Fast SFC-based point cloud change detection

Vitali Diaz, Peter van Oosterom, Martijn Meijers,

Edward Verbree, Thijs van Lankveld and Nauman Ahmed

TU Delft

netherlands eScience center

nD-PointCloud

http://nd-pc.org/

November 12, 2024

Outline

1. Motivation
2. Direct point cloud-based change detection
3. Fast Space-Filling Curve-based point cloud change detection
4. Conclusions
5. Future work

1. Motivation: The dream

nd-pc.org

Point Clouds as a digital representation of the (built) environment

nd-pc.org

AHN3
point cloud from
Airborne Laser Scanning
(ALS)

Digital Surface Model (DSM)
Point cloud using aerial
photographs

Point Clouds as a digital representation of the (built) environment

nD-PointCloud
nd-pc.org

AHN3
point cloud from
Airborne Laser Scanning
(ALS)

Digital Surface Model (DSM)
Point cloud using aerial
photographs

Indoor, occluded areas, shadow zones?

TU Delft netherlands eScience center

Point Clouds as a digital representation of the (built) environment

nd-pc.org

AHN3
point cloud from
Airborne Laser Scanning
(ALS)

Digital Surface Model (DSM)
Point cloud using aerial
photographs

Indoor, occluded areas, shadow zones?

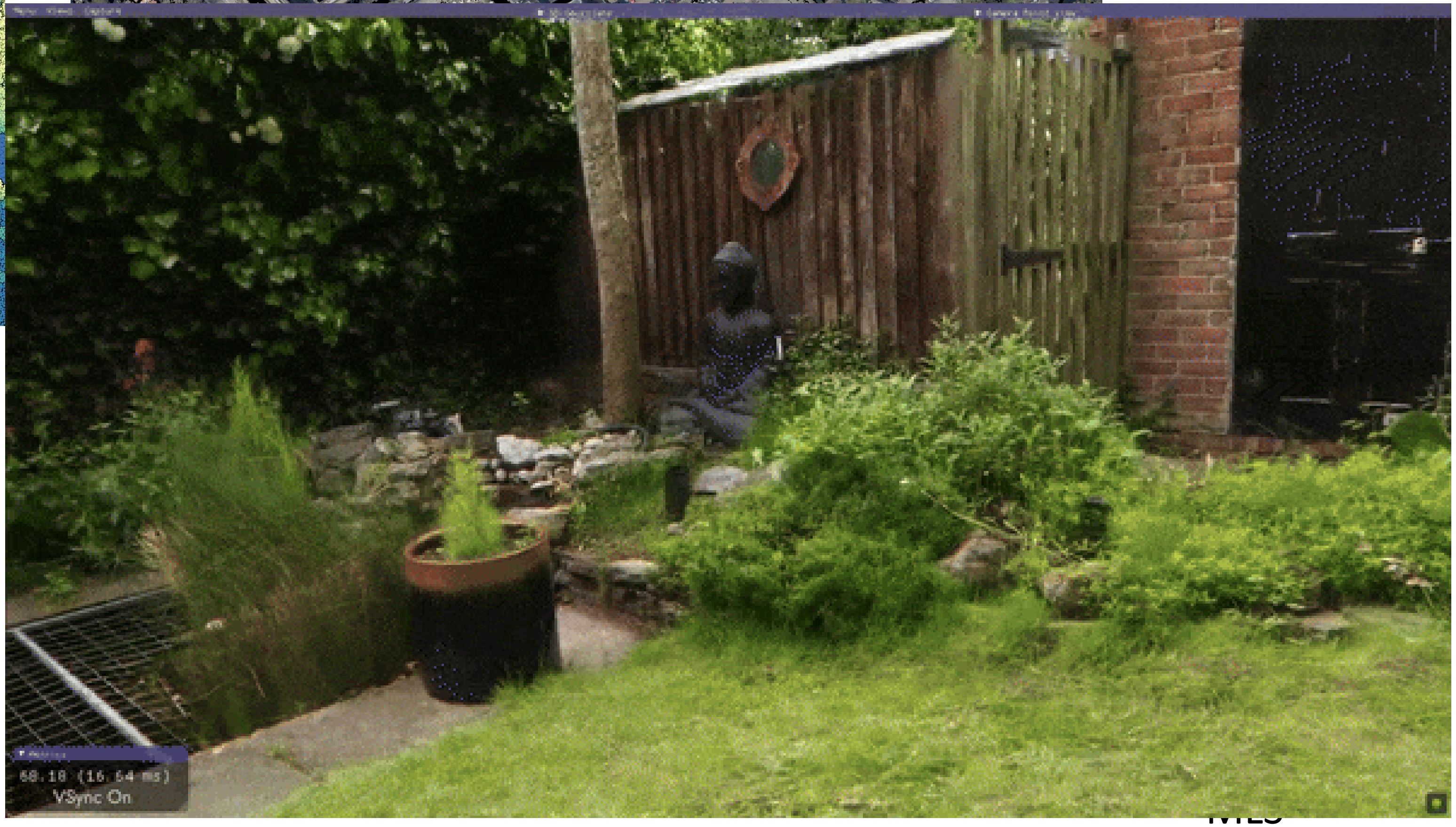
filling the data gap, and improving the nD representation of the (built) environment

Mobile Laser Scanning
MLS

Point Clouds as a digital representation of the (built) environment

nD-PointCloud
nd-pc.org

AHN3
point cloud from
Airborne Laser Scanning
(ALS)

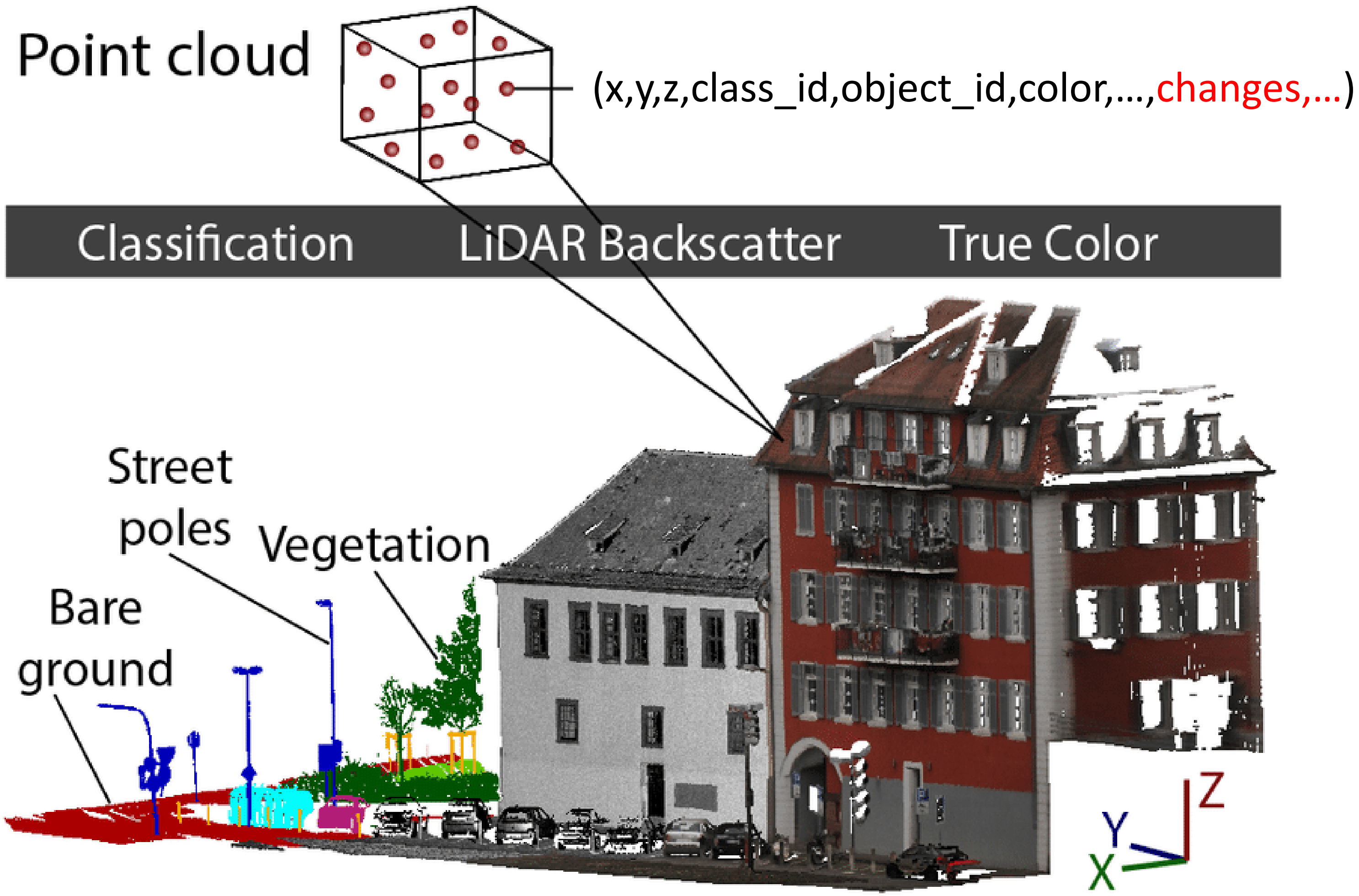


Rendering:
3D Gaussian Splatting

https://huggingface.co/papers/2308.04079

TU Delft
netherlands eScience center

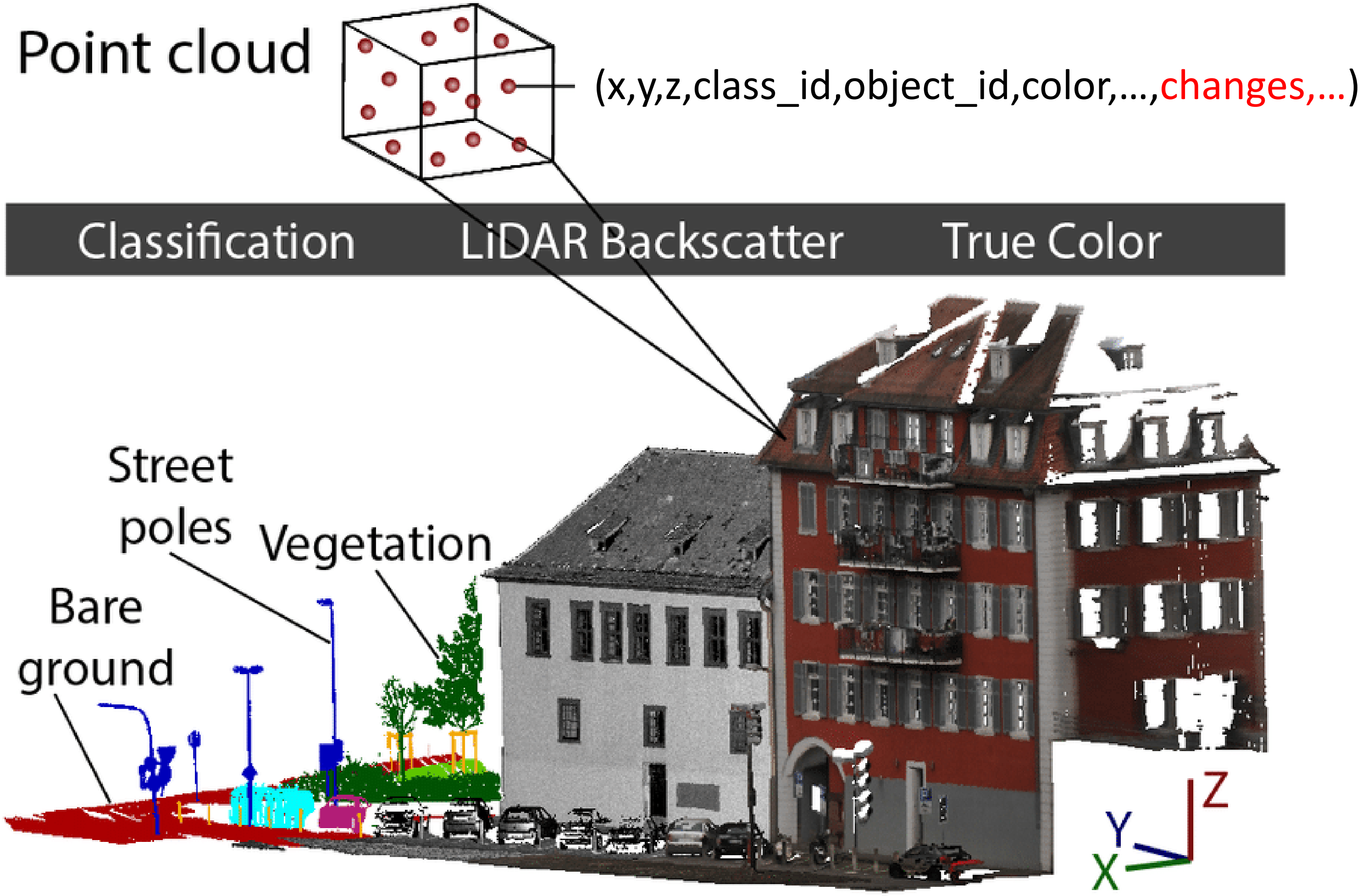
Beside x,y,z coordinates, other attributes are (and can be) attached to every single point



Ghamisi et al. (2019). IEEE GRS Magazine, https://doi.org/10.1109/mgrs.2018.2890023

Beside x,y,z coordinates, other attributes are (and can be) attached to every single point

nd-pc.org



Ghamisi et al. (2019). IEEE GRS Magazine, https://doi.org/10.1109/mgrs.2018.2890023

Visualising additional attributes

1 Motivation

Point clouds as direct data sources for change detection

Point clouds provide rich information for visualization and serve as direct data sources for change detection analysis, enabling analysts to detect, quantify, and interpret spatial changes directly without data conversion and additional storage (for grid conversion).

Leveraging inherent point cloud characteristics

By utilizing the unique characteristics of point clouds, researchers and practitioners can gain valuable insights into dynamic processes, supporting informed decision-making, planning, and development.

Research contribution

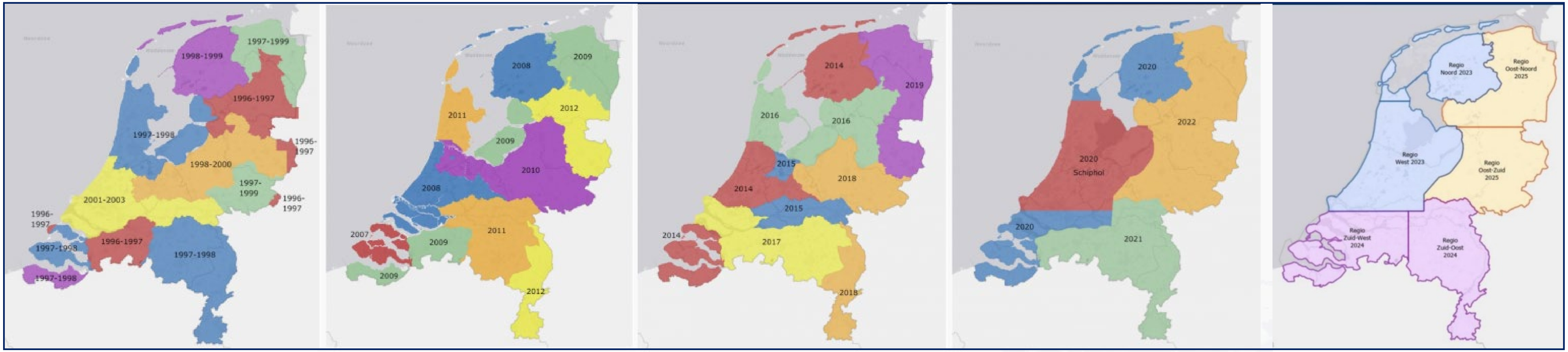
This research advances the development of direct point cloud change detection methods and explores optimization strategies to enable fast computations on large-scale datasets, such as the massive point cloud of the entire Netherlands (AHN).

The data (and the challenge): the height model of the Netherlands

nd-pc.org

The 'Actueel Hoogtebestand Nederland' (AHN) is the massive point cloud dataset acquired through laser altimetry and covers height information of the Netherlands

https://www.ahn.nl/



AHN1 (1996 - 2003) | AHN2 (2007 – 2012) | AHN3 (2014 – 2019) | AHN4 (2020 – 2022) | **AHN5 (2023- in progress)**

6 to 10 points per m^2 | 10 to 14 points per m^2

TU Delft | netherlands eScience center

Using a bow is not an option because I'm not going to turn stones into arrows.
The choice is to use the slingshot to throw stones.

So the questions:
-which method and why?

-how to develop the implementation for fast computation?

2. Direct point cloud-based change detection

nD-PointCloud
nd-pc.org

TU Delft netherlands eScience center

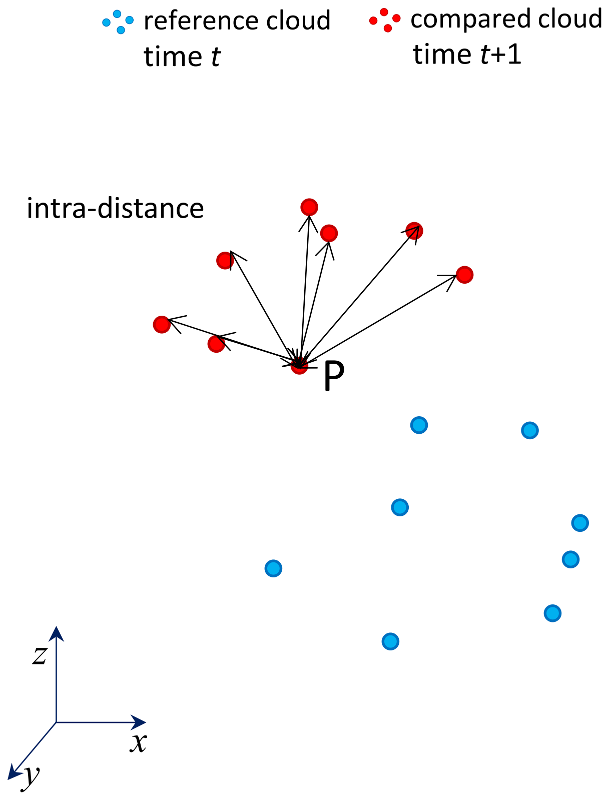
2 Direct pc-based change detection

- two experiments to find the suitable pc-based method:
  - (1) controlled offset,
  - (2) comparison over spatio-temporal point clouds
- range of algorithms (as available in CloudCompared)
- reimplemented in Matlab for full control
- testing with various data sets (AHN, CoastScan, lake, bunny, etc)

2 Direct pc-based change detection

Some definitions

Diaz et al. (2024)
https://doi.org/10.1007/978-3-031-43699-4_20



Intra-distance: distance between points within the same point cloud.

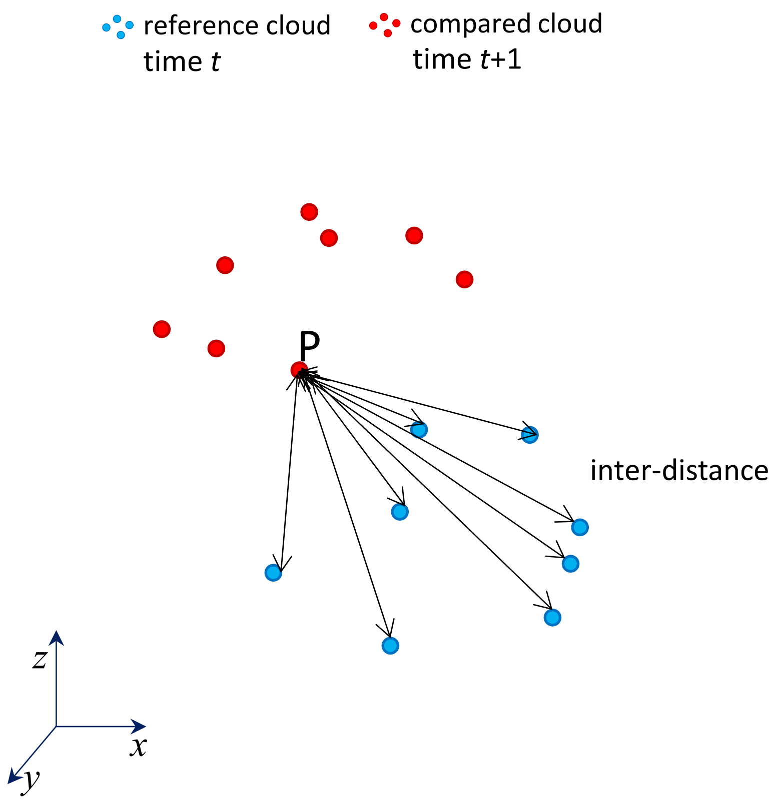
For point *P*, distance is calculated as the average of the Euclidian distances of its *k* nearest neighbors.

Intra-distance is calculated for each point.

2 Direct pc-based change detection

Some definitions

Diaz et al. (2024)
https://doi.org/10.1007/978-3-031-43699-4_20



Intra-distance: distance between points within the same point cloud.

For point *P*, distance is calculated as the average of the Euclidian distances of its *k* nearest neighbors.

Intra-distance is calculated for each point.

Inter-distance: distance between two corresponding point clouds taken at different epochs (also known as c2c distance).

It is calculated for each point of the compared cloud, providing the spatial dissimilarities/similarities between the two point clouds.

There are various methods to calculate it.

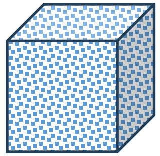
2 Direct pc-based change detection

nD-PointCloud
nd-pc.org

Experiment 1

Controlled offset test:

(1) Select a point cloud as the "reference cloud"



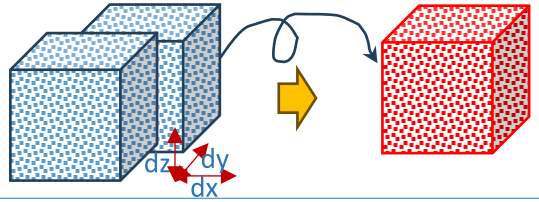
(2) Calculate intra-distance for every point of the reference cloud and the average intra-distance.

intra-distance
$\text{intra-distance}_{avg}$

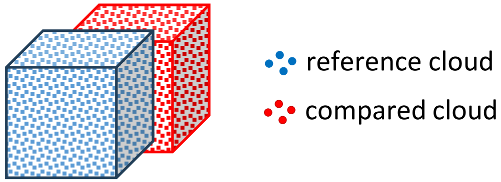
(3) Set artificial offsets based on the average intra-distance (i.e., below, above) to analyze a range of controlled offset scenarios.

dx,dy,dz

(4) Apply the offsets to all points of the reference cloud to create the "compared cloud" for each controlled offset scenario.



(5) Calculate the inter-point distance (c2c distance) between the compared and the reference cloud. Eight different methods.



(6) Finally, evaluate each method to analyze its accuracy in capturing the applied offset.

Diaz et al. (2024)
https://doi.org/10.1007/978-3-031-43699-4_20

TU Delft
netherlands eScience center

2 Direct pc-based change detection

nD-PointCloud
nd-pc.org

Experiment 1

Diaz et al. (2024)
https://doi.org/10.1007/978-3-031-43699-4_20

- Testing 4 different datasets (bunny, lake, CoastScan, and AHN)
- 3 different types of offsets: vertical, horizontal, and diagonal
- 8 different methods

TU Delft netherlands eScience center

2 Direct pc-based change detection

nD-PointCloud
nd-pc.org

Inter-distance (C2C distance)

-Simple approach	
• The nearest neighbor	dx,dy,dz
-Weighted methods	
• Natural Neighbor Interpolation (NNI)	dz
• Inverse Distance Weight (IDW)	dz
-Local modelling	
Point-Model	
• Least squares plane	dx,dy,dz
• Linear interpolation	dx,dy,dz
• 2.5D triangulation	dx,dy,dz
• Quadratic (height function)	dx,dy,dz
Model-Model	dx,dy,dz
• Multiscale Model to Model Cloud Comparison (M3C2)	dx,dy,dz

- time_cost_querying
- time_cost_modelling
- time_cost_distance_calculation

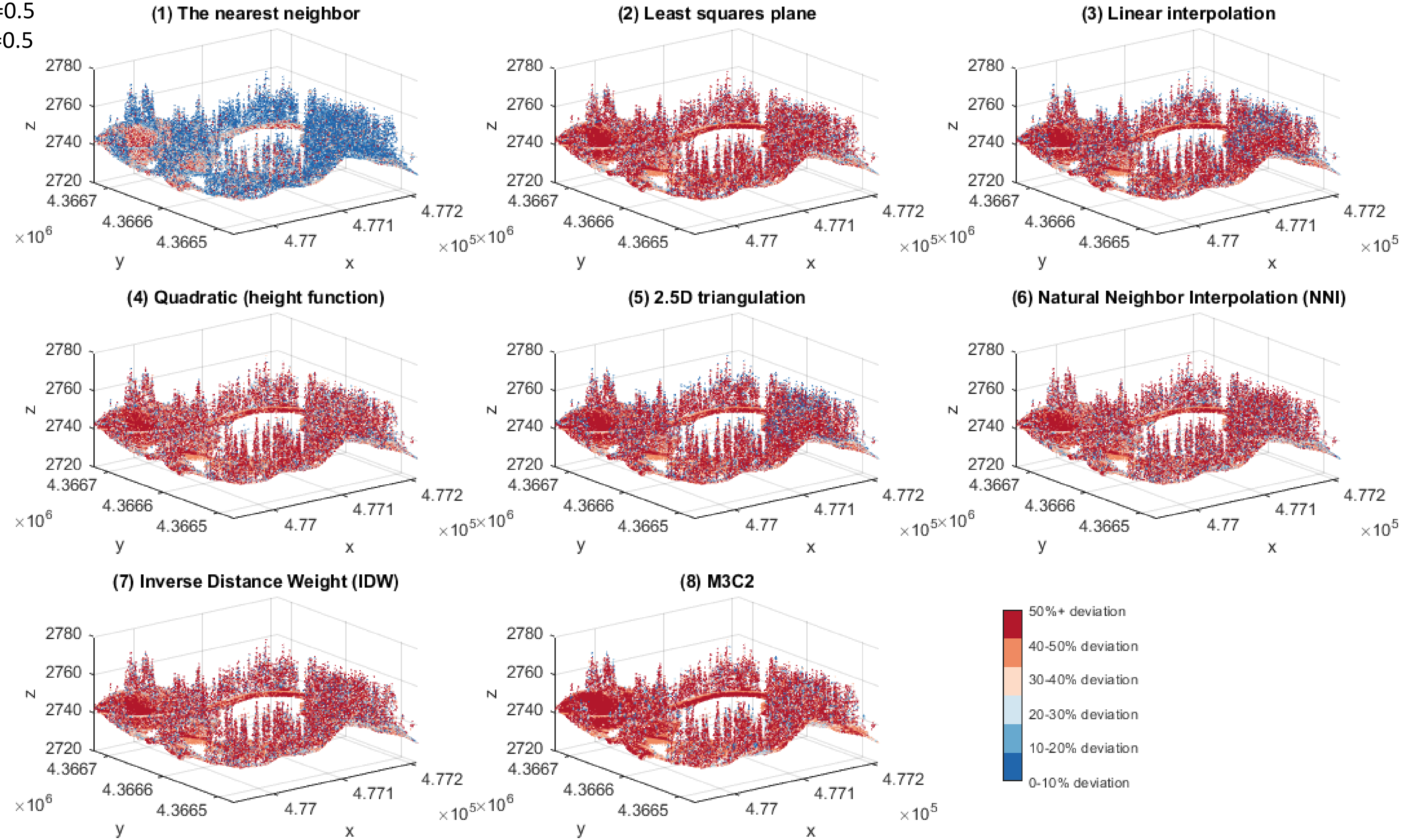
TU Delft netherlands eScience center

Diaz et al. (2024)
https://doi.org/10.1007/978-3-031-43699-4_20

Average intra-distance = 1.06 m

dx=0.5 m
dy=0.5
dz=0.5

Experiment 1

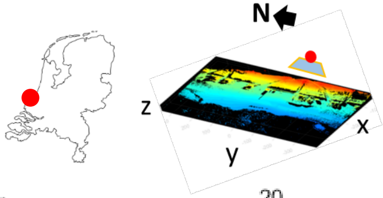


Diaz et al. (2024)
https://doi.org/10.1007/978-3-031-43699-4_20

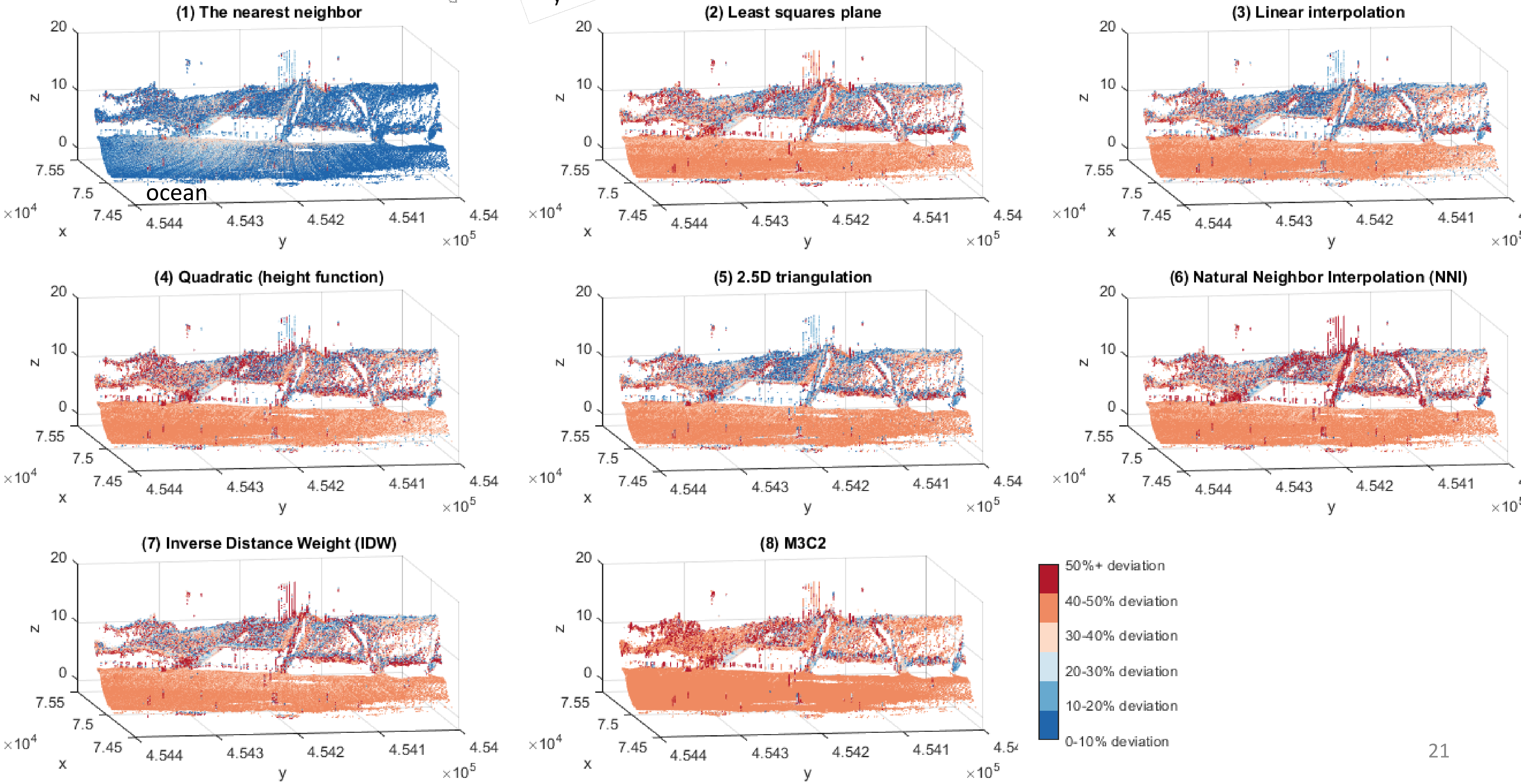
Experiment 1: CoastScan data with controlled offset

Average intra-distance = 0.396 m

dx=0.2 m
dy=0.2
dz=0.2



Diaz et al. (2024)
https://doi.org/10.1007/978-3-031-43699-4_20



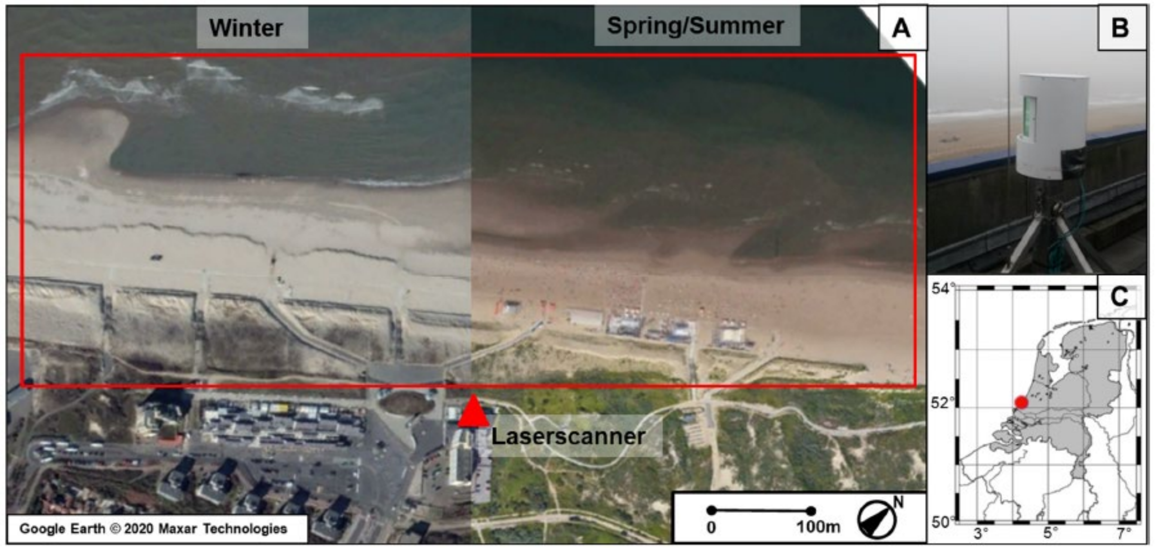
2 Direct pc-based change detection

Experiment 2

Diaz et al. (2024)
https://doi.org/10.5194/egusphere-egu24-8191

Change detection with spatio-temporal point clouds

Kijkduin beach-dune system



(A) Aerial photo of the beach-dune in Kijkduin and the scan Area within the red rectangle.
(B) The laser scanner positioned on a hotel next to the beach (indicated by a red triangle in A).
(C) The location of Kijkduin (52.07°N, 4.22°E) in The Netherlands.

Vos et al. (2022)
https://doi-org.tudelft.idm.oclc.org/10.1038/s41597-022-01291-9

Performance evaluation

1) Comparison with the distances calculated with the raster-to-raster approach

2) Time cost comparison

- time_cost_querying
- time_cost_modelling
- time_cost_distance_calculation

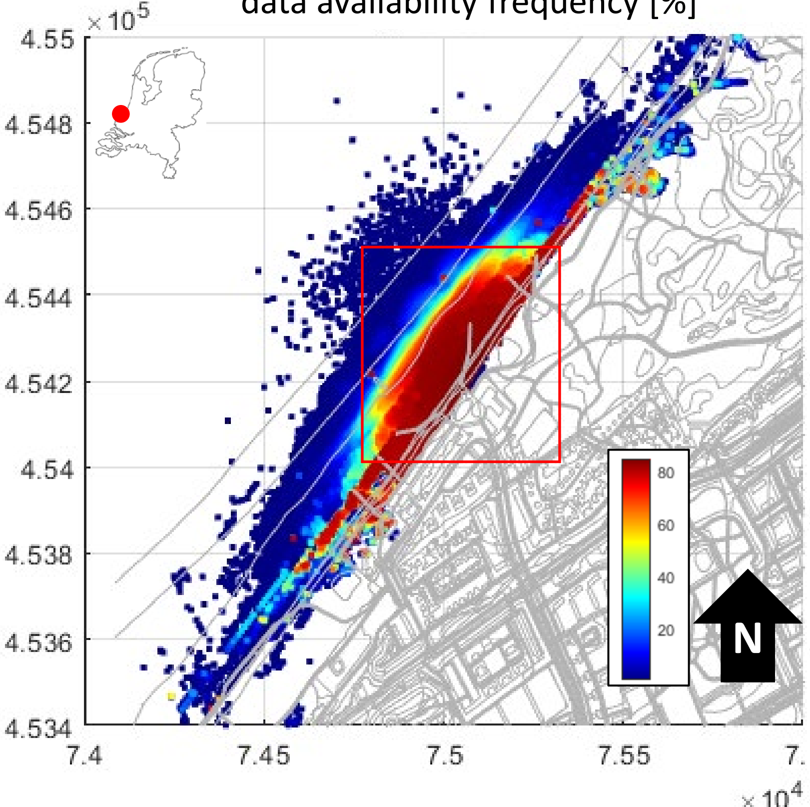
TU Delft netherlands eScience center

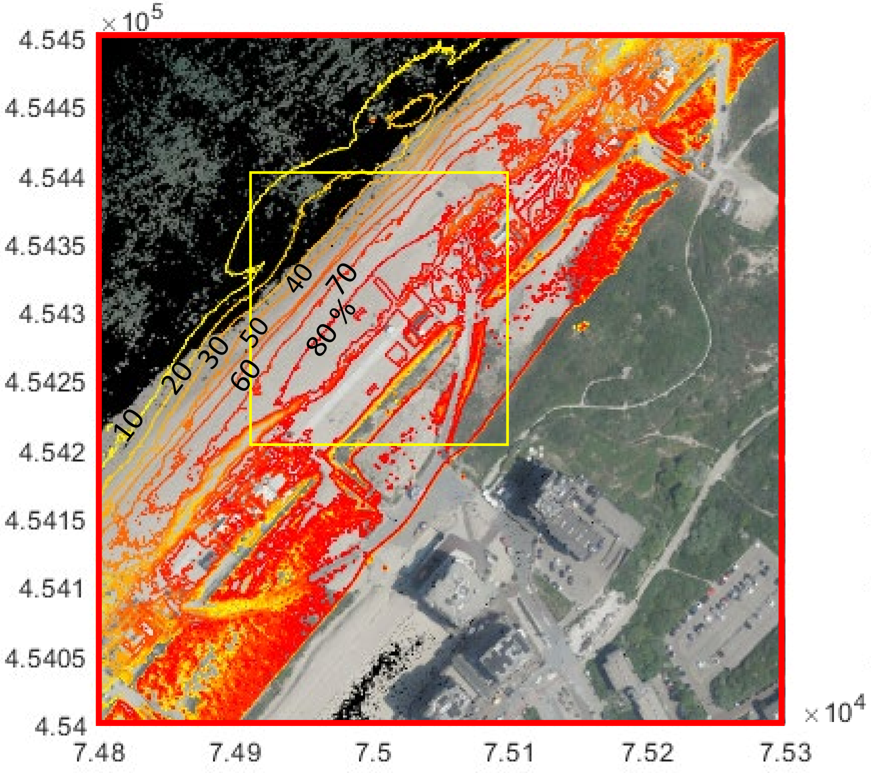
Experiment 2: spatio-temporal CoastScan data

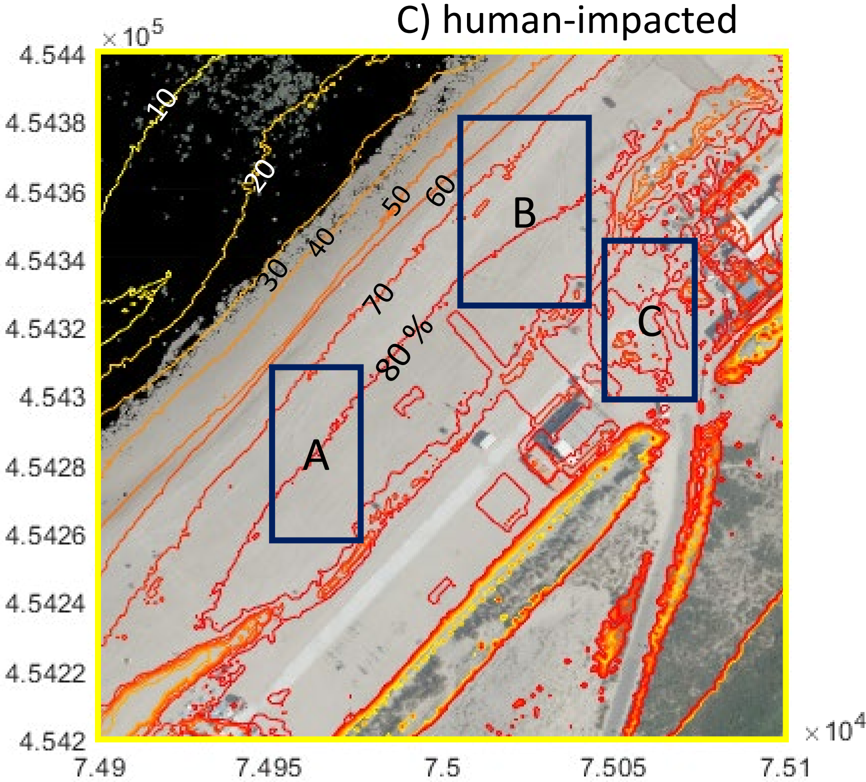
period 2017/01/01 to 2017/04/18
2,187 epochs

A) partially natural
B) intermediate
C) human-impacted

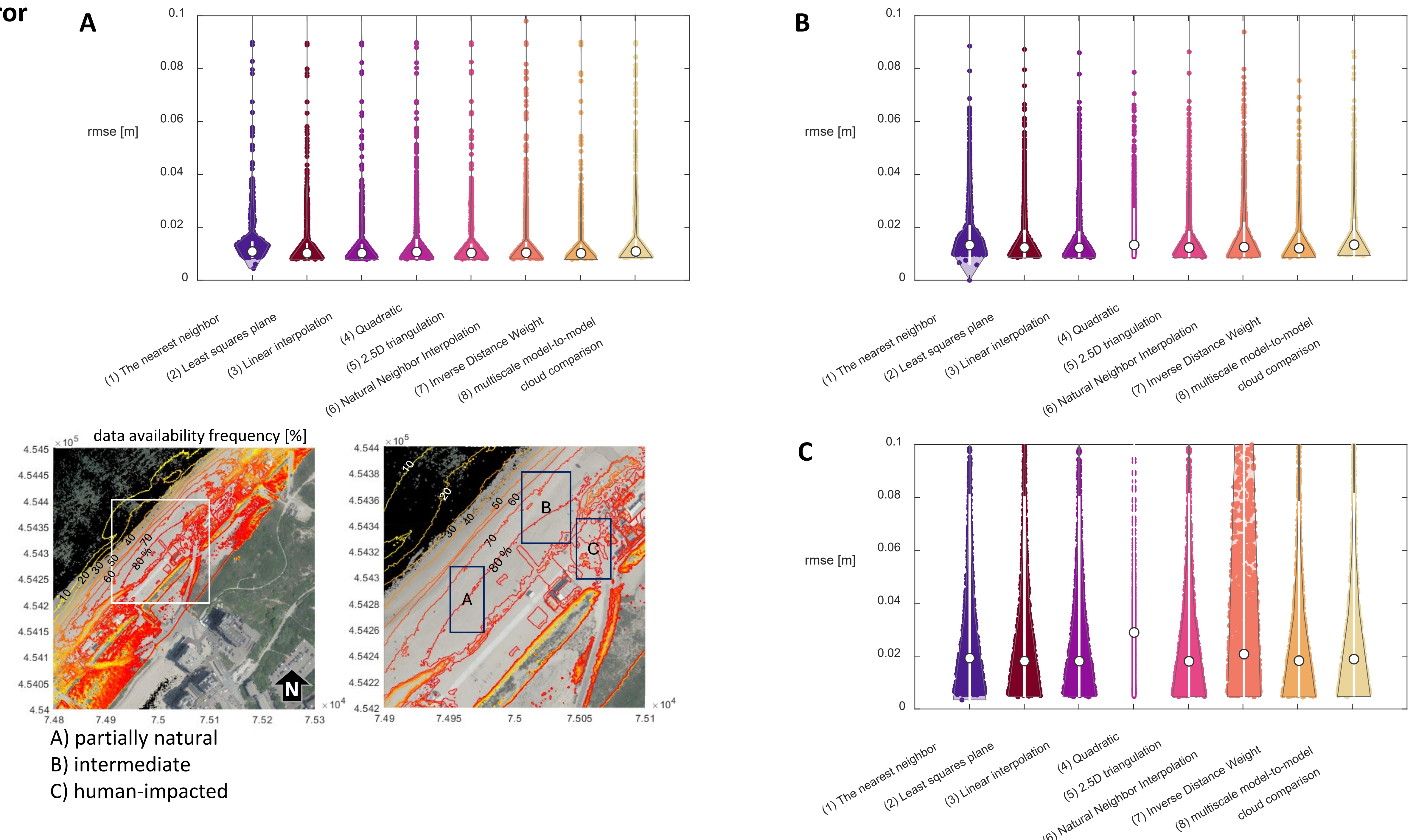
data availability frequency [%]



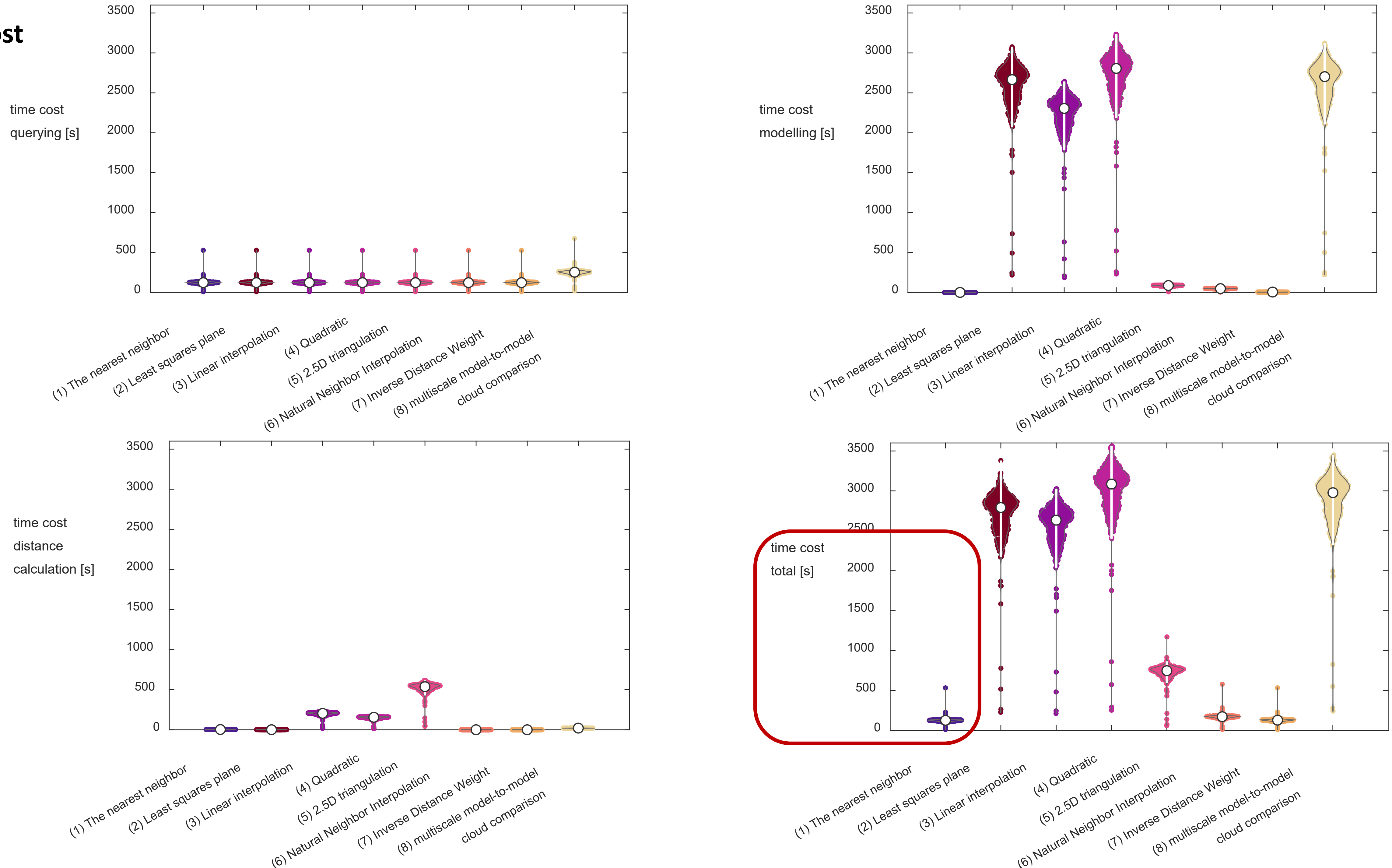




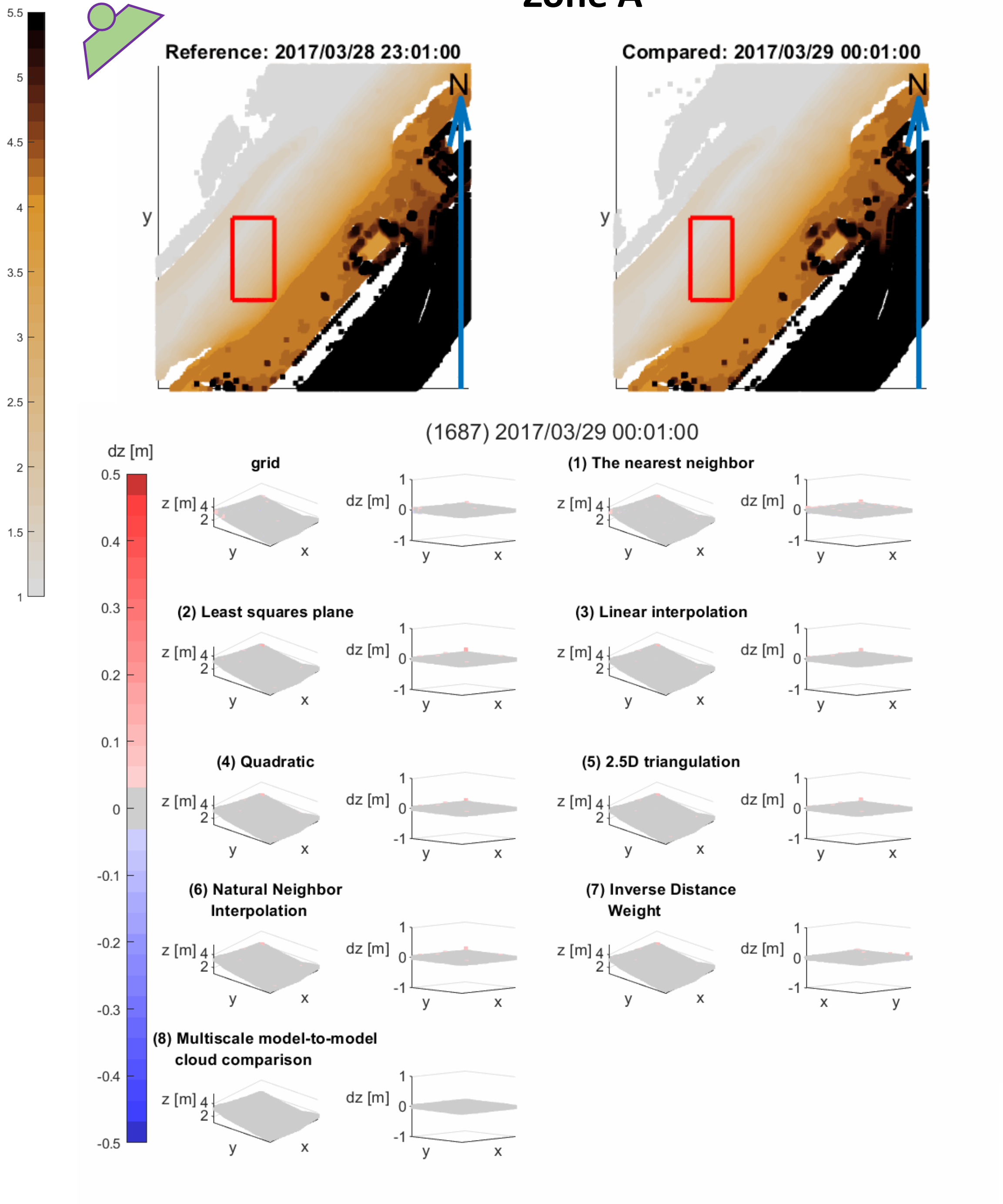
Error


A
B
C
rmse [m]
0.1
0.08
0.06
0.04
0.02
0
(1) The nearest neighbor
(2) Least squares plane
(3) Linear interpolation
(4) Quadratic
(5) 2.5D triangulation
(6) Natural Neighbor Interpolation
(7) Inverse Distance Weight
(8) multiscale model-to-model cloud comparison
data availability frequency [%]
10
20
30
40
50
60
70
80 %
N
A
B
C
A) partially natural
B) intermediate
C) human-impacted

Time cost

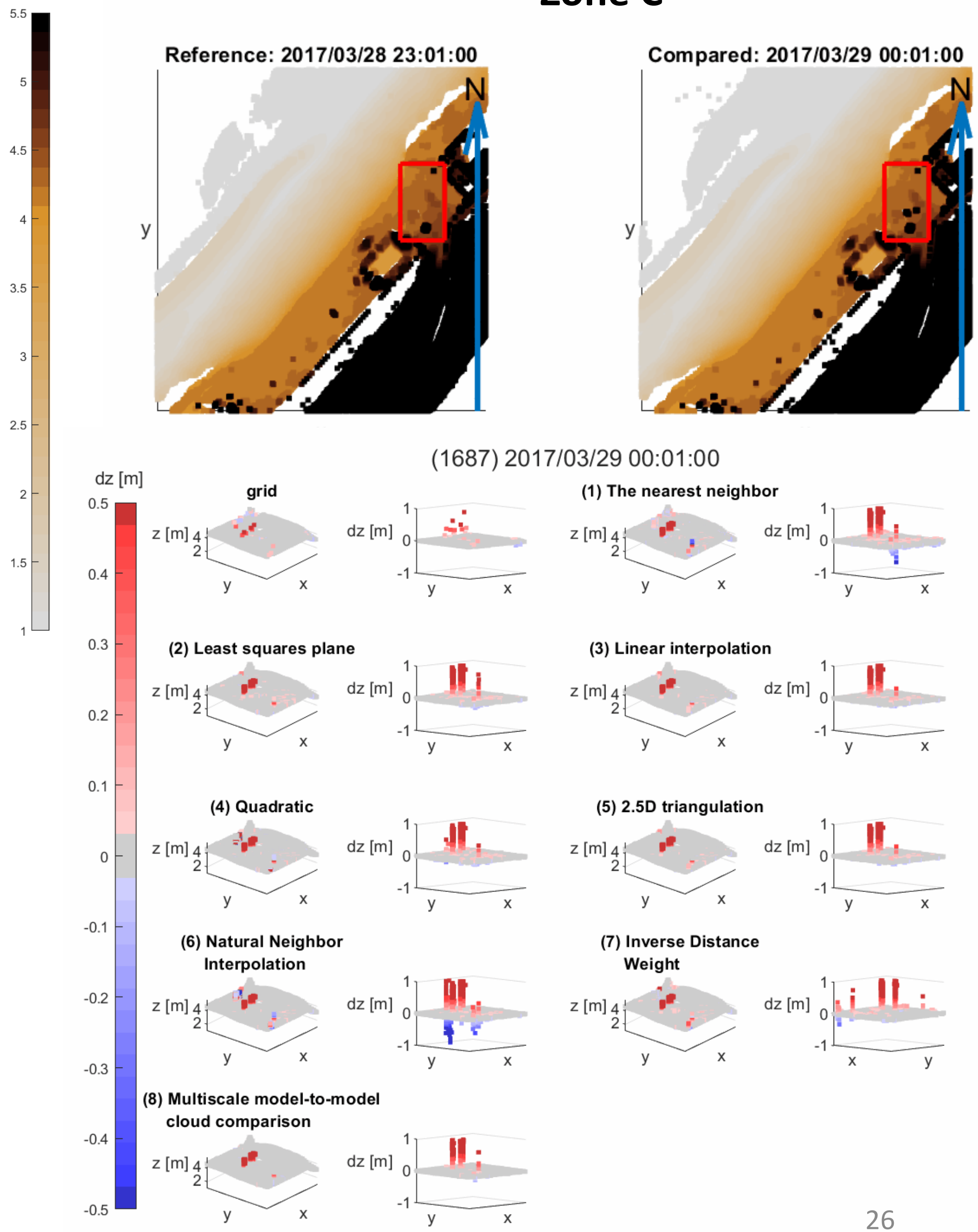

time cost querying [s]
time cost modelling [s]
time cost distance calculation [s]
time cost total [s]
3500
3000
2500
2000
1500
1000
500
0
(1) The nearest neighbor
(2) Least squares plane
(3) Linear interpolation
(4) Quadratic
(5) 2.5D triangulation
(6) Natural Neighbor Interpolation
(7) Inverse Distance Weight
(8) multiscale model-to-model cloud comparison

Zone A



z [m]
Reference: 2017/03/28 23:01:00
Compared: 2017/03/29 00:01:00
N
y
(1687) 2017/03/29 00:01:00
dz [m]
grid
(1) The nearest neighbor
(2) Least squares plane
(3) Linear interpolation
(4) Quadratic
(5) 2.5D triangulation
(6) Natural Neighbor Interpolation
(7) Inverse Distance Weight
(8) Multiscale model-to-model cloud comparison

Zone C

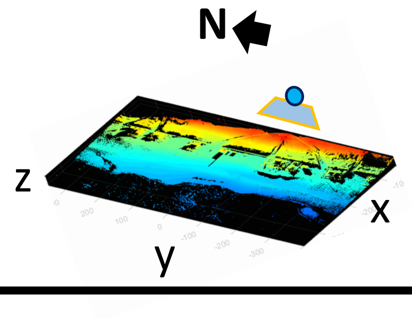


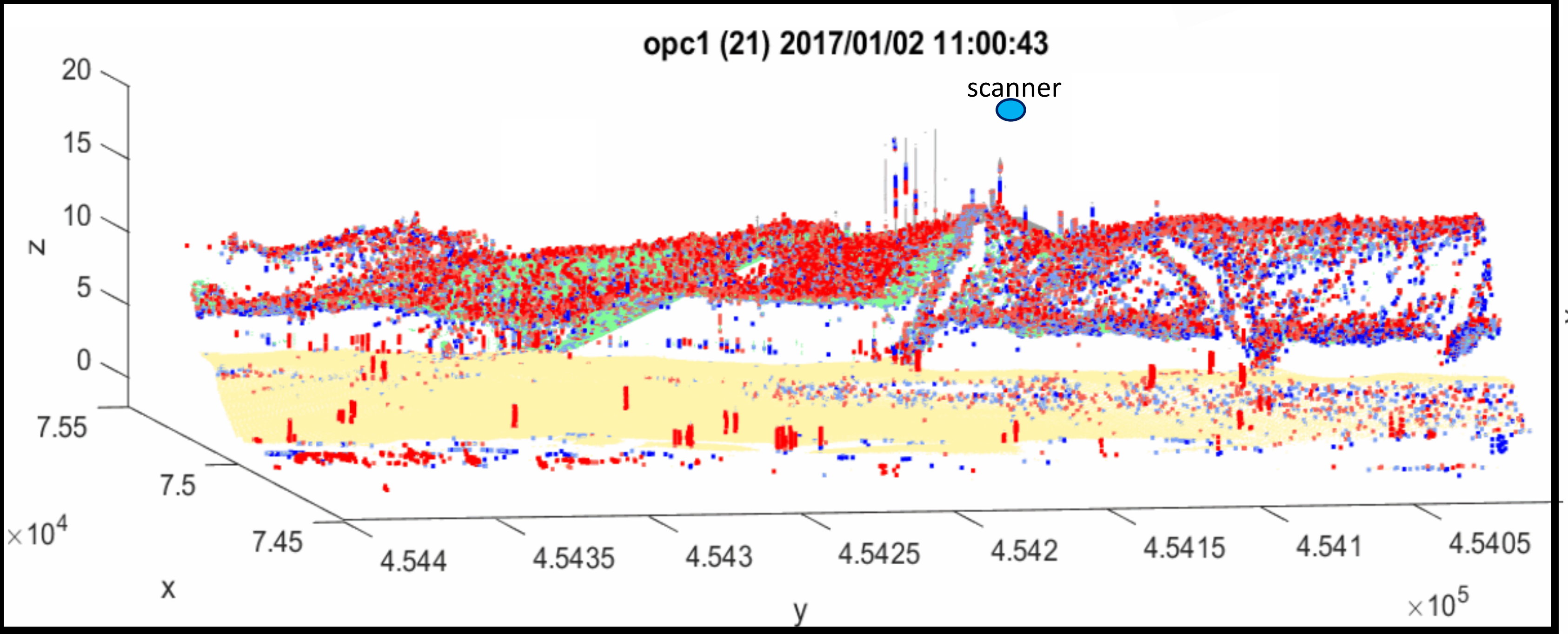
z [m]
Reference: 2017/03/28 23:01:00
Compared: 2017/03/29 00:01:00
N
y
(1687) 2017/03/29 00:01:00
dz [m]
grid
(1) The nearest neighbor
(2) Least squares plane
(3) Linear interpolation
(4) Quadratic
(5) 2.5D triangulation
(6) Natural Neighbor Interpolation
(7) Inverse Distance Weight
(8) Multiscale model-to-model cloud comparison

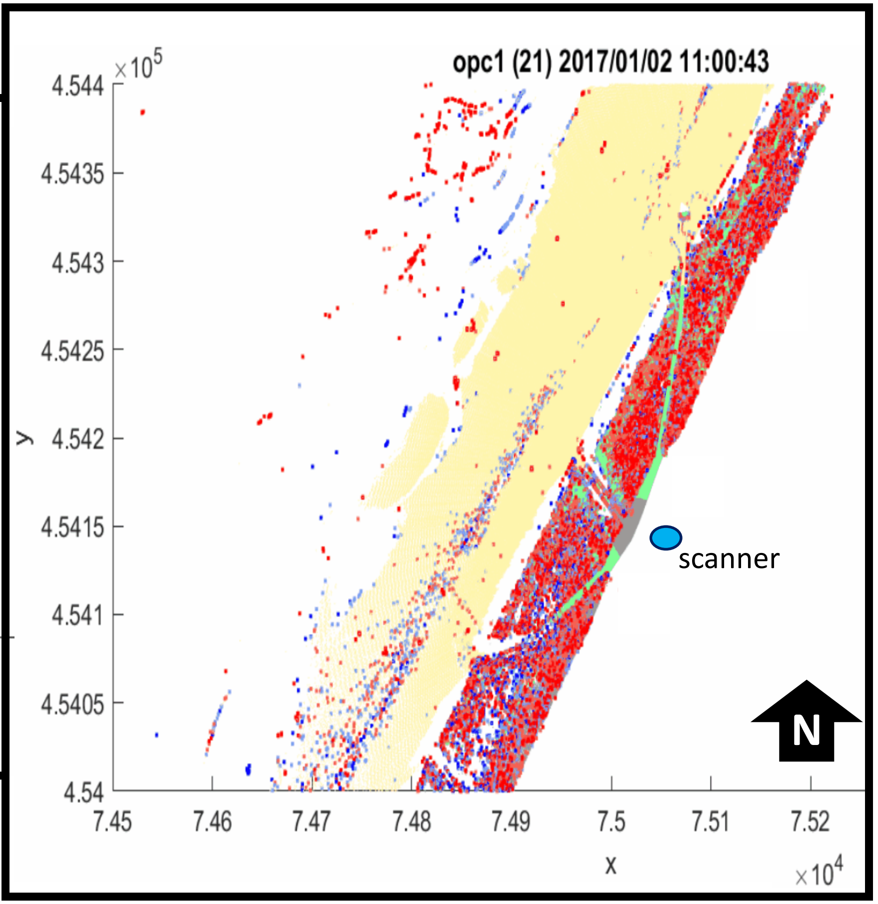
Spatio-temporal CoastScan data

Opc1: 3D Nearest Neighbor(NN)

Changes are detected efficiently







● dz > 0

● dz < 0

3. Fast Space-Filling Curve-based point cloud change detection

nD-PointCloud
nd-pc.org

TU Delft netherlands eScience center

SFC-based point cloud change detection

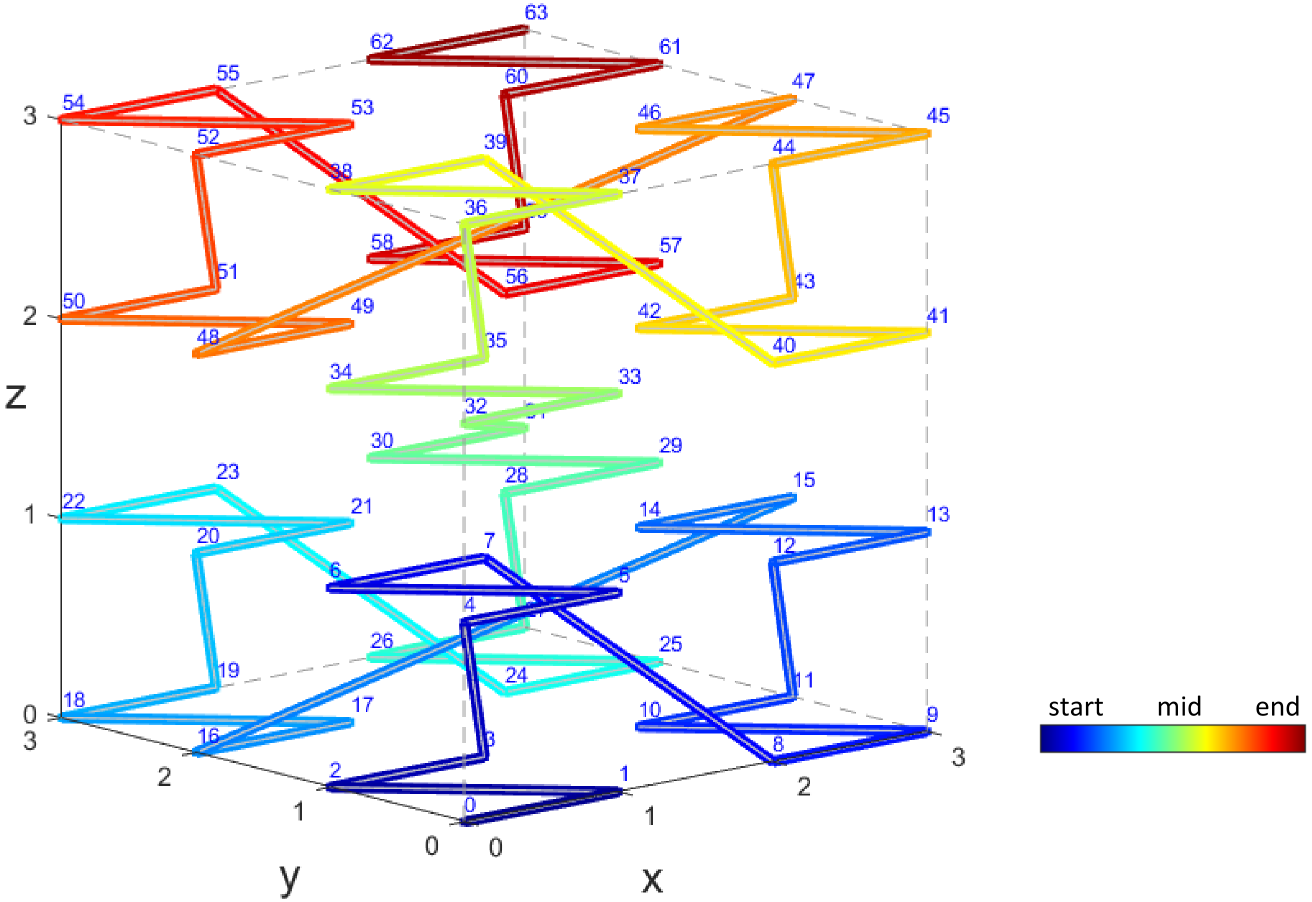
Once the optimal c2c distance method was identified, this phase aimed to develop an efficient implementation.

A 3D SFC-based point cloud method was developed to find the nearest neighbor, taking advantage of SFC's dimensionality reduction, enabling fast and efficient access to multidimensional data.

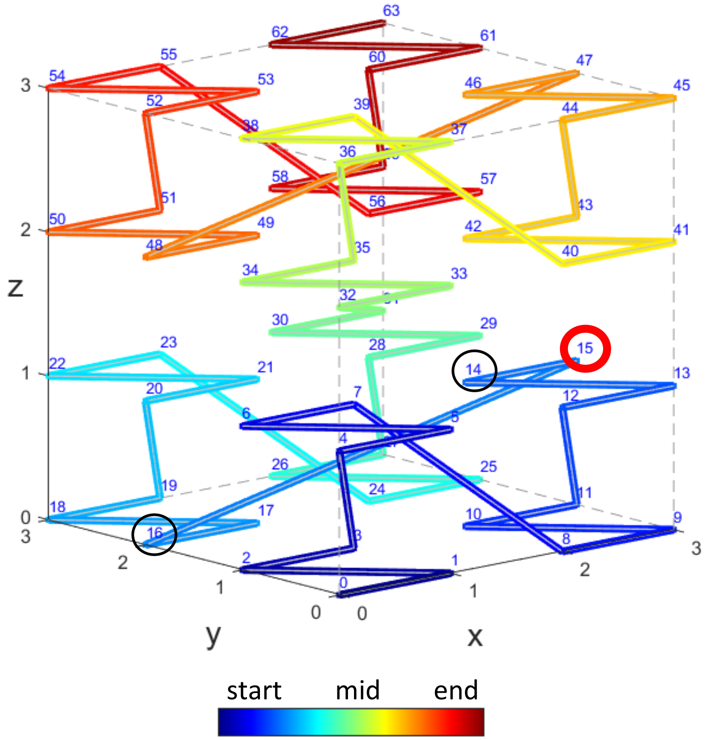
The kd-tree algorithm was the benchmark for finding the 'real' nearest neighbor (NN).

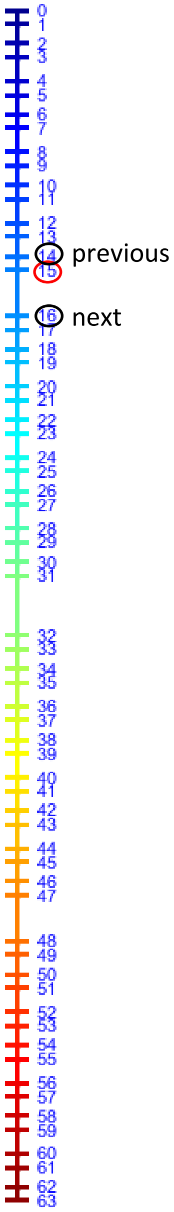
The results of the 3D SFC-based method were compared with those of the kd-tree algorithm.

SFC-based point cloud change detection



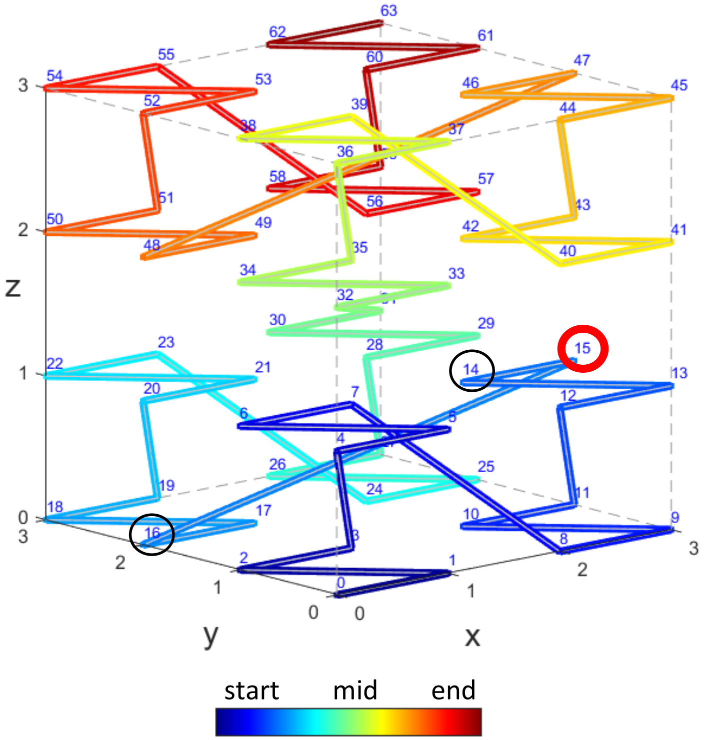
SFC-based point cloud change detection

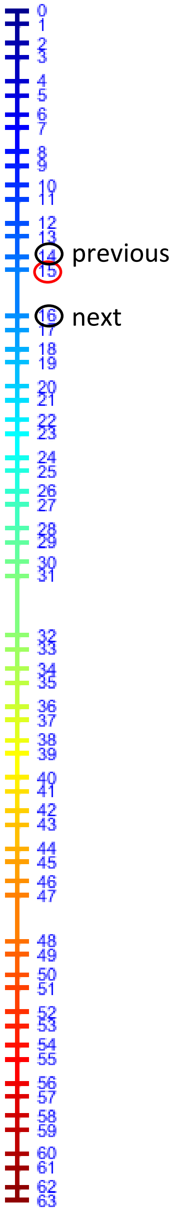




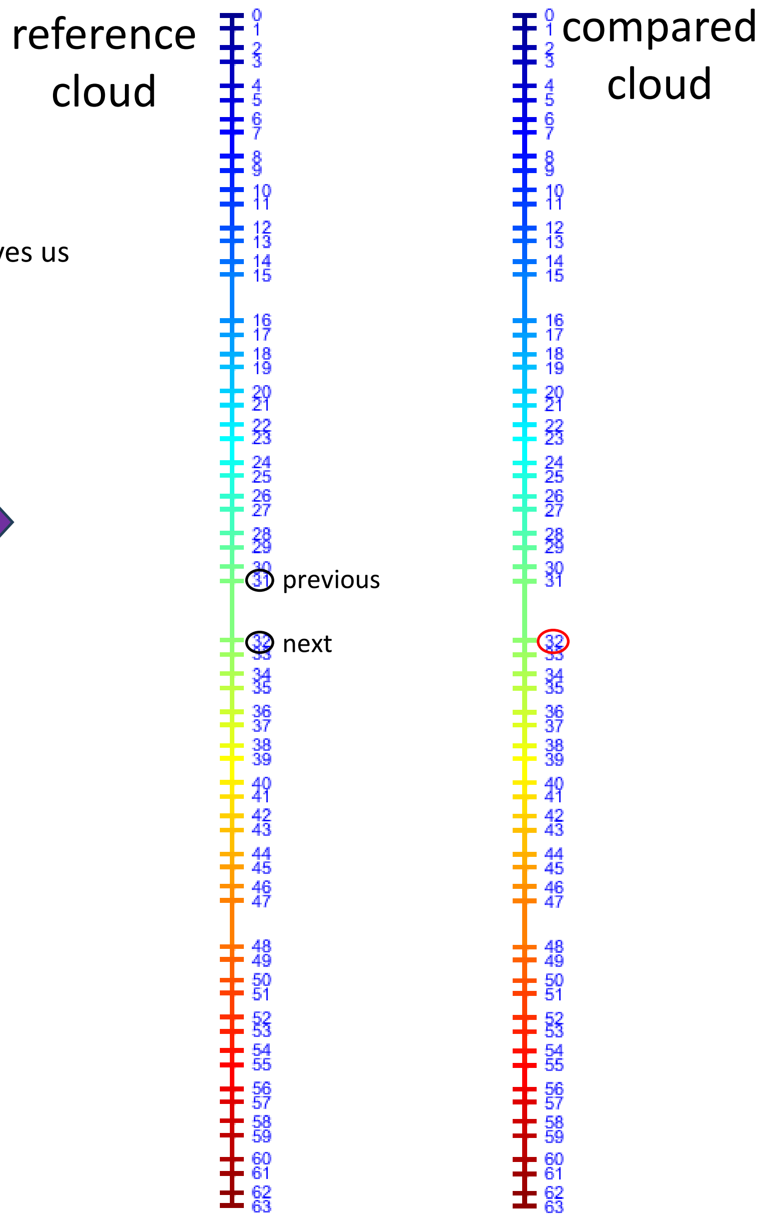
Euclidean distance gives us
the closest neighbor

SFC-based point cloud change detection





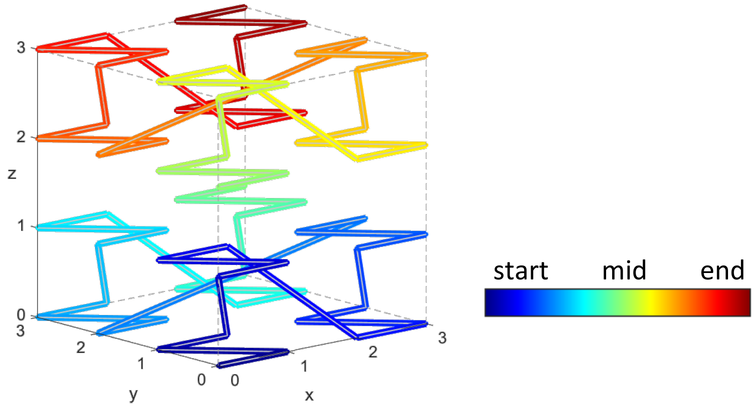
Euclidean distance gives us the closest neighbor



SFC-based point cloud change detection

- Preparation of point cloud, per epoch

1. scaling and offsetting
2. SFC key calculation
3. Sorting SFC key column

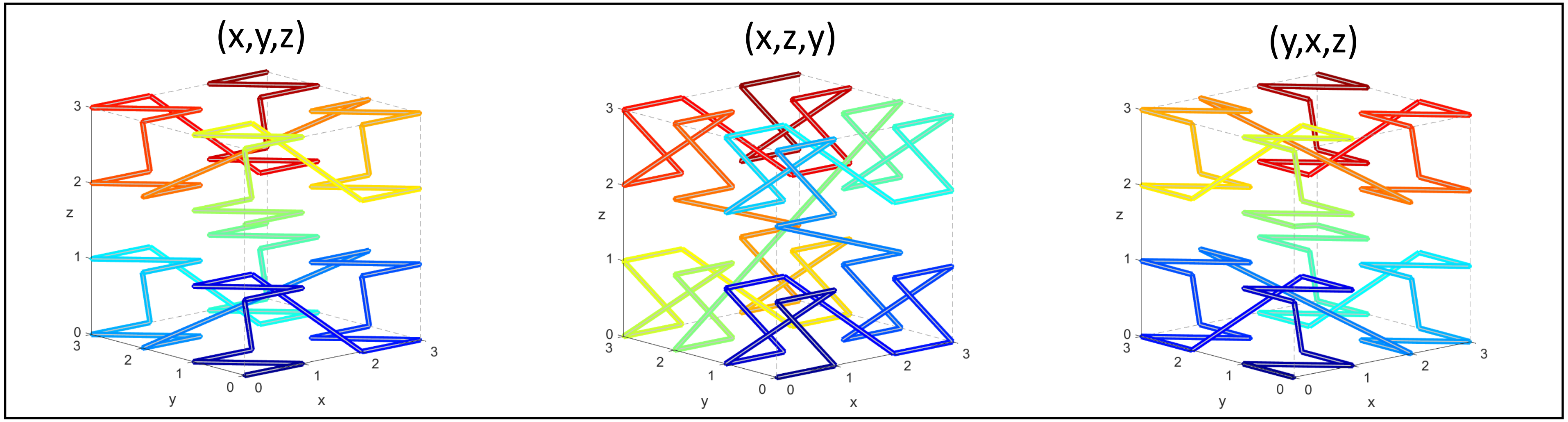


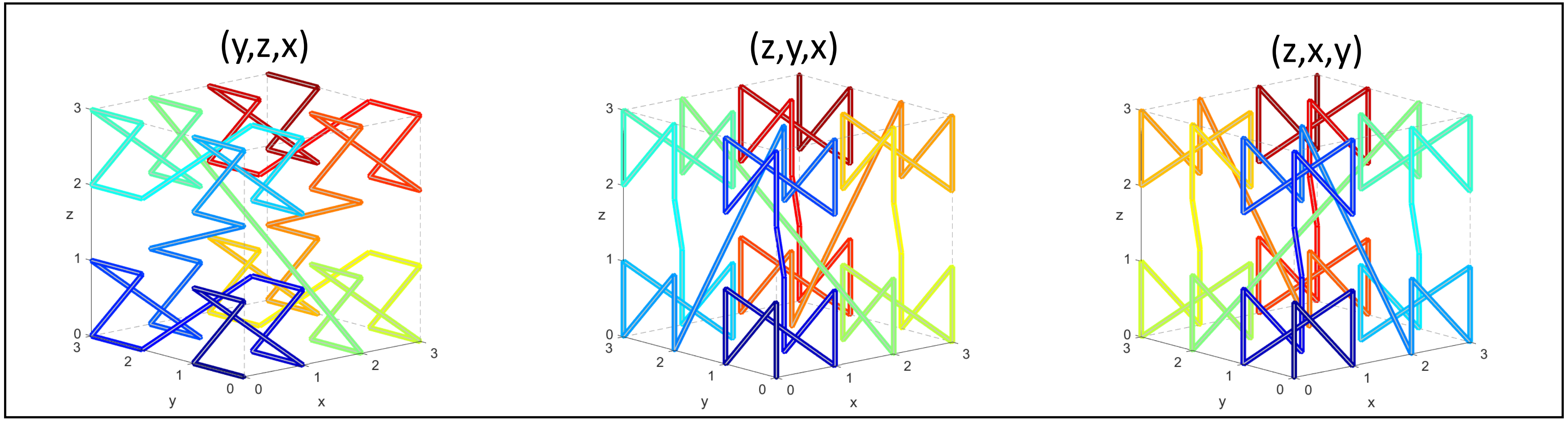
- Batch process NN distance calculation: for every point P in epoch 2:

1. search the previous and next SFC key of the calculated SFC key in epoch 1
2. decode and unscale these two SFC keys (previous and next)
3. find the approximate nearest neighbor (NN) of the 2 candidates
4. output d and dz

SFC-based point cloud change detection

order for SFC key calculation



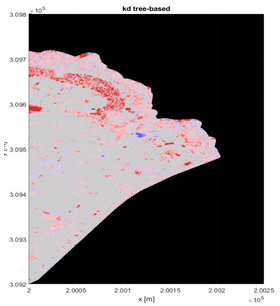


Does it matter?

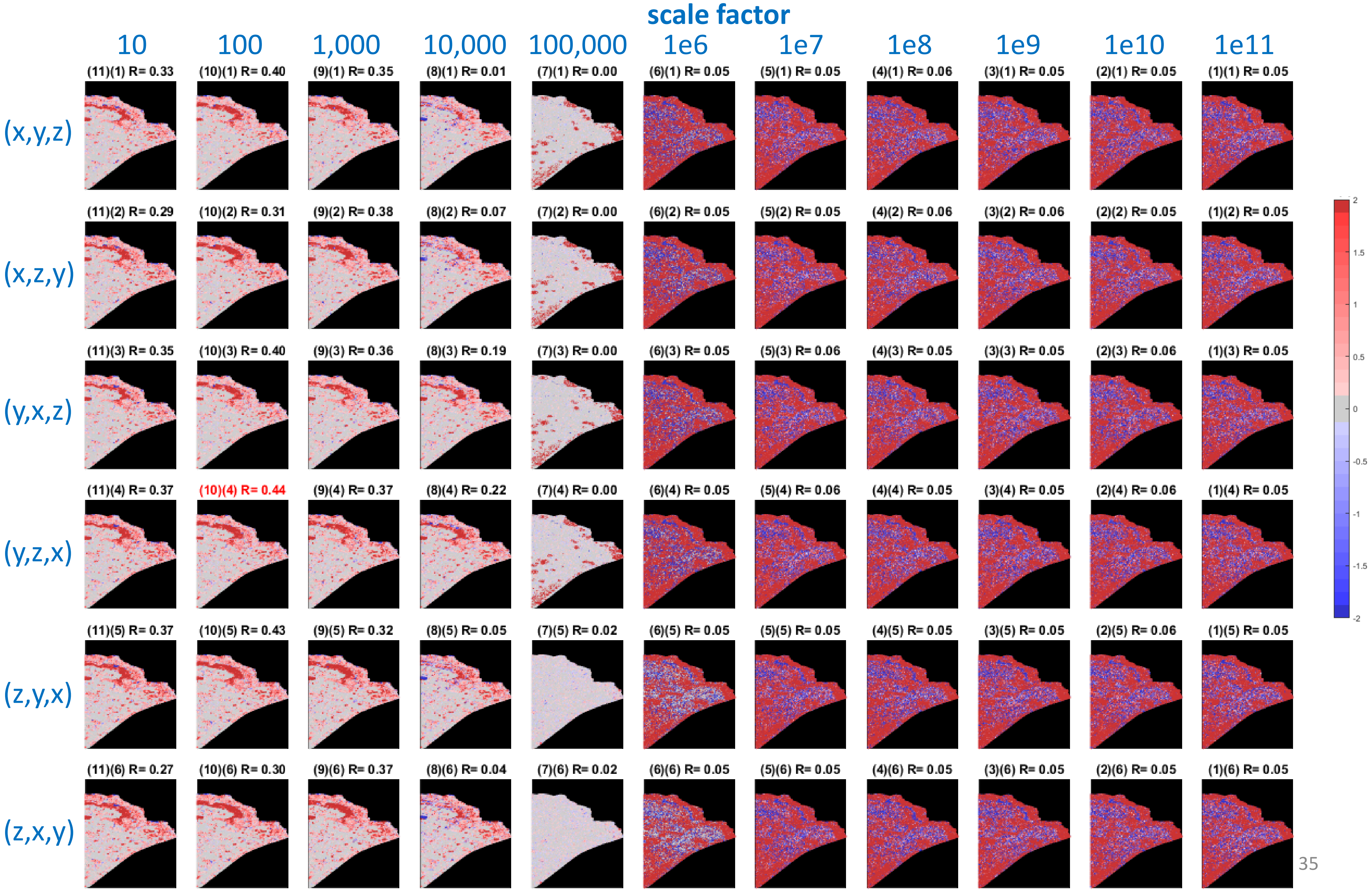
TU Delft

netherlands eScience center

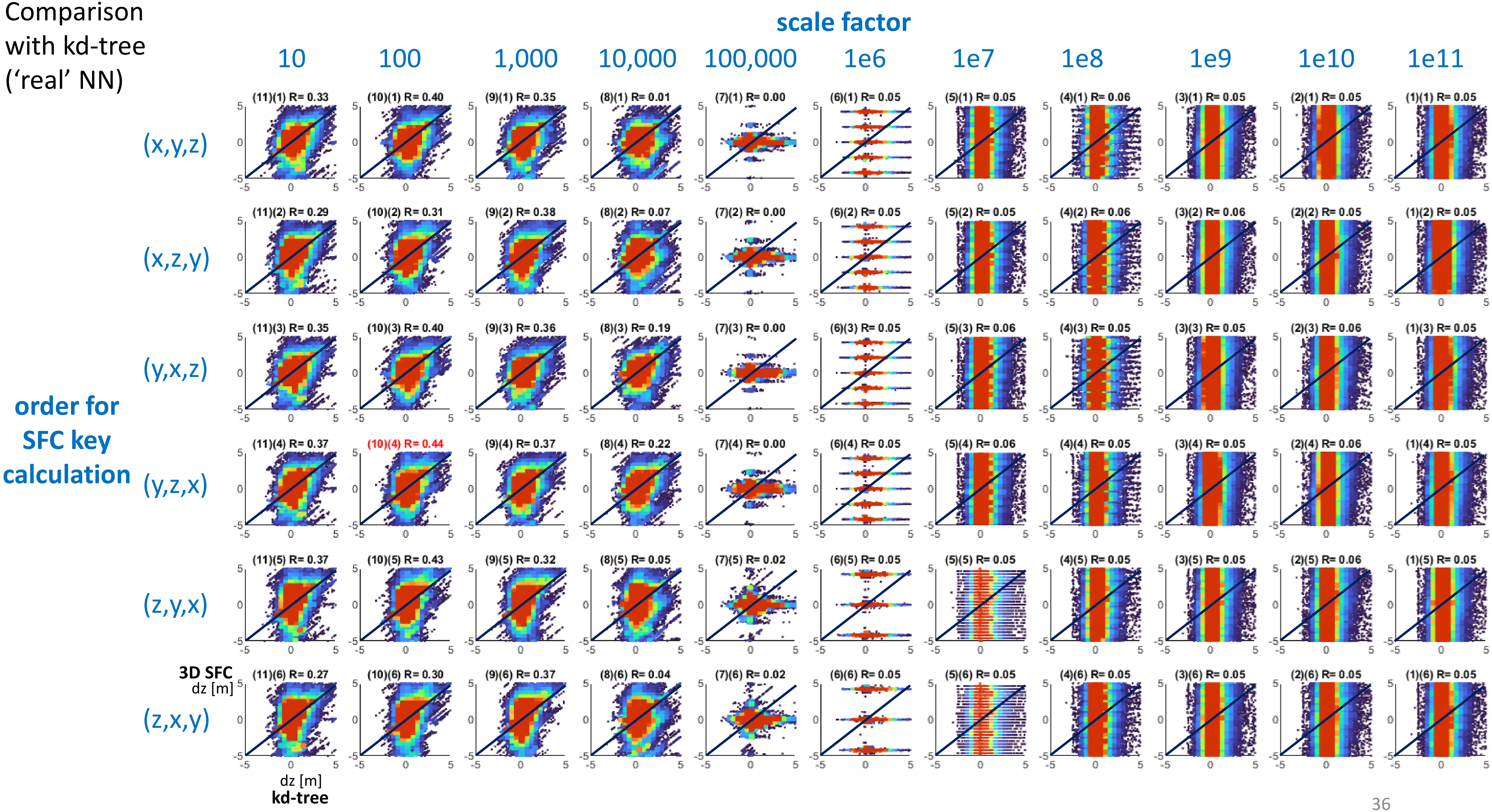
Comparison with kd-tree ('real' NN)

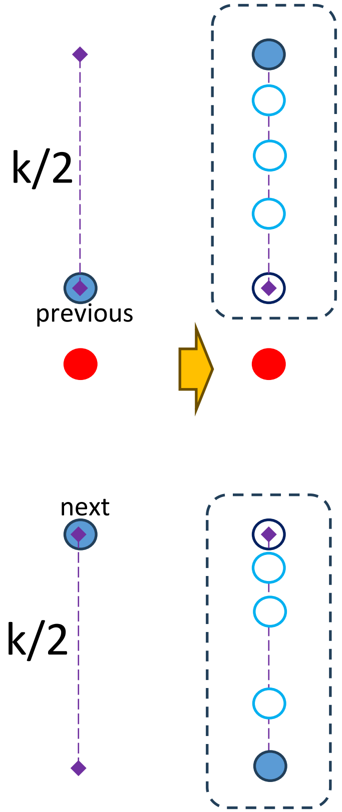


order for SFC key calculation



Comparison with kd-tree ('real' NN)





Euclidean distance to identify the closest neighbor

scale factor 10

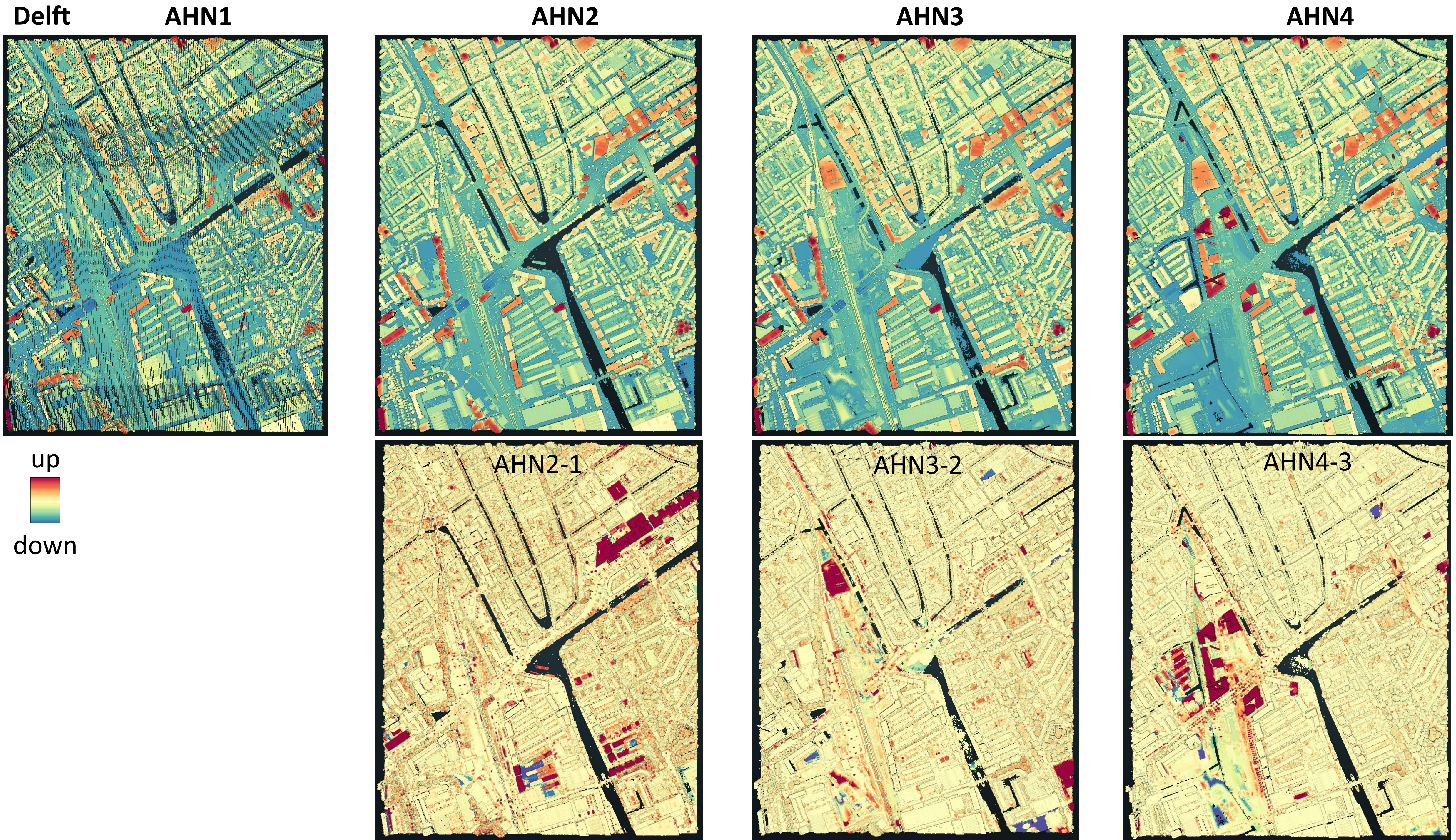
Order for SFC key calculation	k points (neighborhood)	R
(x,y,z)	8	0.3888
(x,y,z)	64	0.4722
(x,y,z)	512	0.6258
(x,y,z)	4,096	0.8101
(x,y,z)	32,768	0.9754
(x,y,z)	**262,144**	**0.9985**
(x,z,y)	8	0.3486
(x,z,y)	64	0.3957
(x,z,y)	512	0.4974
(x,z,y)	4,096	0.7848
(x,z,y)	32,768	0.9633
(x,z,y)	**262,144**	**0.9888**
(y,x,z)	8	0.4260
(y,x,z)	64	0.5130
(y,x,z)	512	0.6568
(y,x,z)	4,096	0.8185
(y,x,z)	32,768	0.9751
(y,x,z)	**262,144**	**0.9990**
(y,z,x)	8	0.4538
(y,z,x)	64	0.5395
(y,z,x)	512	0.6913
(y,z,x)	4,096	0.8772
(y,z,x)	32,768	0.9800
(y,z,x)	**262,144**	**0.9990**
(z,y,x)	8	0.4178
(z,y,x)	64	0.5598
(z,y,x)	512	0.6862
(z,y,x)	4,096	0.7971
(z,y,x)	32,768	0.9808
(z,y,x)	**262,144**	**0.9990**
(z,x,y)	8	0.3147
(z,x,y)	64	0.4099
(z,x,y)	512	0.5460
(z,x,y)	4,096	0.7813
(z,x,y)	32,768	0.9739
(z,x,y)	**262,144**	**0.9910**

scale factor 100

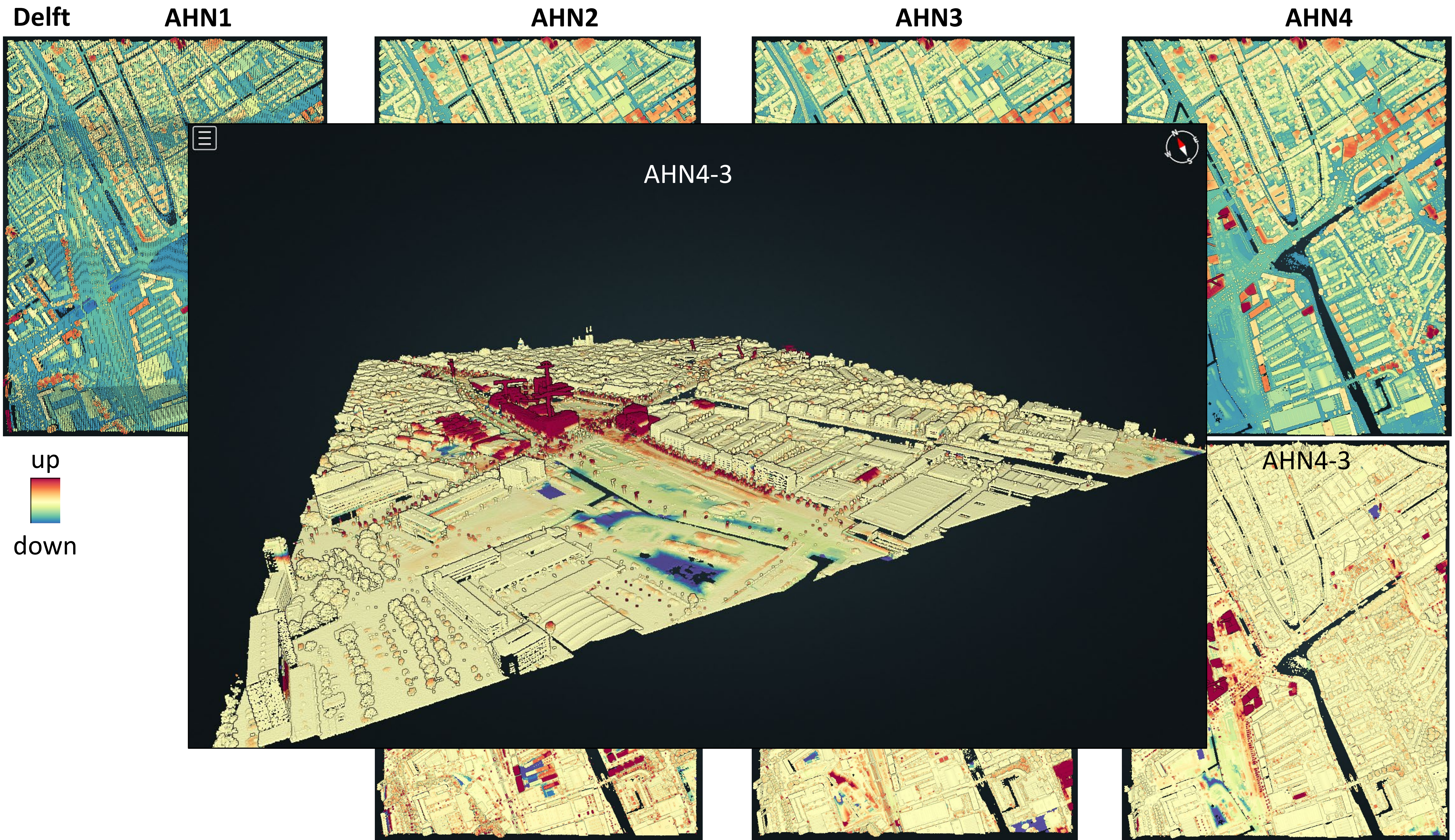
Comparison with kd-tree ('real' NN)

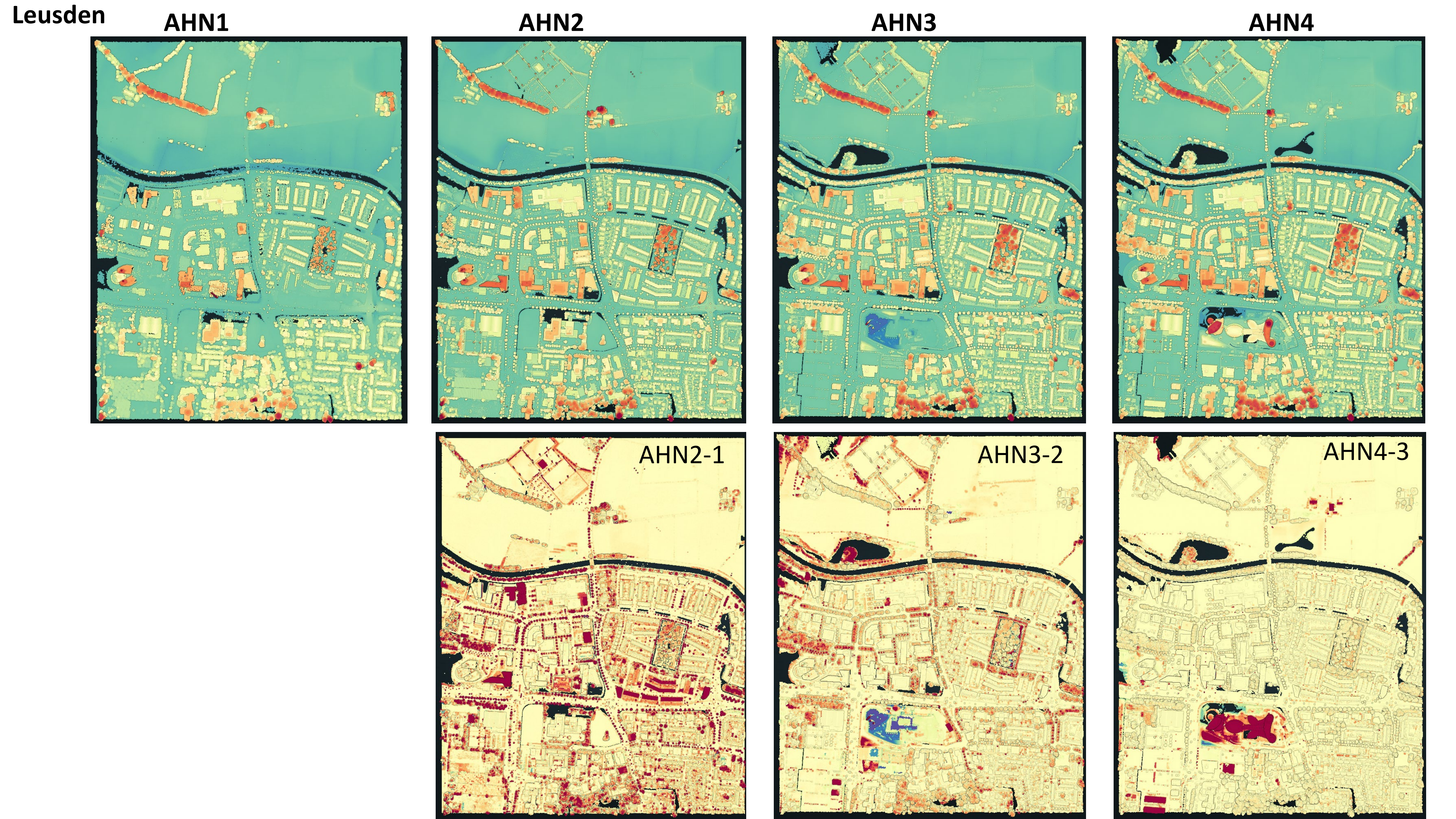
Order for SFC key calculation	k points (neighborhood)	R
(x,y,z)	8	0.4460
(x,y,z)	64	0.5379
(x,y,z)	512	0.6229
(x,y,z)	4,096	0.8697
(x,y,z)	32,768	0.9655
(x,y,z)	**262,144**	**0.9969**
(x,z,y)	8	0.3566
(x,z,y)	64	0.4077
(x,z,y)	512	0.5022
(x,z,y)	4,096	0.6299
(x,z,y)	32,768	0.9850
(x,z,y)	**262,144**	**0.9969**
(y,x,z)	8	0.4391
(y,x,z)	64	0.5549
(y,x,z)	512	0.7004
(y,x,z)	4,096	0.8731
(y,x,z)	32,768	0.9756
(y,x,z)	**262,144**	**0.9964**
(y,z,x)	8	0.4650
(y,z,x)	64	0.5949
(y,z,x)	512	0.8029
(y,z,x)	4,096	0.9243
(y,z,x)	32,768	0.9820
(y,z,x)	**262,144**	**0.9964**
(z,y,x)	8	0.5030
(z,y,x)	64	0.5923
(z,y,x)	512	0.7316
(z,y,x)	4,096	0.9303
(z,y,x)	32,768	0.9824
(z,y,x)	**262,144**	**0.9966**
(z,x,y)	8	0.3510
(z,x,y)	64	0.4325
(z,x,y)	512	0.5148
(z,x,y)	4,096	0.8740
(z,x,y)	32,768	0.9862
(z,x,y)	**262,144**	**0.9969**

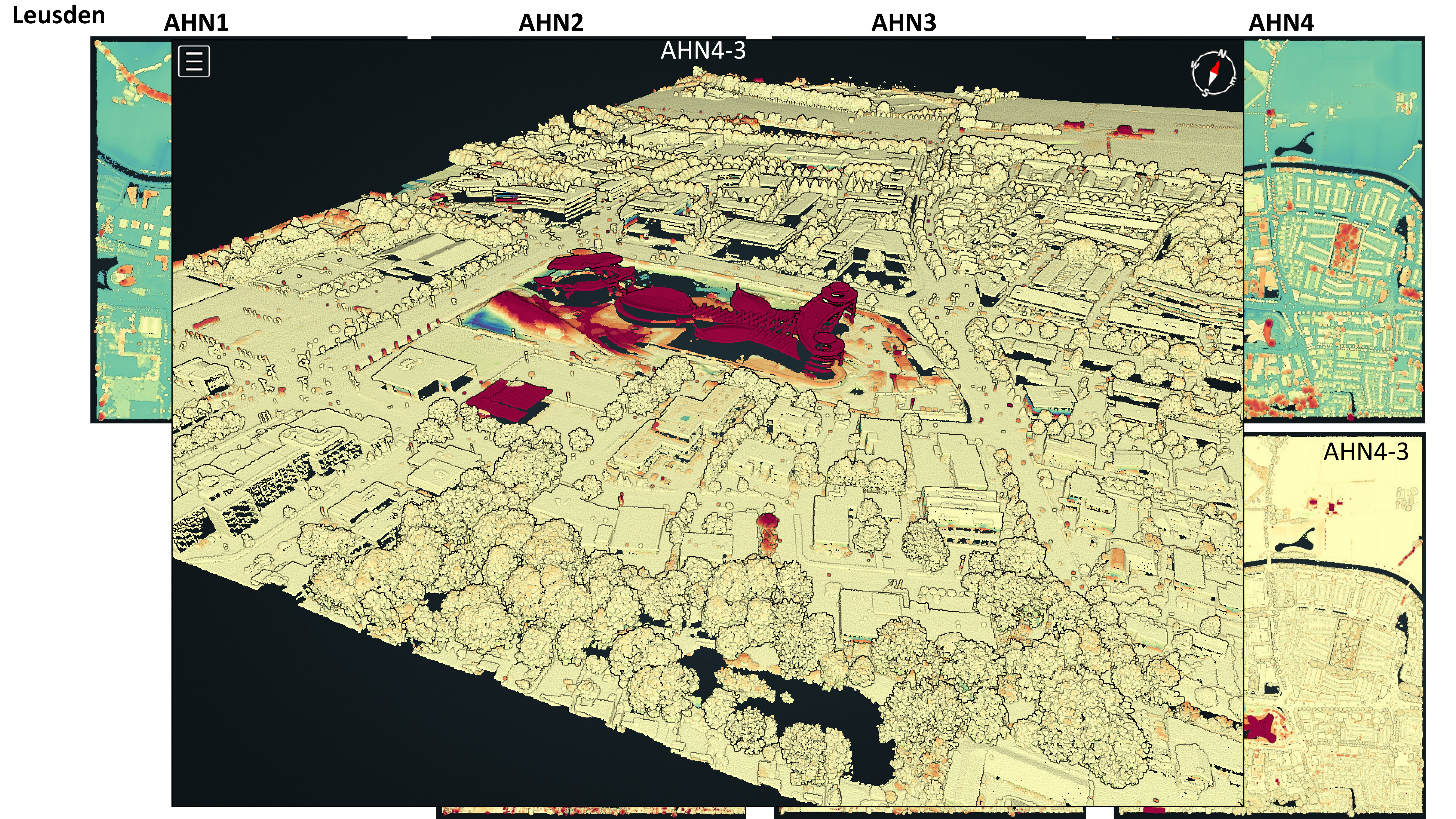
TU Delft   netherlands eScience cer



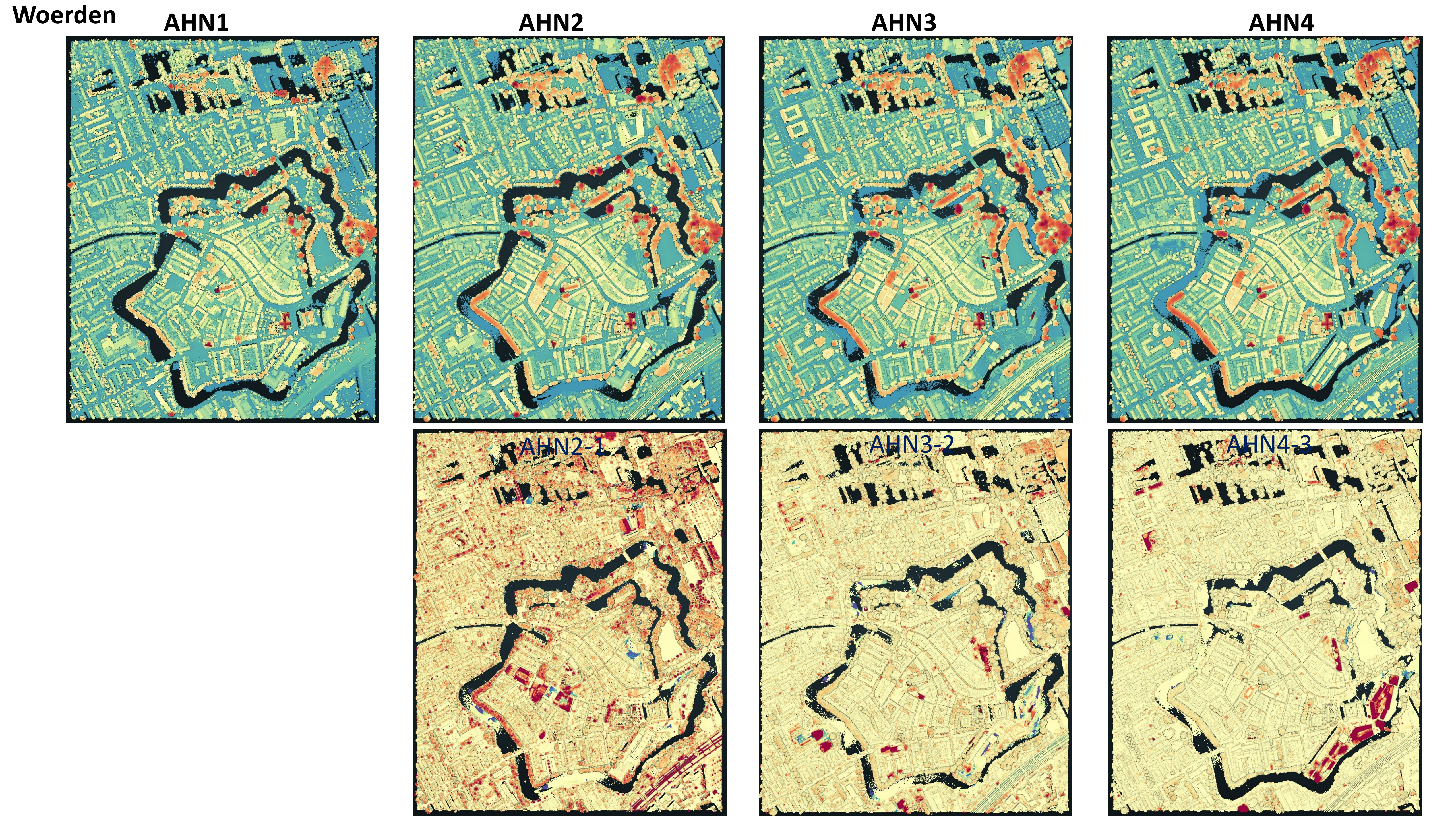
Delft
AHN1
AHN2
AHN3
AHN4
up
down
AHN2-1
AHN3-2
AHN4-3


Delft
AHN1
AHN2
AHN3
AHN4
AHN4-3
AHN4-3
up
down

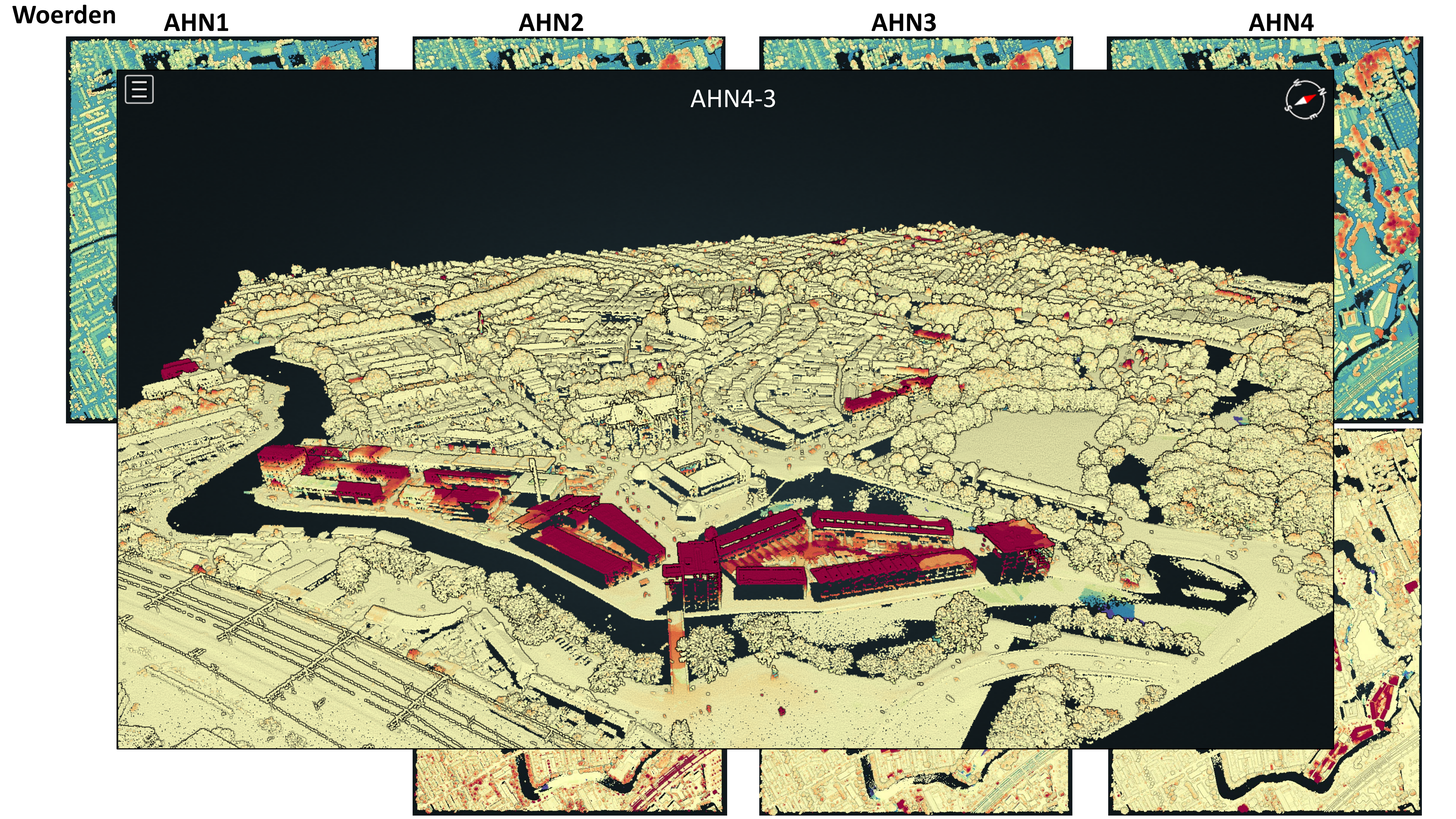

Leusden
AHN1
AHN2
AHN3
AHN4
AHN2-1
AHN3-2
AHN4-3



Leusden
AHN1
AHN2
AHN3
AHN4
AHN4-3
N
E
S
W
AHN4-3



Woerden
AHN1
AHN2
AHN3
AHN4
AHN2-1
AHN3-2
AHN4-3


Woerden
AHN1
AHN2
AHN3
AHN4
AHN4-3

nd-pc.org

4 Conclusions

- 8 methods for change detection analysis with spatiotemporal point clouds.
- Nearest Neighbor is the best candidate for the c2c distance calculation.
- Development of an SFC-based c2c distance calculation method for fast implementation in massive point clouds.

5 Future work

- Finishing calculation of changes for AHN 1 to 4
- Visualizing the changes in Potree viewer

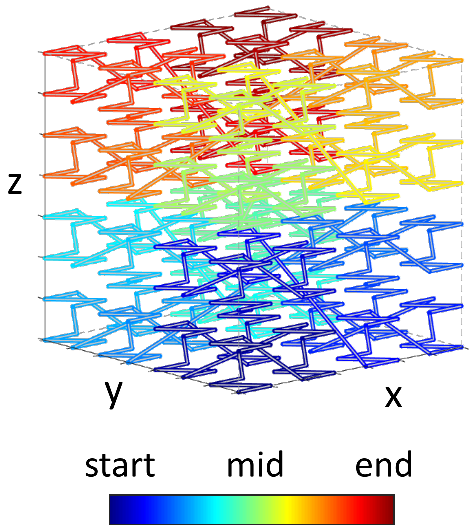
- Publications:
  - Cloud-to-cloud method for change detection in spatio-temporal point clouds: CoastScan data

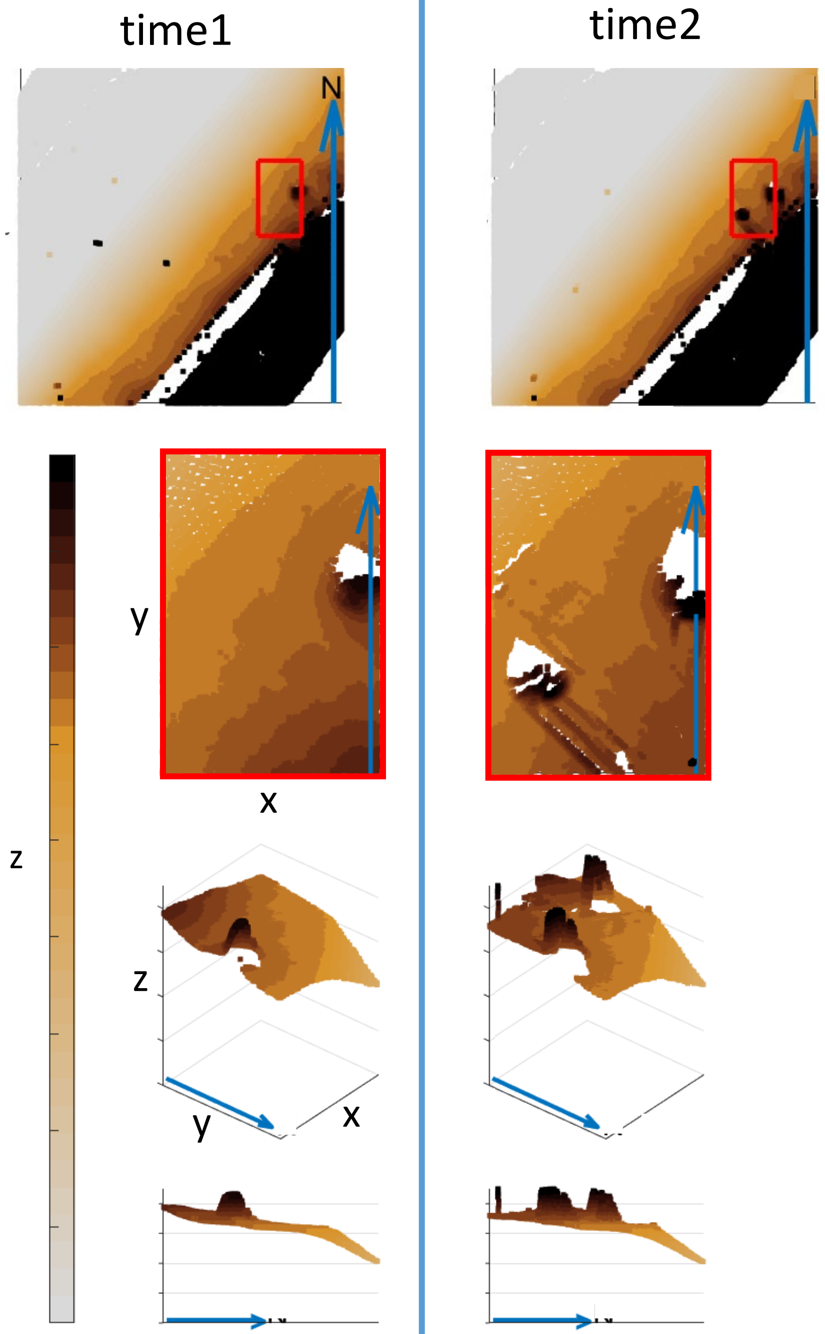
  - Fast Space Filling Curve-based point cloud change detection: development and implementation

*Intelligent method for monitoring and predicting changes in spatio-temporal point cloud data

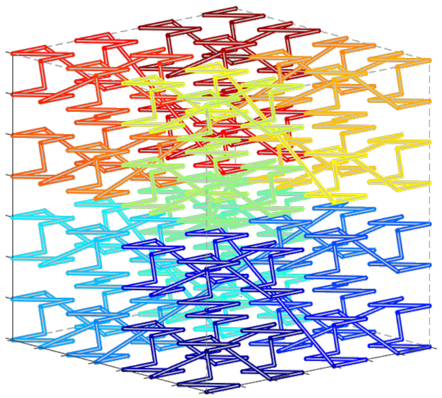
nD-PointCloud
nd-pc.org

3D_space-filling_curve1





3D_space-filling_curve2



*intelligent indicates using ML algorithms to segment, characterise, and predict changes

TU Delft
netherlands eScience center

Thanks!

Fast SFC-based point cloud change detection

Vitali Diaz Mercado
v.diazmercado@tudelft.nl

nD-PointCloud
nd-pc.org

TU Delft

netherlands eScience center