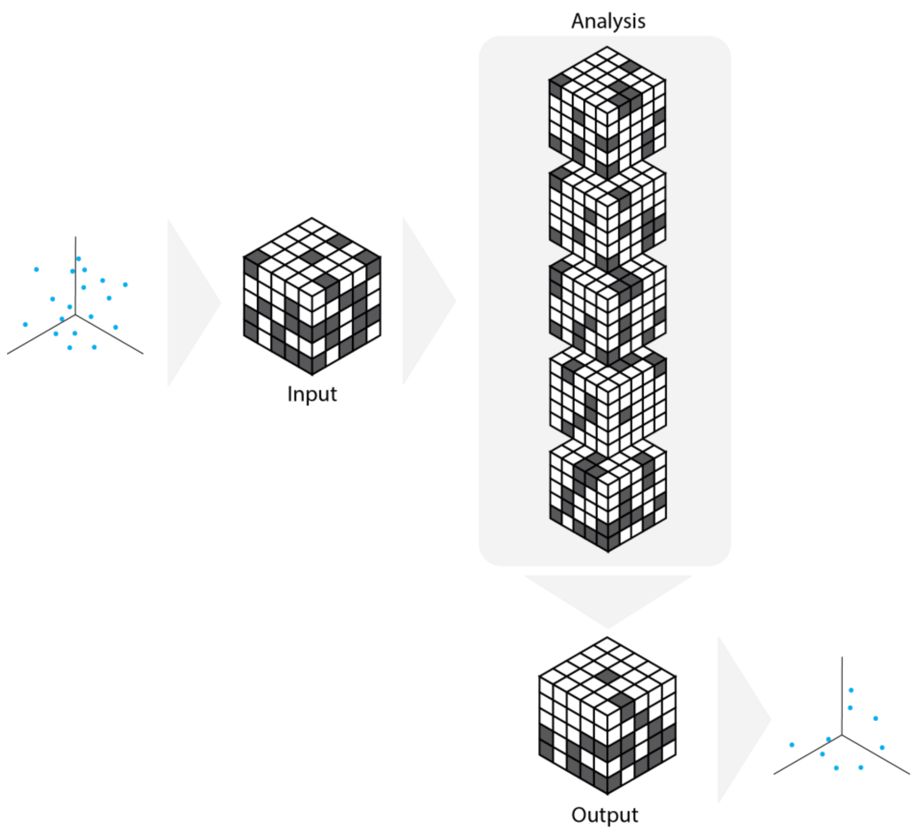
# Methodology



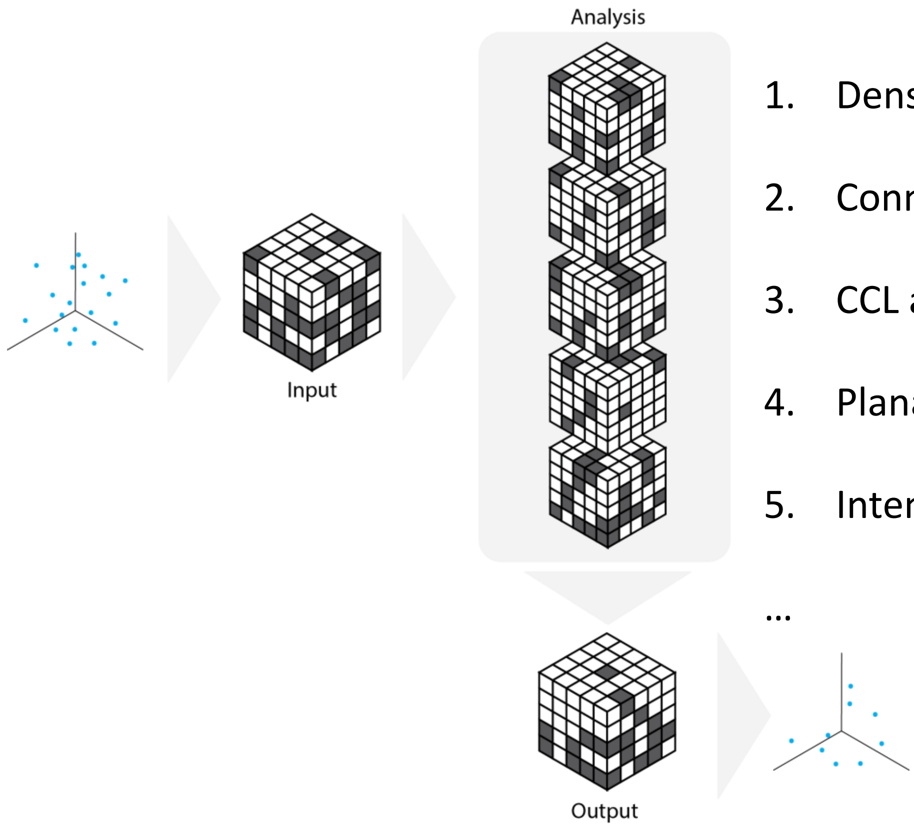

## <u>5 different operations</u>

- Each operation classifies outliers
- Series of operations to minimizes False Positives (FP)

# Voxel-based Solution



# Voxel-based Solution

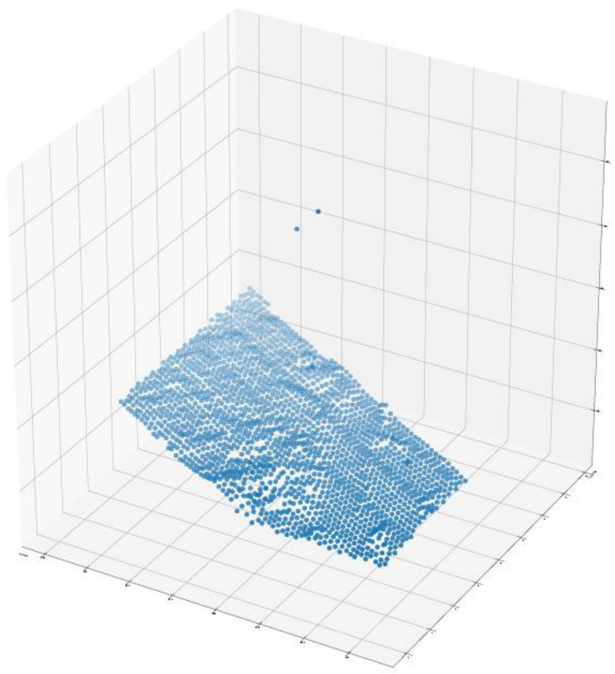



1. Density
2. Connected Components Labeling (CCL)
3. CCL after closing
4. Planarity
5. Intensity

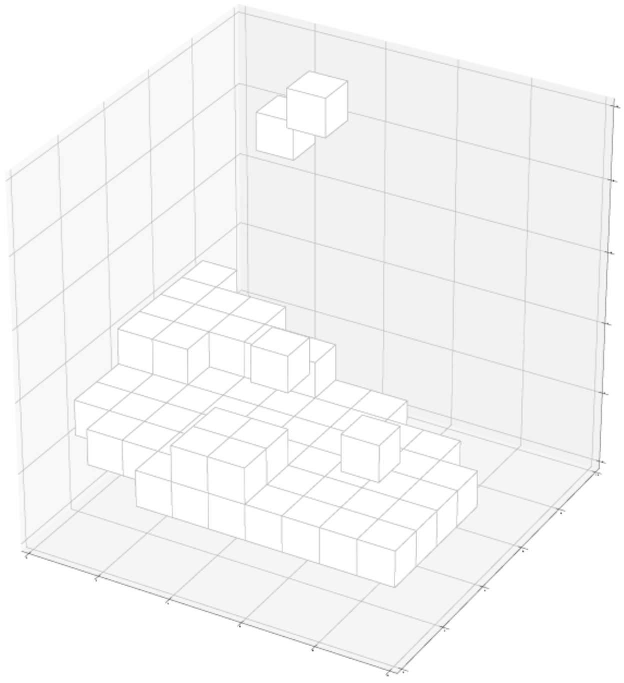
…

TU Delft

# Voxelization



Source point cloud

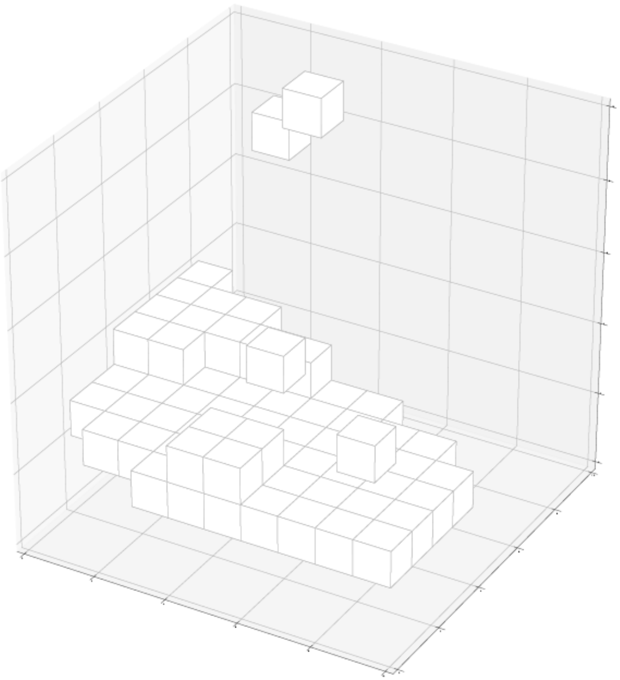


Binary 3D grid

# Voxelization

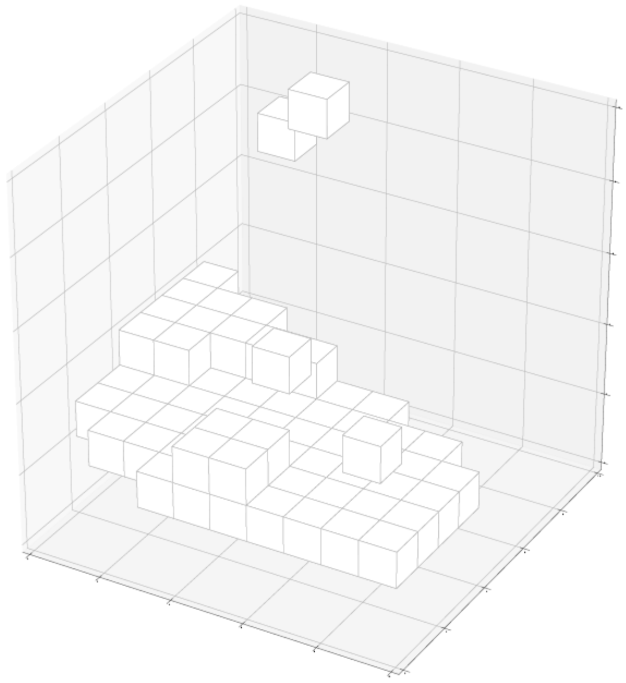
<u>Voxel size selection:</u>

1. The density of the point cloud
2. Size of features
3. Processing time

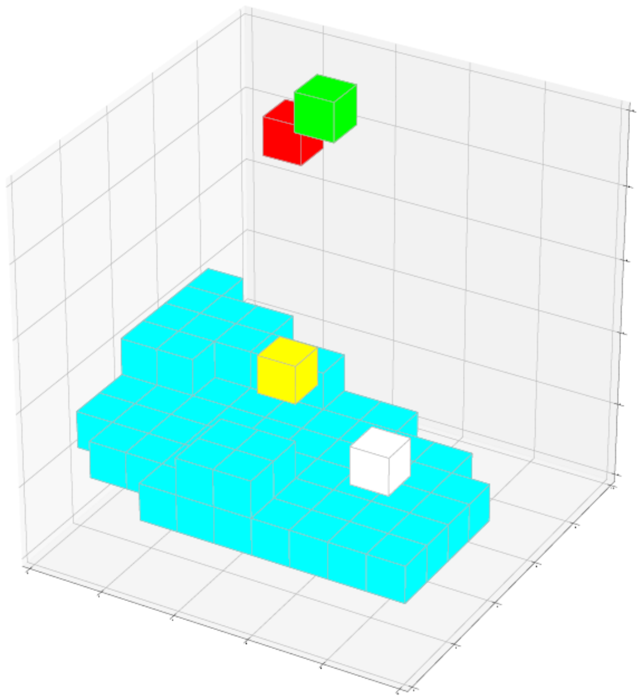


Binary 3D grid

# (2/5) Connected Components Labeling (CCL)



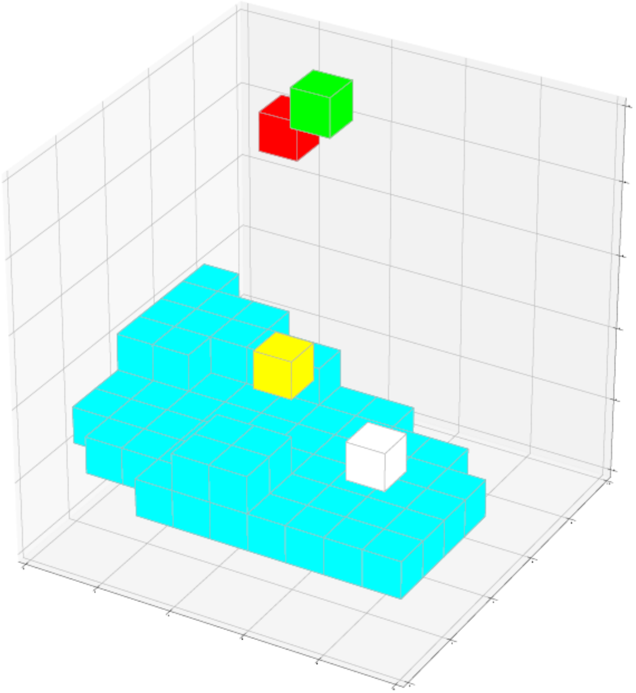
3D grid



Labelled regions

# (2/5) Connected Components Labeling (CCL)

1. Find largest connected component
2. Classify all points not in largest component as outlier

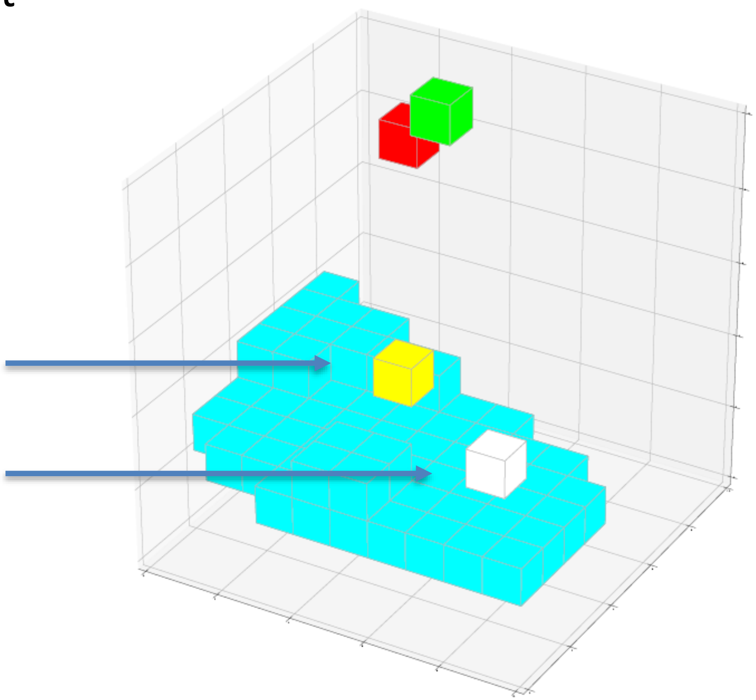


Labelled regions

# (2/5) Connected Components Labeling (CCL)

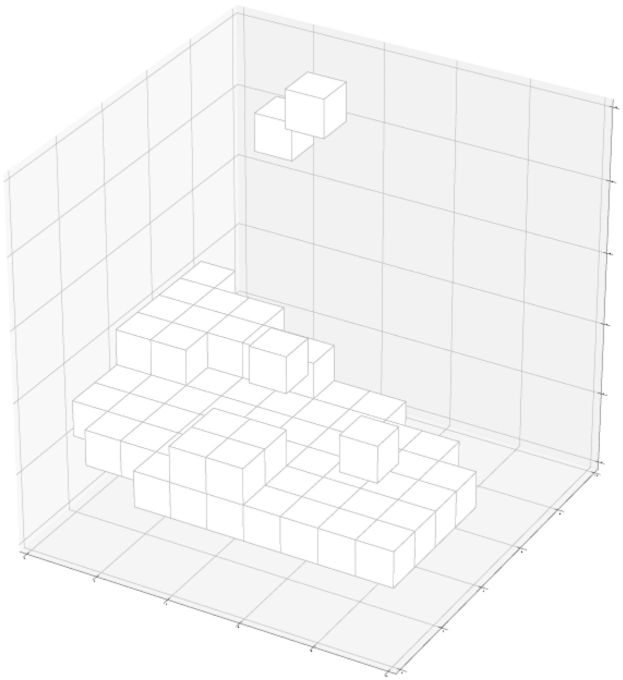
1. Find largest connected component
2. Classify all points not in largest component as outlier

Outliers?

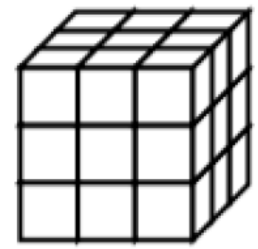


Labelled regions

# (3/5) Closing---Morphological Operator



3D grid (B)

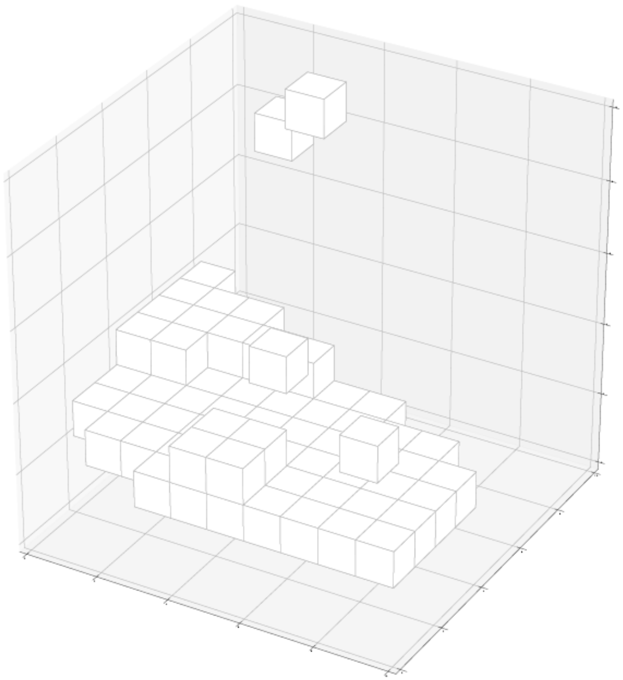


3 x 3 x 3 structuring element (S)

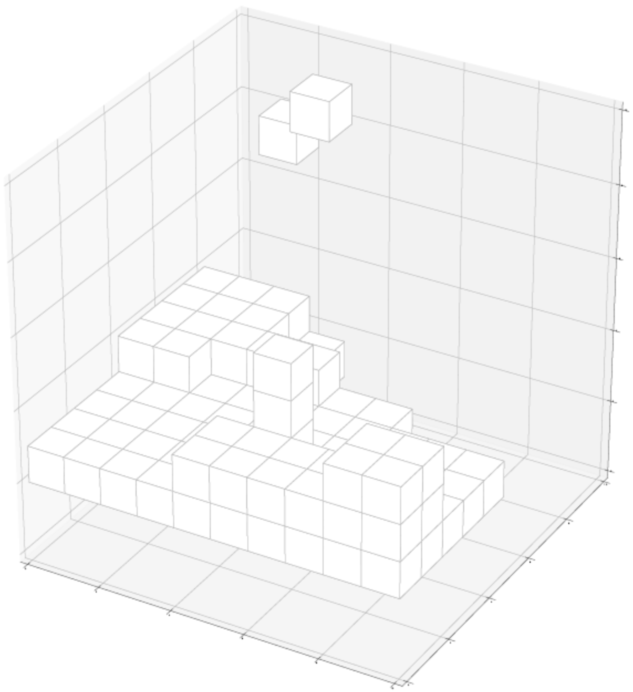
<u>Closing</u>
Dilation followed by erosion

$$B \bullet S = (B \oplus S) \ominus S$$

# (3/5) Closing---Morphological Operator

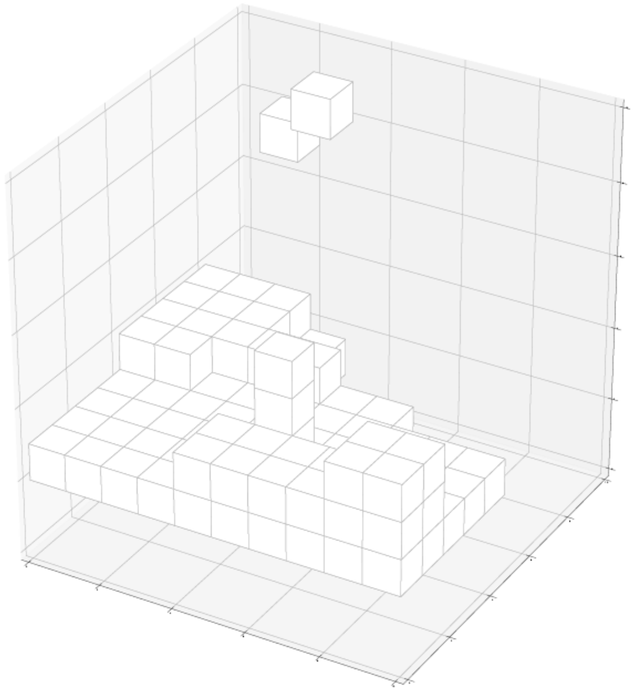


3D grid

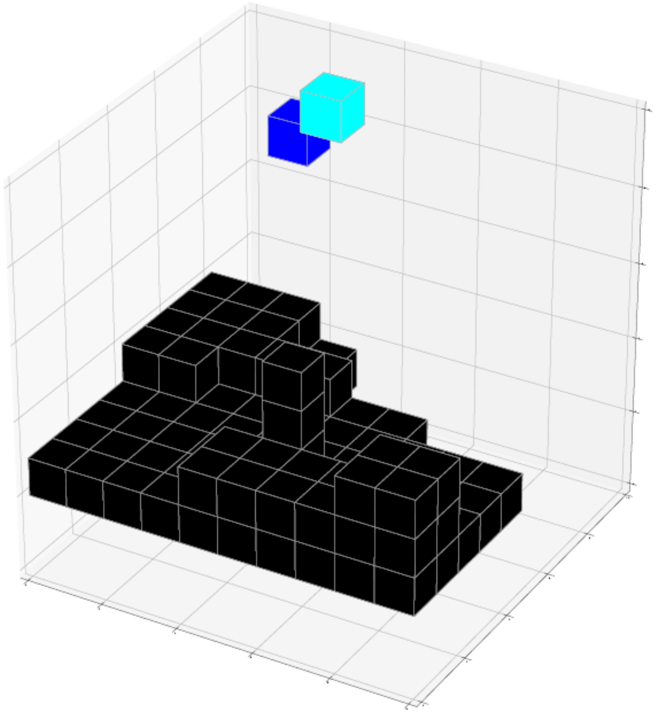


Closed grid

# (3/5) CCL after *closing*

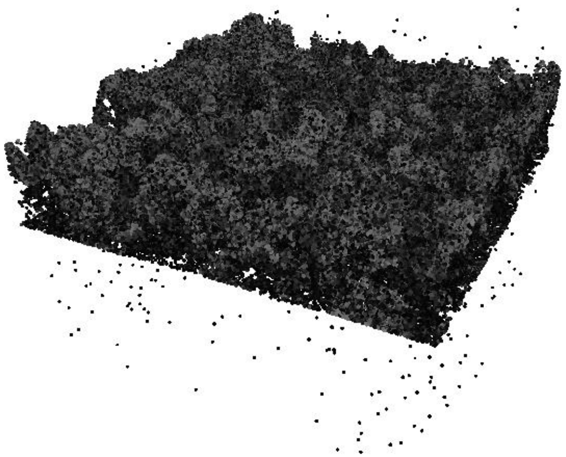


Closed grid

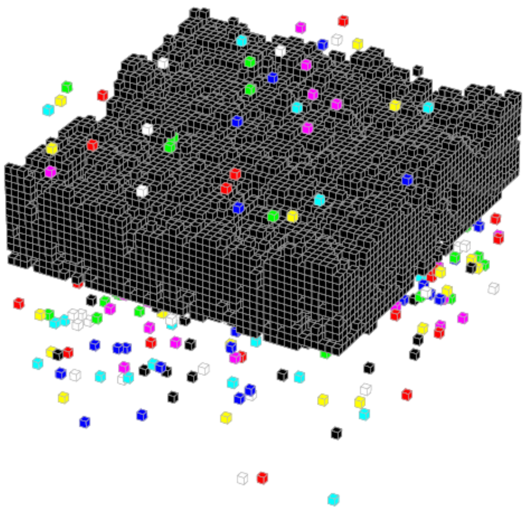


Labelled regions

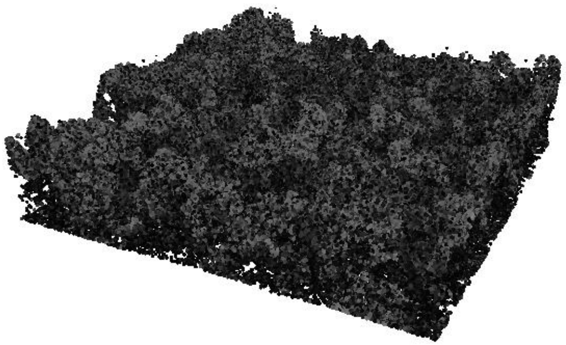
# Why CCL works



Source point cloud



Labelled regions

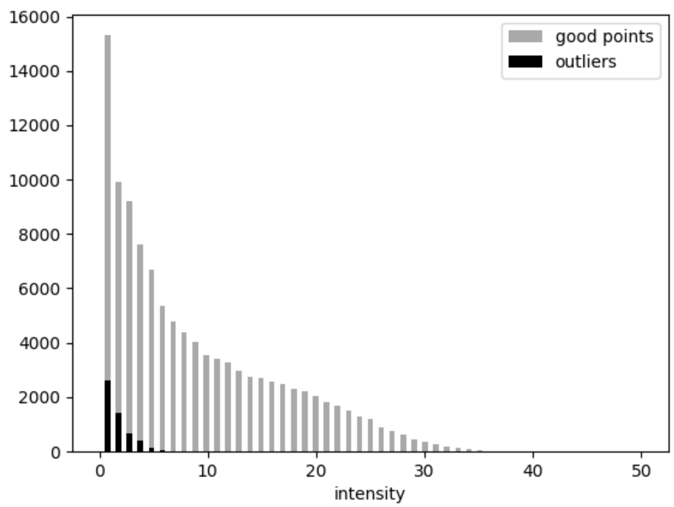


Cleaned point cloud

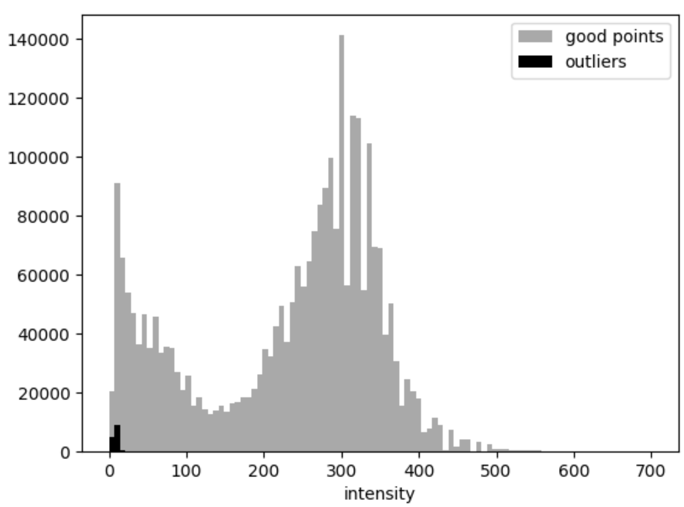
TU Delft

# (4/5) Intensity

- Detect good points---not outliers

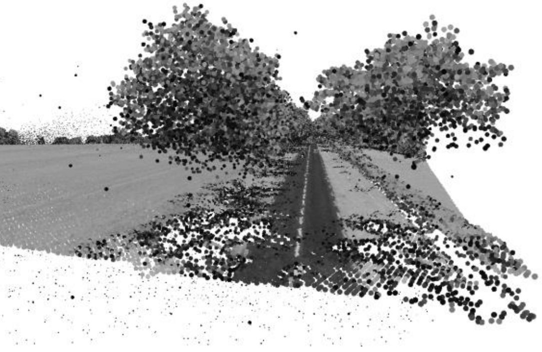



Data: Aerodata

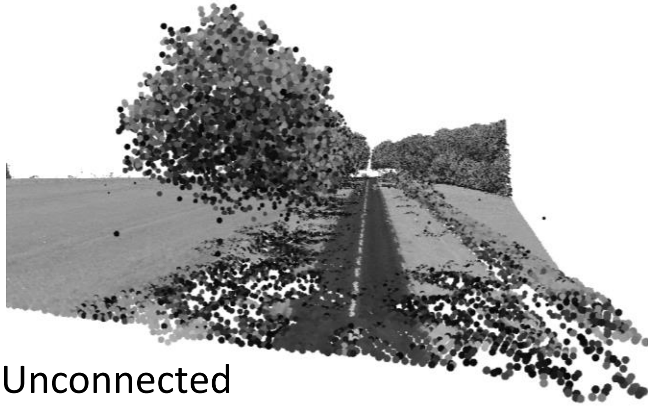



Data: Deltares

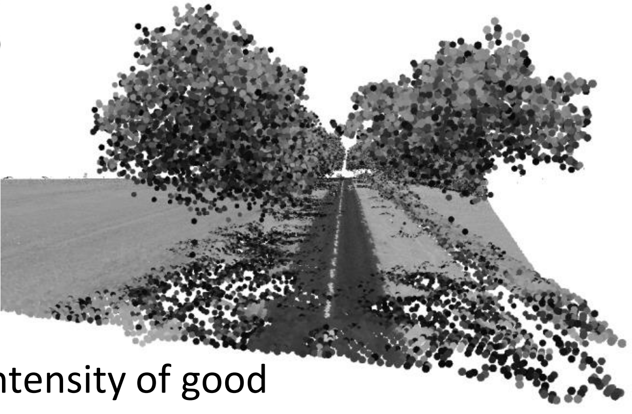
# Why this works



Raw data
(Aerodata)

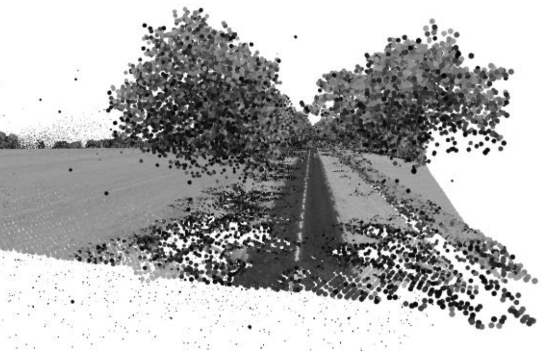


Unconnected
trees removed

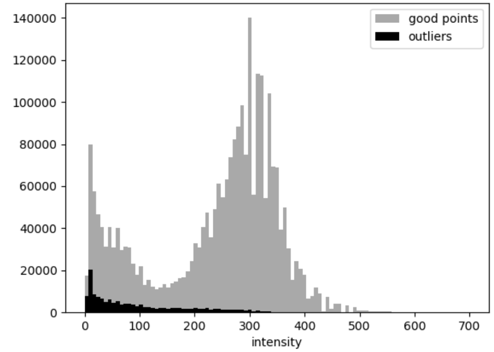


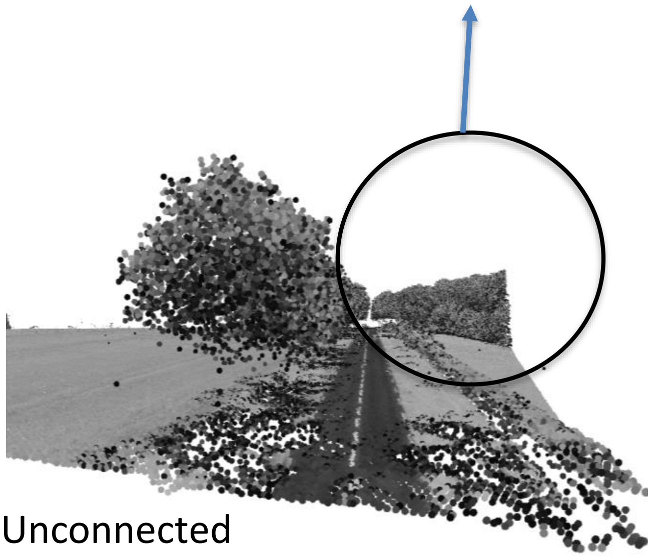
Intensity of good
points

# Why this works

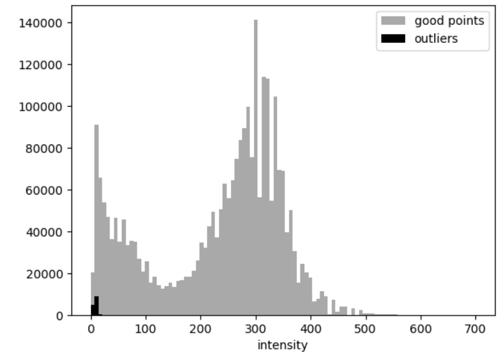


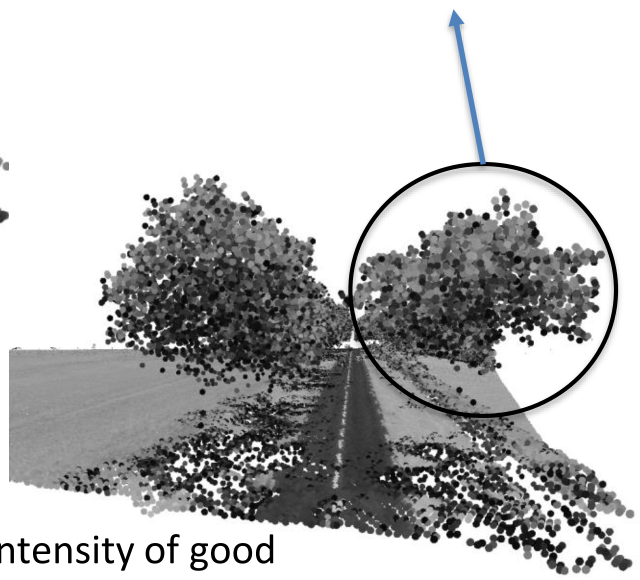
Raw data
(Aerodata)



Unconnected
trees removed



Intensity of good
points

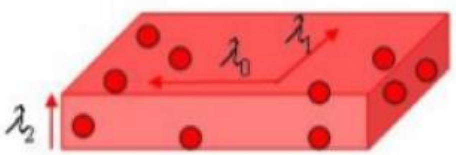
# (5/5) Planarity

- Outliers usually form a scattered region and rarely fit in a plane

$\lambda_0 \approx \lambda_1 \approx \lambda_2$

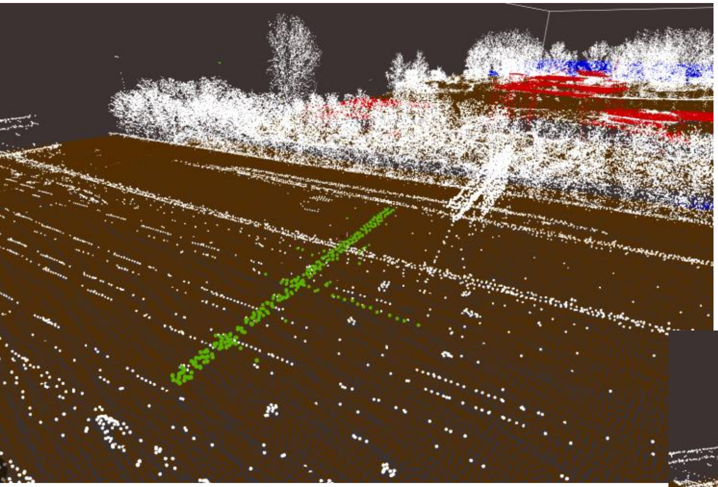
Scatter
= outlier

$\lambda_0 \approx \lambda_1 \gg \lambda_2$

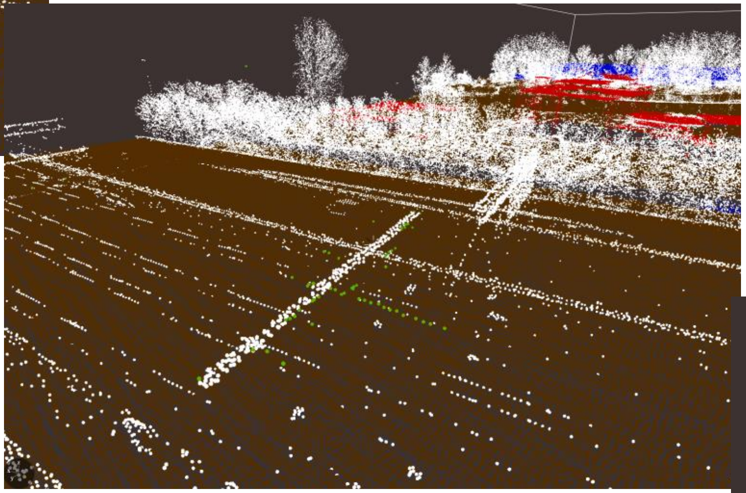



Planar
= no outlier

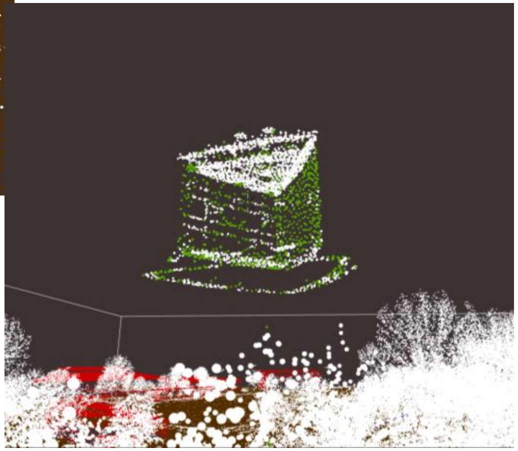
# Why this works



Unconnected street signs
Data: AHN3

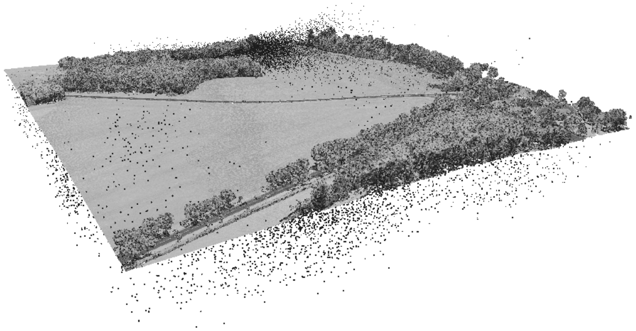


Signs are planar → no outlier

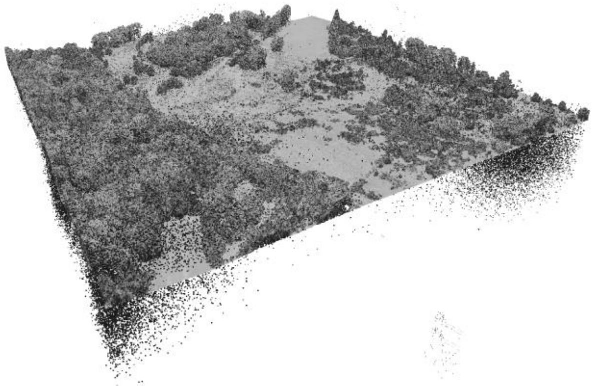


Planar features

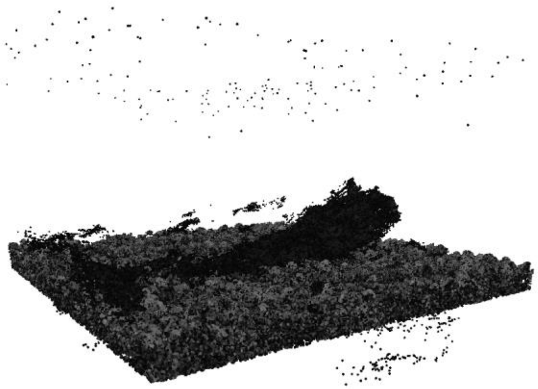
# Experiments: Datasets



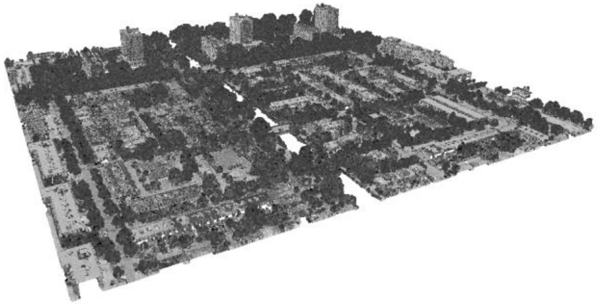
(A1) Aerodata



(A2) Aerodata



(B) Deltares



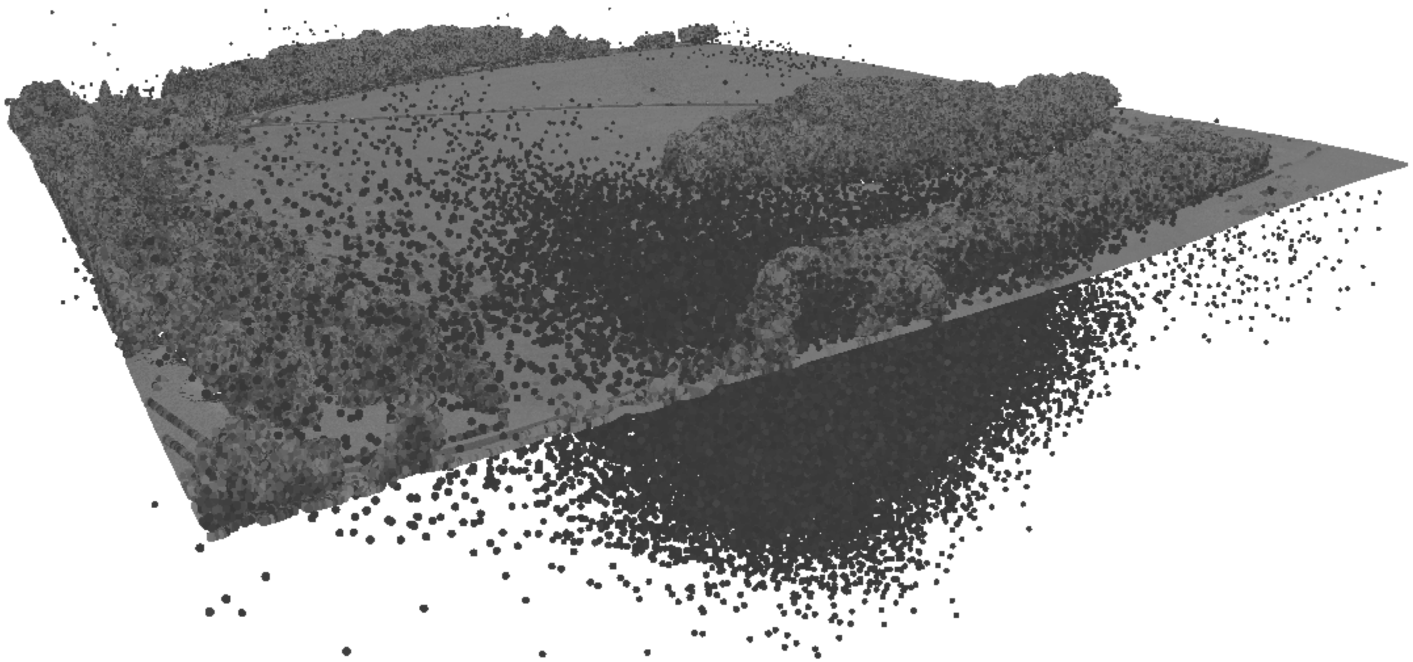
(C) AHN3

(D) Kadaster

# Datasets

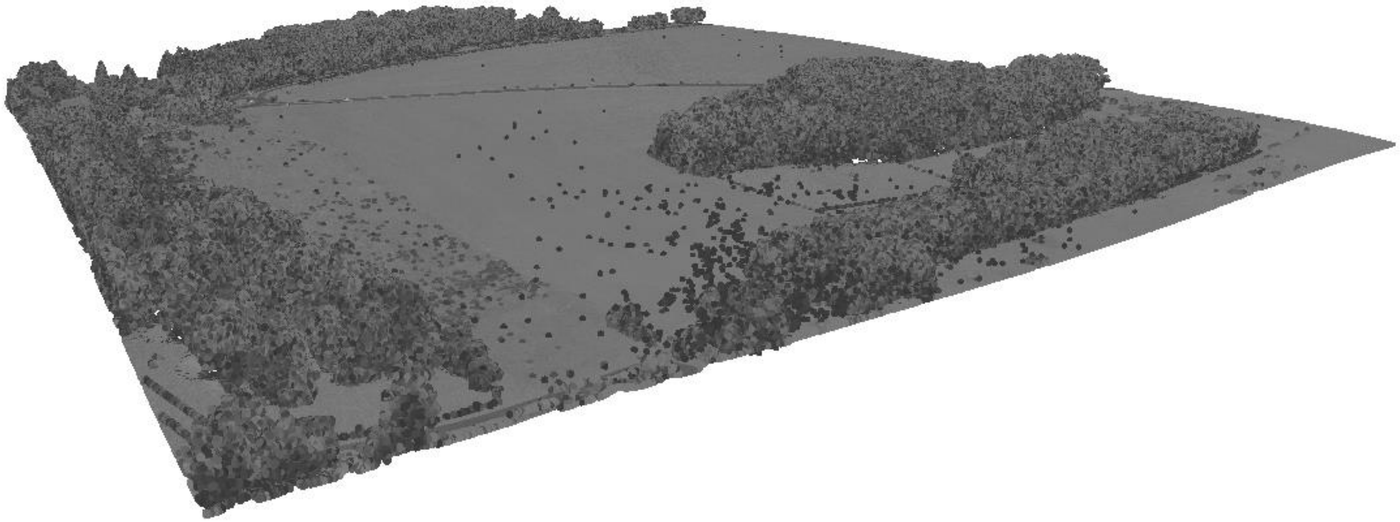
| | *Point cloud* | | | | |
|---|---|---|---|---|---|
| | A1 | A2 | B | C | D |
| *Source* | Aerodata | Aerodata | Deltares | AHN3 | Kadaster |
| *Technique* | ALS | ALS | ALS | ALS | DIM |
| *Area (km)* | 0.5 x 0.5 | 0.5 x 0.5 | 0.5 x 0.5 | 0.5 x 0.5 | 0,5 x 0,5 |
| *N points* | 5.7 mln | 8.2 mln | 1.7 mln | 4.7 mln | 5.5 mln |
| *Points per $m^2$* | 23 | 33 | 7 | 19 | 22 |
| *Outliers* | Many | Many | Many | None | None |
| *Ground truth* | Yes | Yes | No | No | No |
| *Environment* | Vegetation, built environment | Vegetation, built environment | Forest | Urban | Urban |

# Result A1



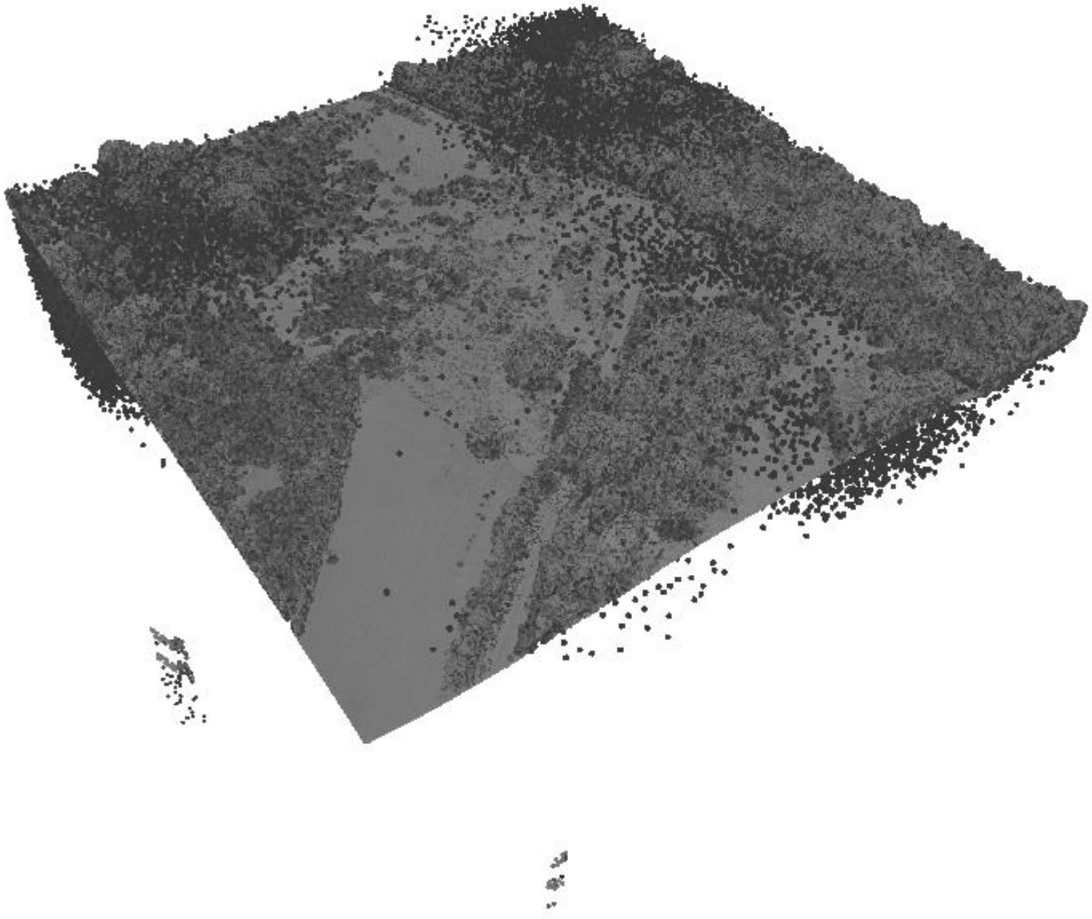
Data: Aerodata

# Result A1



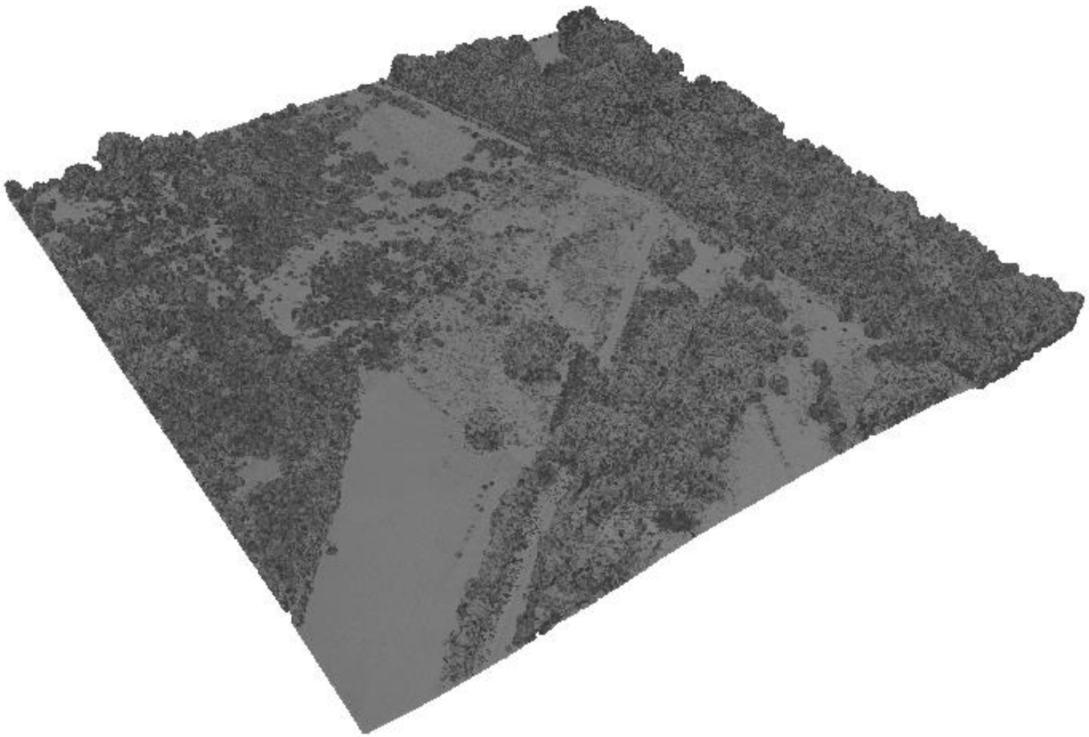
Data: Aerodata

# Results A2



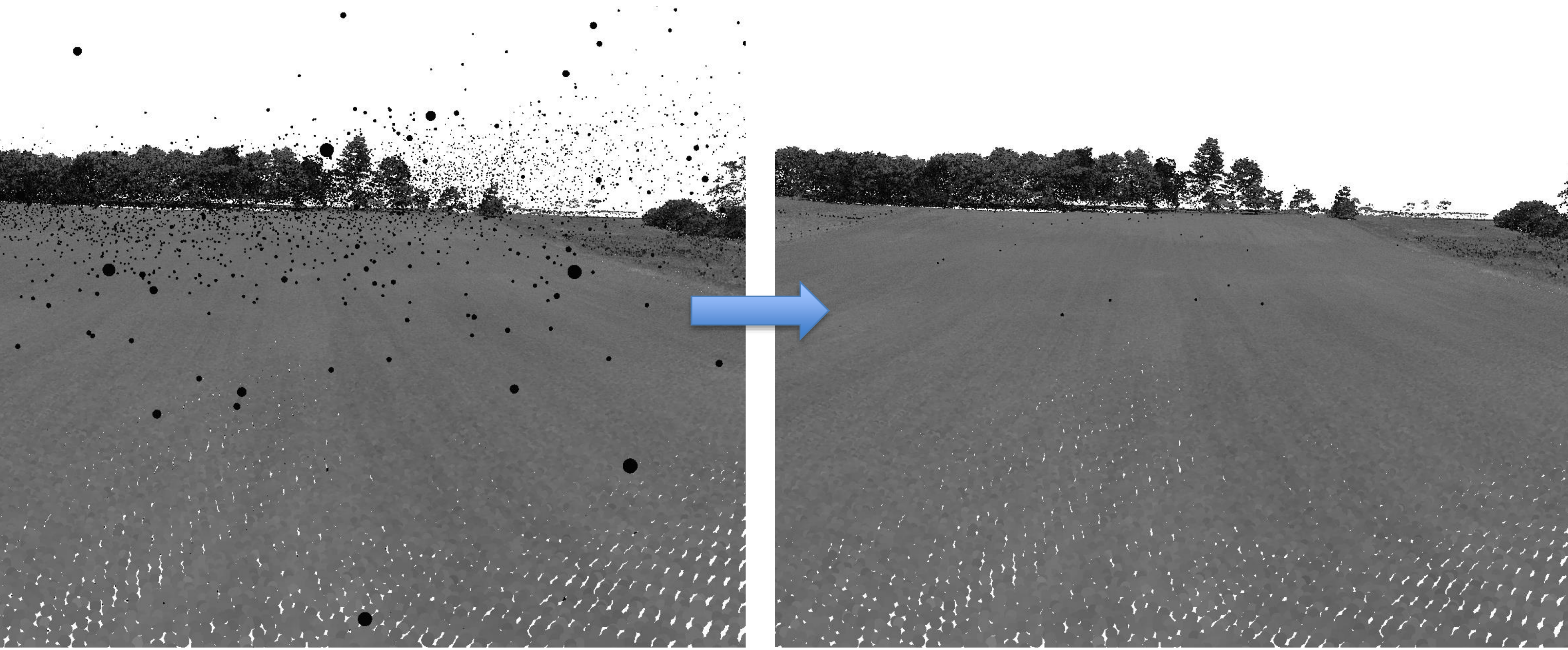
Data: Aerodata

# Results A2



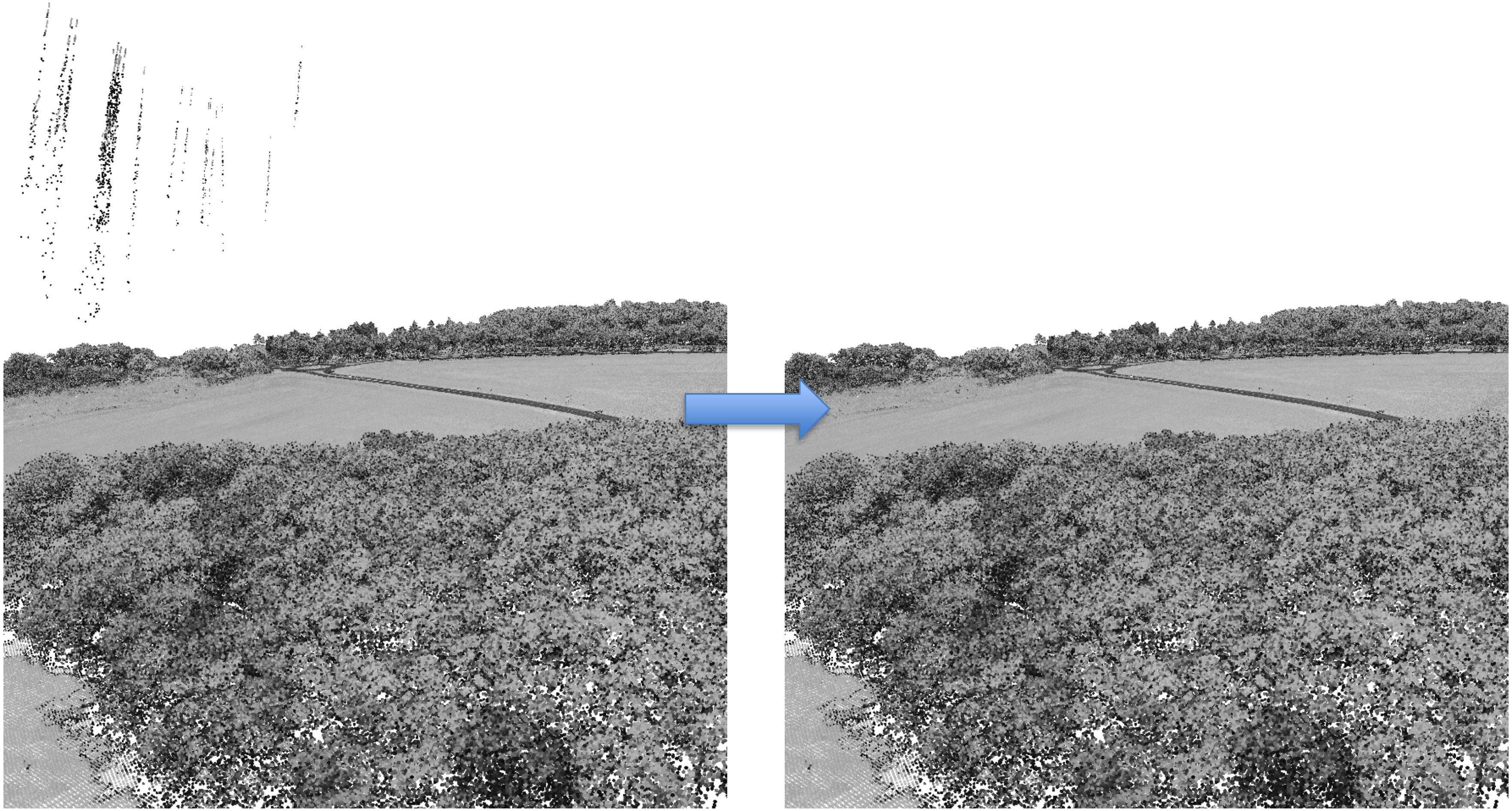
Data: Aerodata

# Type-1 outliers



Data: Aerodata

# Type-2 outliers



Data: Aerodata

# Type-3 outliers

Data: Aerodata

# Type-3 outliers

Data: Aerodata

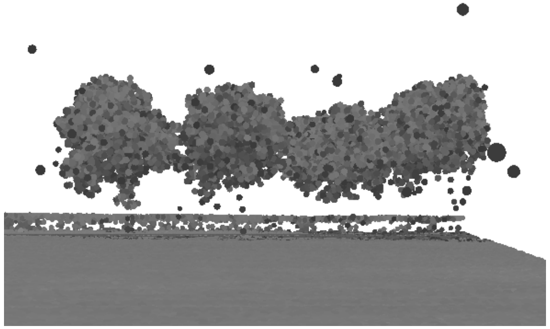
# Type-3 outliers

# Type-3 outliers

# Type-3 outliers

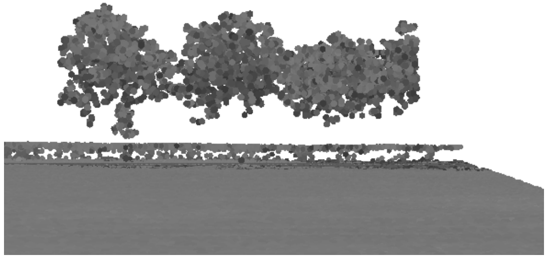
# Type-3 outliers

# Results series of operations



Source point cloud

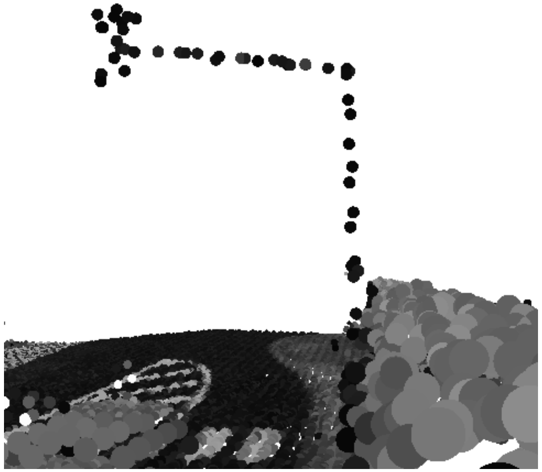
CCL



CCL
CCL after closing
Intensity
Planarity
Density

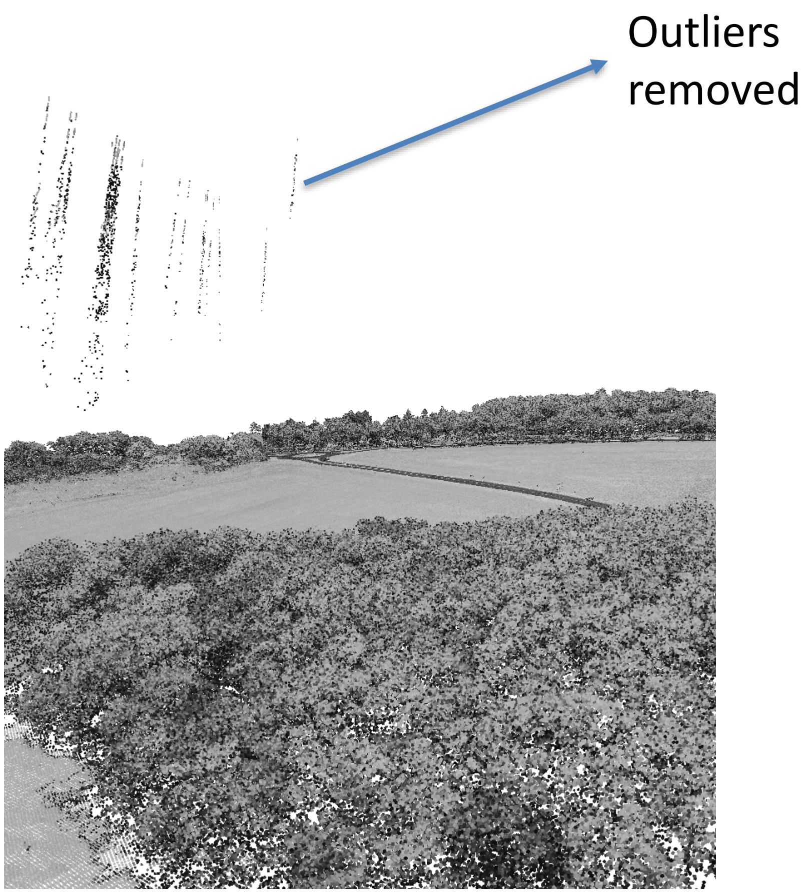
# Results

String of outliers

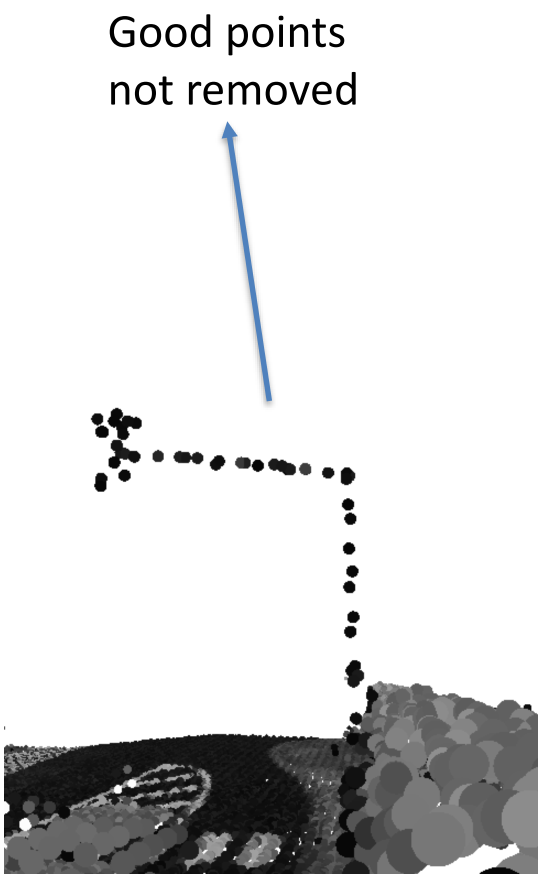


Street post

# Results




String of outliers




Street post

# Quality Assessment

- $$Sensitivity = \frac{TP}{TP + FN}$$

- $$Precision = \frac{TP}{TP + FP}$$

- $$False\ Positive\ Rate\ (FPR) = \frac{FP}{TN + FP}$$

- $$False\ Negative\ Rate\ (FNR) = \frac{FN}{TP + FN}$$

# Confusion Matrix

**A1**
Voxel size 75 cm

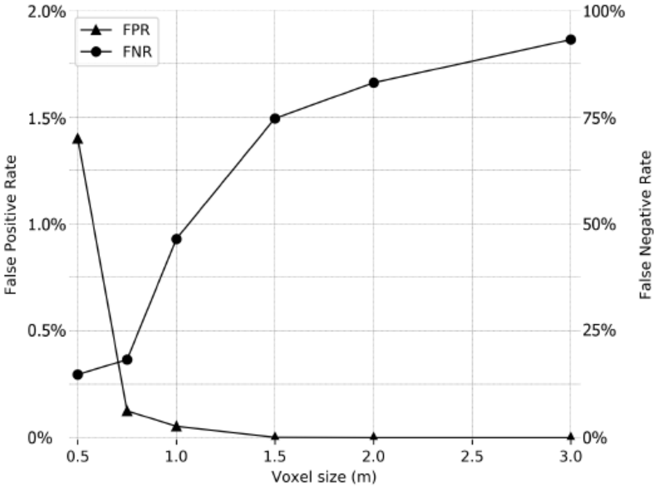
| | n = 5,743,977 | *True Condition* Positive | Negative | |
|---|---|---|---|---|
| *Predicted* | **Positive** | 68,134 | 7,109 | Sensitivity = 82.2 |
| *Condition* | **Negative** | 14,786 | 5,653,948 | Precision = 90.6 |
| | | FNR = 17.8 | FPR = 0.12 | |

**A2**
Voxel size 75 cm

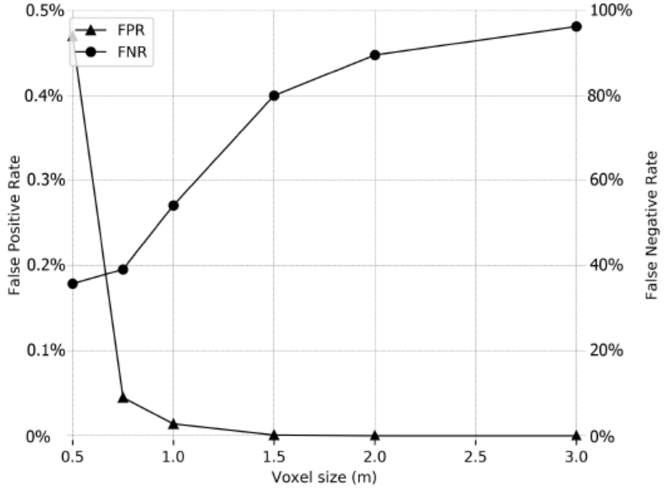
| | n = 8,275,821 | *True Condition* Positive | Negative | |
|---|---|---|---|---|
| *Predicted* | **Positive** | 66,204 | 3,740 | Sensitivity = 60.6 |
| *Condition* | **Negative** | 43,668 | 8,162,209 | Precision = 95.3 |
| | | FNR = 39.7 | FPR = 0.04 | |

TU Delft

# Accuracy / voxel size

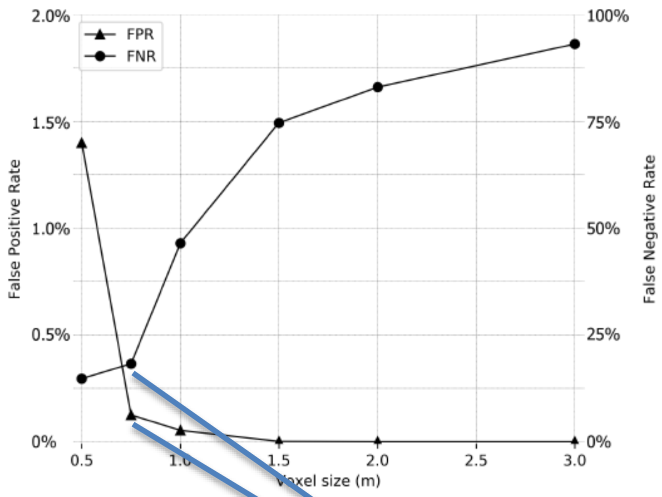



(a) A1 (Aerodata).

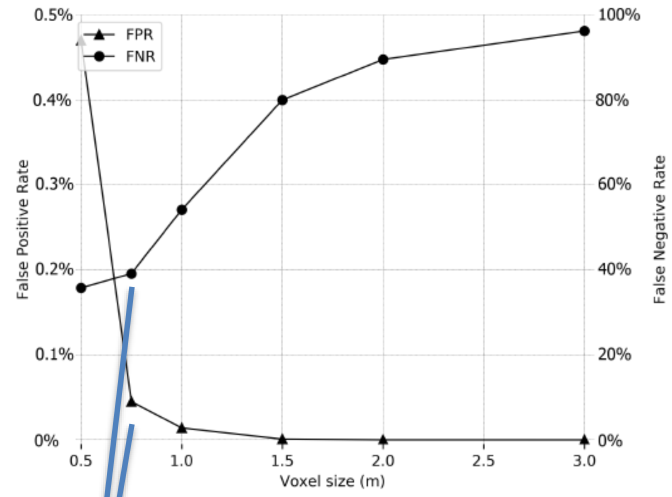



(b) A2 (Aerodata).

# Accuracy / voxel size




(a) A1 (Aerodata).




(b) A2 (Aerodata).

**0.75 m**

# Results series of operations

| A1 | *Method* | TP | FP | FN | FNR | FPR |
|---|---|---|---|---|---|---|
| | **Density** | 47,331 | 2,688 | 35,589 | 42.92 | 0.05 |
| | CCL | 69,008 | 166,908 | 13,912 | 16.78 | 2.95 |
| | CCL **after closing** | 66,257 | 5,726 | 16,663 | 20.10 | 0.10 |
| | LiDAR **intensity** | 78,558 | 2,145,036 | 4,362 | 5.26 | 37.89 |
| | **Planarity** | 76,473 | 1,369,895 | 6,447 | 7.77 | 24.20 |
| | **Overall** | 68,134 | 7,109 | 15,475 | 18.66 | 0.12 |
| A2 | | | | | | |
| | **Density** | 50,142 | 2,875 | 59,730 | 54.36 | 0.04 |
| | CCL | 69,073 | 11,317 | 40,799 | 37.13 | 0.14 |
| | CCL **after closing** | 64,194 | 1,730 | 45,678 | 41.57 | 0.02 |
| | LiDAR **intensity** | 75,983 | 532,129 | 33,889 | 31.84 | 6.52 |
| | **Planarity** | 85,240 | 2,786,722 | 24,632 | 22.42 | 34.13 |
| | **Overall** | 66,204 | 3,740 | 43,668 | 39.74 | 0.05 |

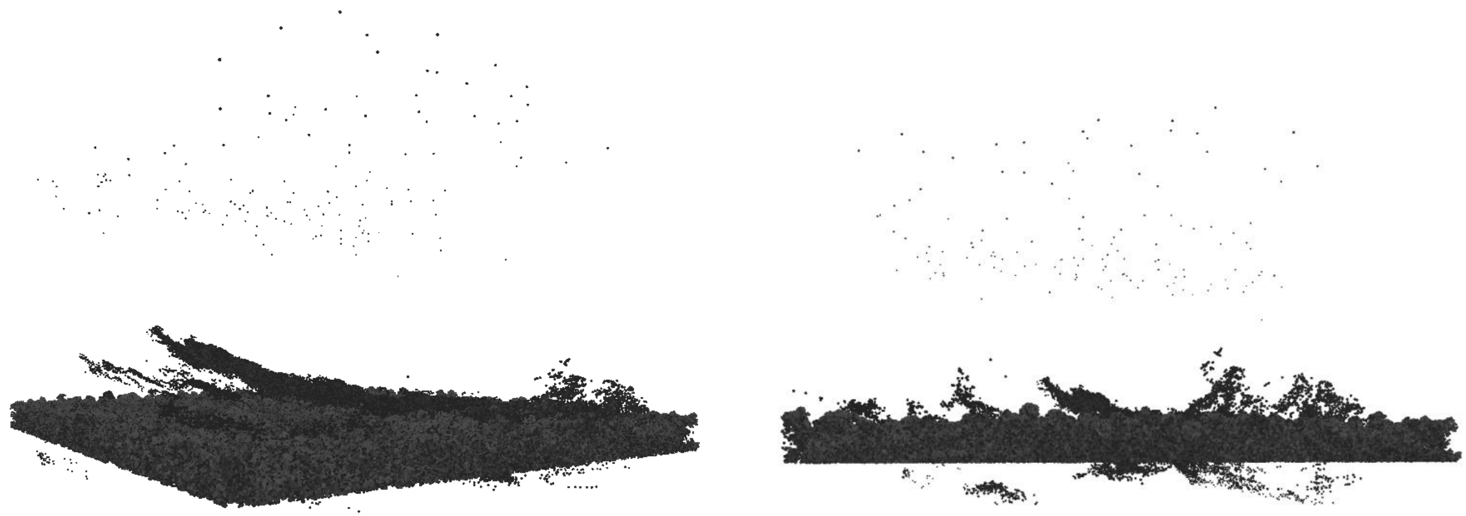
# Results series of operations

| A1 | Method | TP | FP | FN | FNR | FPR |
|---|---|---|---|---|---|---|
| | **Density** | 47,331 | 2,688 | 35,589 | 42.92 | 0.05 |
| | CCL | 69,008 | 166,908 | 13,912 | 16.78 | 2.95 |
| | CCL **after closing** | 66,257 | 5,726 | 16,663 | 20.10 | 0.10 |
| | LiDAR **intensity** | 78,558 | 2,145,036 | 4,362 | 5.26 | 37.89 |
| | **Planarity** | 76,473 | 1,369,895 | 6,447 | 7.77 | 24.20 |
| | **Overall** | 68,134 | 7,109 | 15,475 | 18.66 | 0.12 |
| A2 | | | | | | |
| | **Density** | 50,142 | 2,875 | 59,730 | 54.36 | 0.04 |
| | CCL | 69,073 | 11,317 | 40,799 | 37.13 | 0.14 |
| | CCL **after closing** | 64,194 | 1,730 | 45,678 | 41.57 | 0.02 |
| | LiDAR **intensity** | 75,983 | 532,129 | 33,889 | 31.84 | 6.52 |
| | **Planarity** | 85,240 | 2,786,722 | 24,632 | 22.42 | 34.13 |
| | **Overall** | 66,204 | 3,740 | 43,668 | 39.74 | 0.05 |

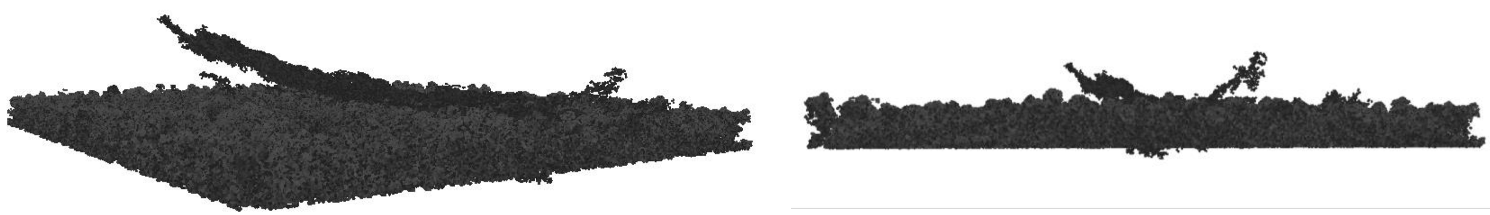
TUDelft

# Results: Common problems (FNR)

Raw point cloud

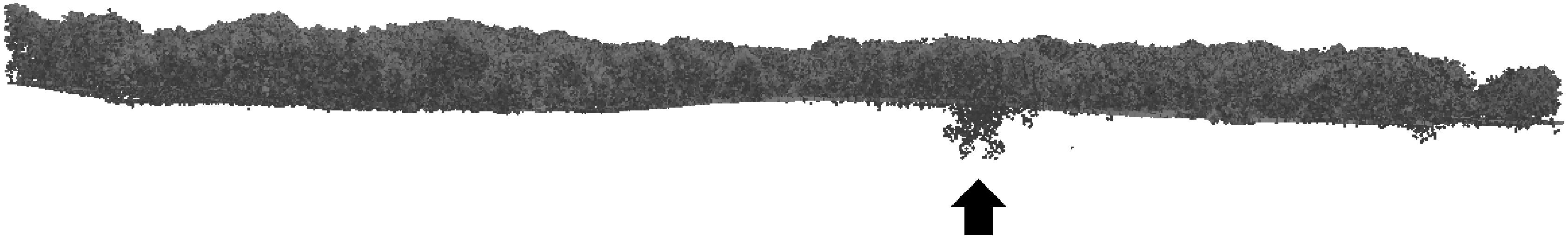


Result



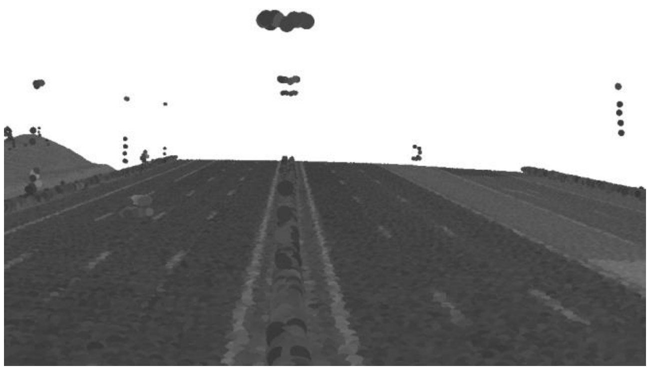
Data: Deltares

# Results: Common problems (FNR)

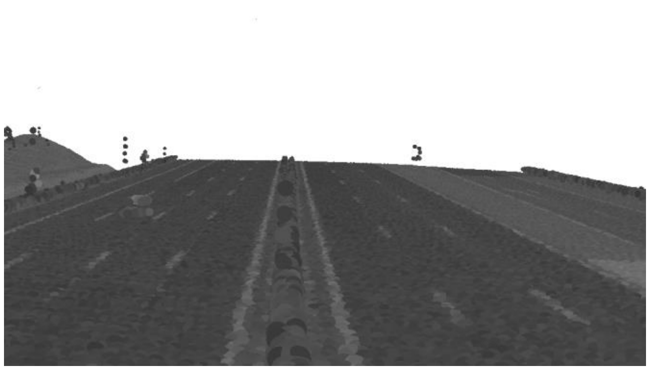


Data: Aerodata

# Results: Common problems (FPR)



Raw data: Street lights



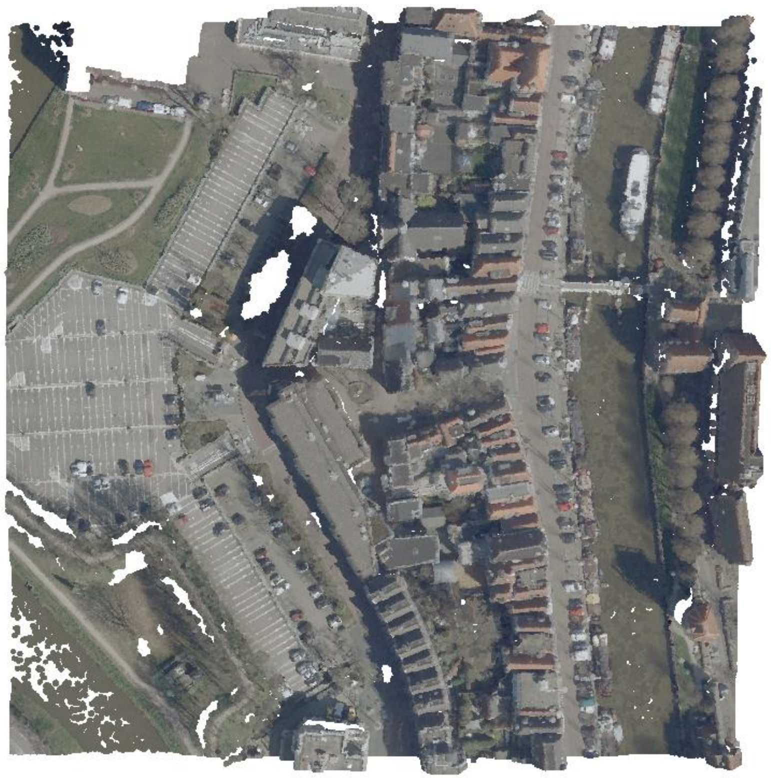
Result: removed street lights

Data: AHN3

# Results: Common problems (FPR)

Point clouds from **Dense Image Matching (DIM):**

Raw data: Dense Image Matching
Kadaster



Good points wrongly removed

# Results: Common problems (FPR)

Point clouds from **Dense Image Matching (DIM):**

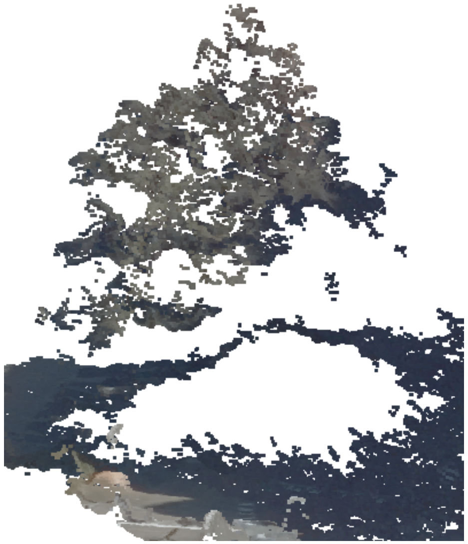
Raw data: Dense Image Matching
Kadaster

Good points wrongly removed

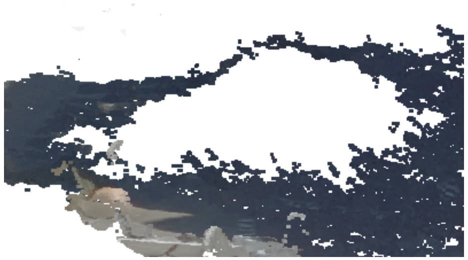
# Results: Common problems (FPR)

Point clouds from **Dense Image Matching (DIM):**

- No penetration with camera (like LiDAR)
  - → more occlusion = less connected features
- No intensity attributes

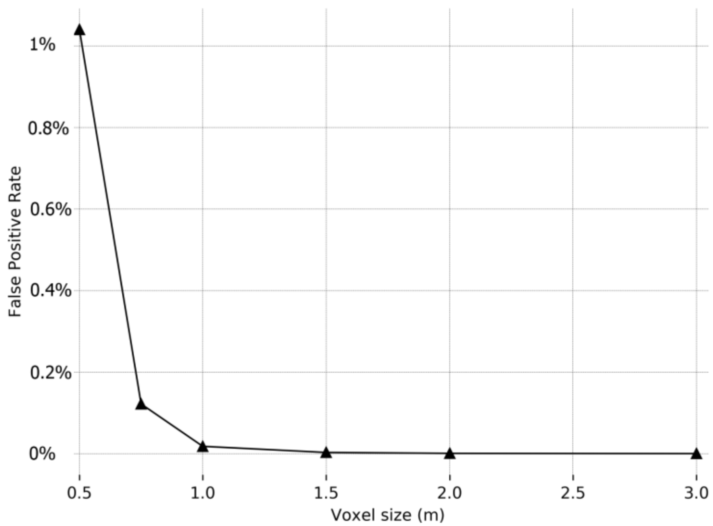


Raw data: Kadaster

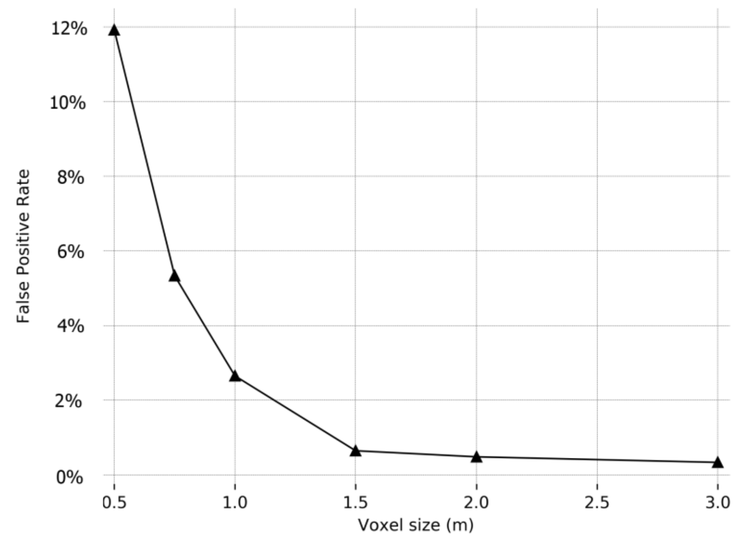


Good points wrongly removed

# Accuracy AHN 3 & DIM point cloud

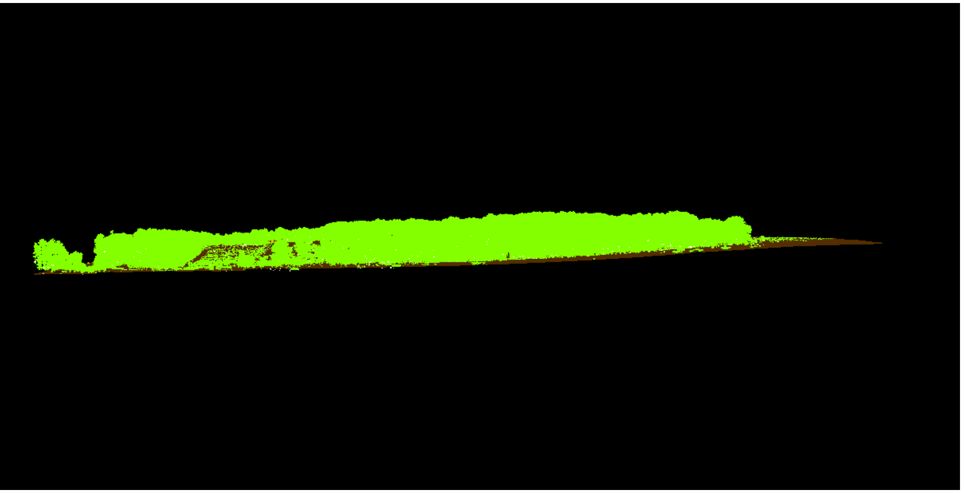



(c) AHN 3

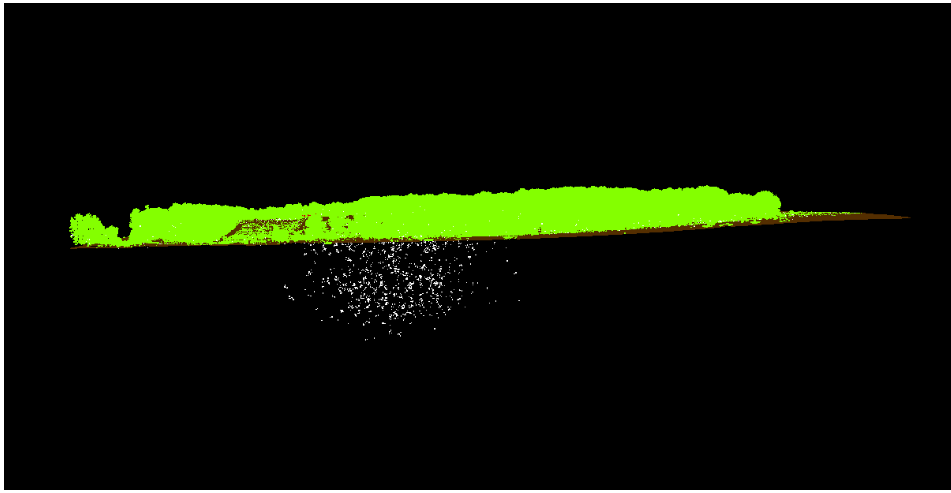



(D) Kadaster (DIM)

# Comparison to LAStools



Proposed method



LAStools

Data: Aerodata (A1)

# Comparison to LAStools

| | n = 5,743,977 | *True Condition* Positive | Negative | |
|---|---|---|---|---|
| *Predicted* | **Positive** | 62,821 | 5,699 | Sensitivity = 75.7 |
| *Condition* | **Negative** | 20,099 | 5,655,358 | Precision = 91.7 |
| | | FNR = 24.2 | FPR = 0.10 | |

Cleaning quality with LAStools.

| | n = 5,743,977 | *True Condition* Positive | Negative | |
|---|---|---|---|---|
| *Predicted* | **Positive** | 68,134 | 7,109 | Sensitivity = 82.2 |
| *Condition* | **Negative** | 14,786 | 5,653,948 | Precision = 90.6 |
| | | FNR = 17.8 | FPR = 0.12 | |

Cleaning quality of A1 with proposed method.

TUDelft

# Comparison to LAStools

| | n = 5,743,977 | *True Condition* Positive | Negative | |
|---|---|---|---|---|
| *Predicted* | **Positive** | 62,821 | 5,699 | Sensitivity = 75.7 |
| *Condition* | **Negative** | 20,099 | 5,655,358 | Precision = 91.7 |
| | | FNR = 24.2 | FPR = 0.10 | |

Cleaning quality with LAStools.

| | n = 5,743,977 | *True Condition* Positive | Negative | |
|---|---|---|---|---|
| *Predicted* | **Positive** | 68,134 | 7,109 | Sensitivity = 82.2 |
| *Condition* | **Negative** | 14,786 | 5,653,948 | Precision = 90.6 |
| | | FNR = 17.8 | FPR = 0.12 | |

Cleaning quality of A1 with proposed method.

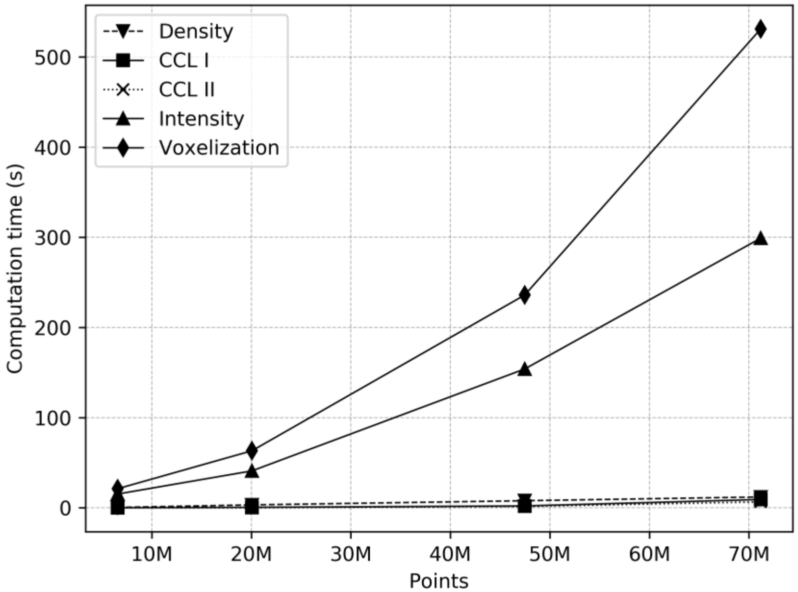
# Comparison to LAStools

- Biggest improvement = on detecting clusters:

Raw point cloud LAStools Proposed method

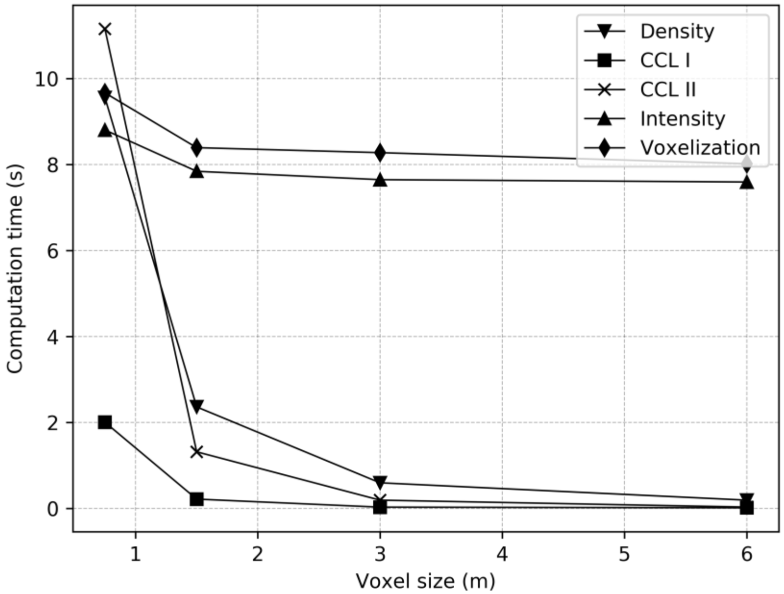
# Computation time

- Time complexity of $O(n)$ for $n$ is number of voxels




(1) Size point cloud

$O(d^2)$

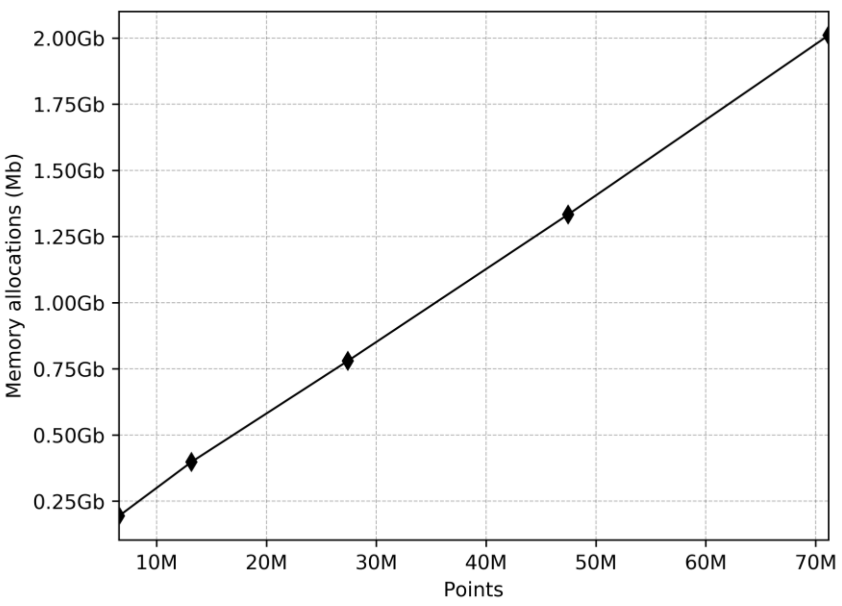



(2) Voxel size

$O(m^3)$

# Discussion & Future Work
## (1/3) Streaming

- Massive **point cloud data** could overload the memory of commodity computers



# Discussion & Future Work
## (1/3) Streaming

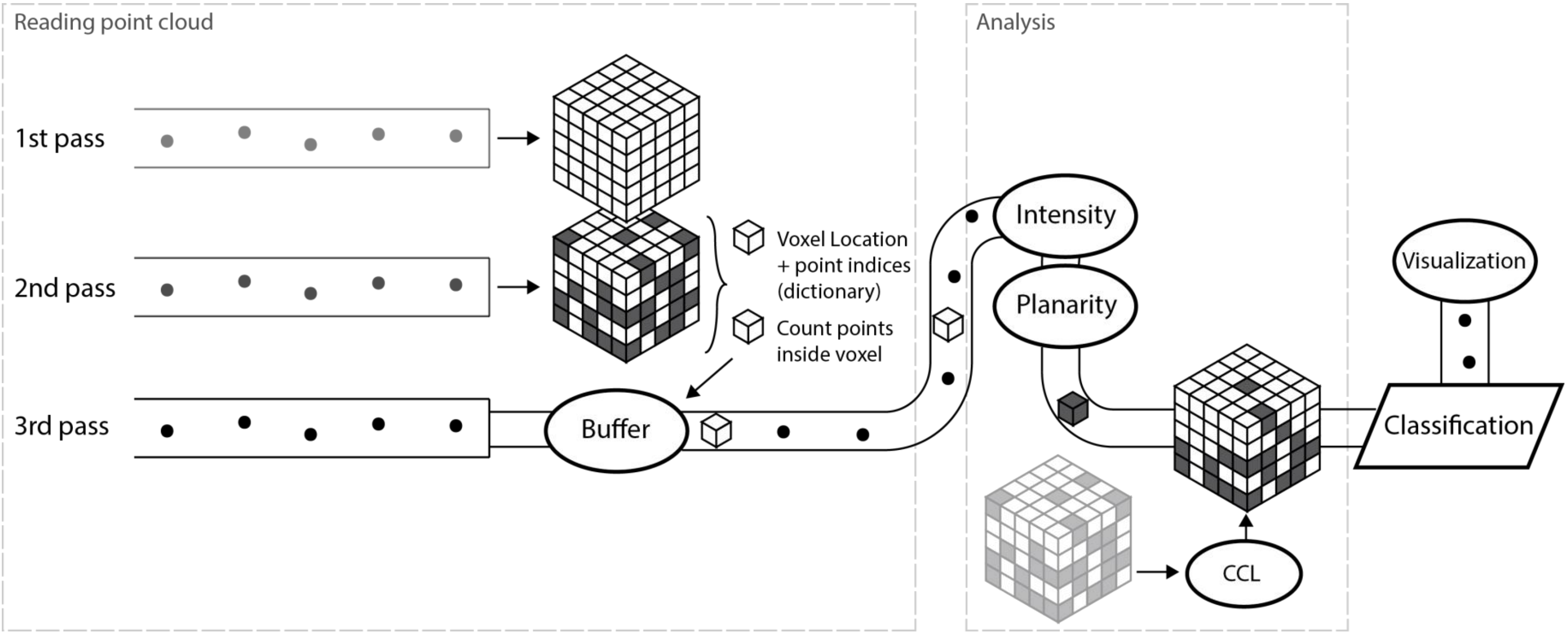
1. **Streaming solution** sequentially read points from the dataset to minimize memory requirements;

2. **Rasterized** data requires far less memory space:

| Raw point cloud | | | Voxelized point cloud | |
|---|---|---|---|---|
| **Point cloud** | **Number of points** | **Size (Mb)** | **Number of voxels** | **Size of grid (Mb)** |
| A1 | 5.7 M | 180 | 44.8 M | 5.1 |

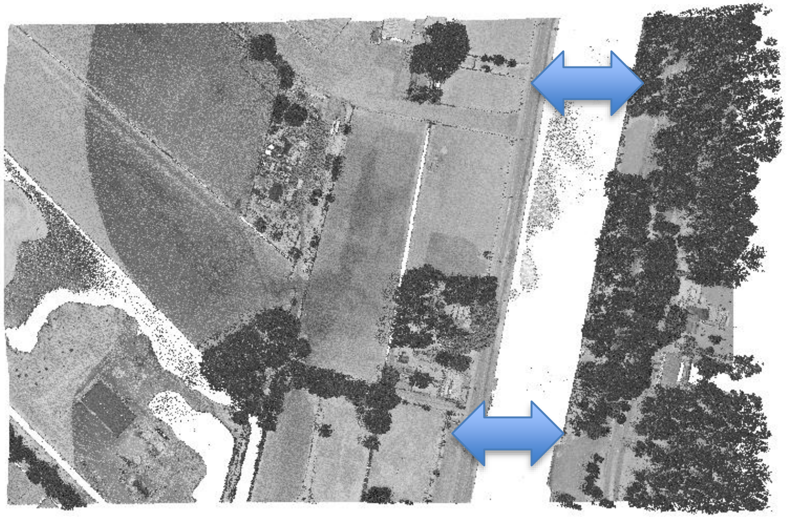
TU Delft

# Discussion & Future Work
## (1/3) Streaming



# Discussion & Future Work
## (2/3) Separation by water

Raw point cloud

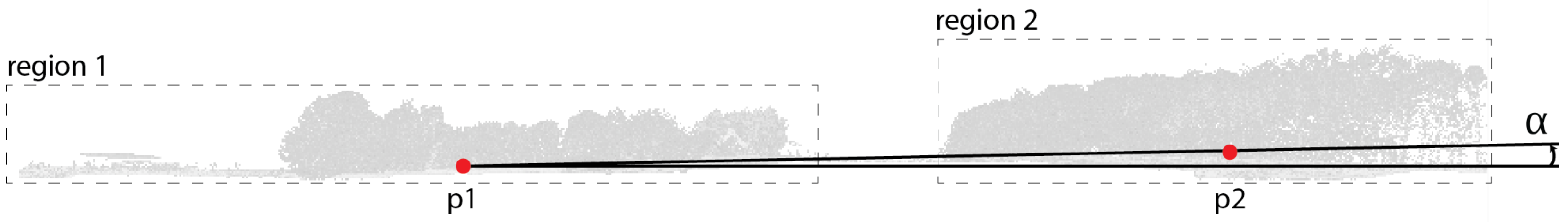


Filtered point cloud

# Discussion & Future Work
## (2/3) Separation by water

Compare incline between regions

- Outliers have large incline
- Terrain points have negligible incline



# Discussion & Future Work
## (3/3) Outlier Classification

- Arbitrary classification rule (threshold) for proposed method
- Supervised learning classifiers could be exploited to classify voxels
  - Predict probability of outlier class
- Use the five proposed operations to extract features, and train a classifier
- Need training data!

# Conclusions

- Detect all types of outliers
  - Problems with connected outliers, or unconnected good points (by water)
- Integration of series of methods in voxel structure
  - Minimize false positives while keeping high sensitivity
- Connected Components Labeling for outlier detection
  - After closing
  - Voxel size 0.75m – 1m
- Dense Image Matching point cloud != LiDAR for outlier detection

# Thanks