

Data cleaning CoastScan data & change detection in spatio-temporal point clouds

Vitali Díaz Mercado



Delft, January 23, 2023

Outline

1. Introduction
2. CoastScan data
3. Data preparation
4. Change detection algorithms
5. Preliminary results
6. Conclusions and future steps

1. Introduction

The spatio-temporal analysis allows to detect changes, hotspots, spatial extent, change rates, durations of different phenomena over space and time.

The detail of this analysis relies on temporal and spatial resolution, which is crucial in certain phenomena.

Point cloud data allows detailed analysis due to the massive amount of information that can be collected, but...

most of the existing spatio-temporal analysis methods and examples have been developed/applied to raster and vector data,

the use in point cloud data has not been fully explored, so there are still questions to be answered.

1. Introduction

As initial steps:

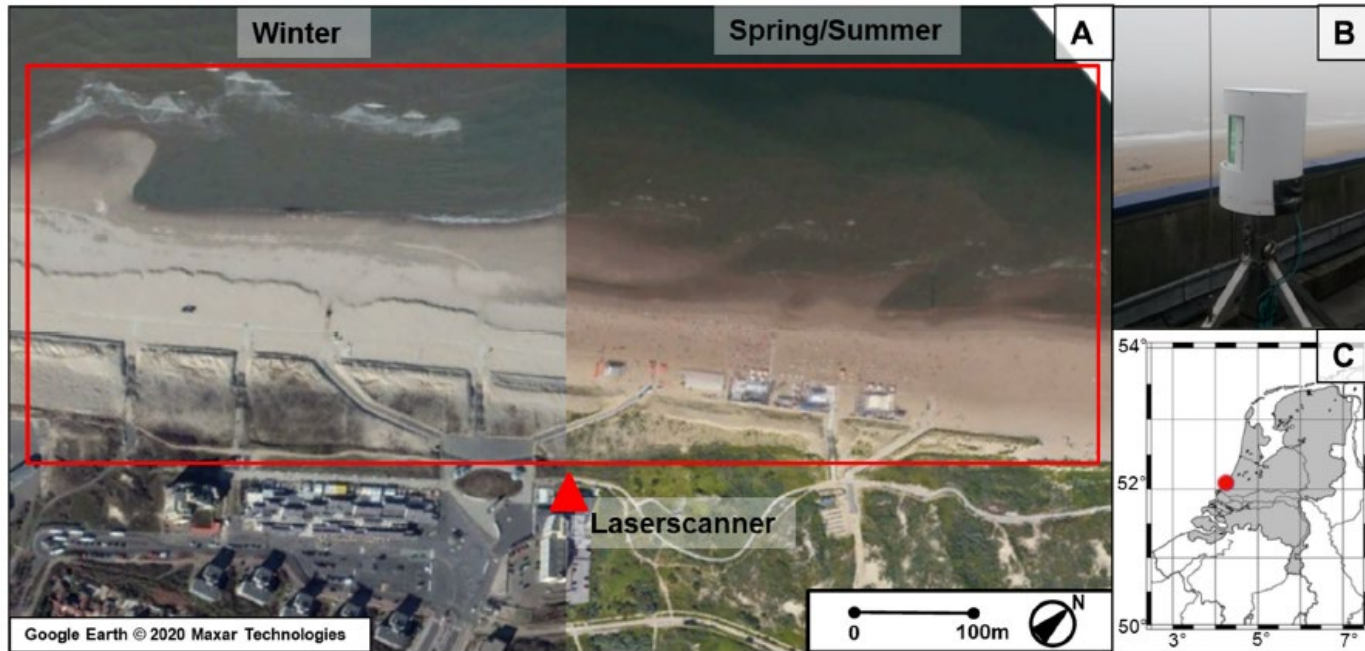
Four use cases have been proposed

- Global reservoir monitor (with Deltares)
- Coast monitor (NL/spot) (with Deltares)
- Automatic identification systems (AIS) data
- Indoor navigation

Explore the file-based application to gain experience and move to a more database-based solution

2. Case study

Kijkduin beach-dune system



A high-resolution 4D geospatial laser scan dataset of the Kijkduin beach-dune system, The Netherlands

Vos et al. (2022)

(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).

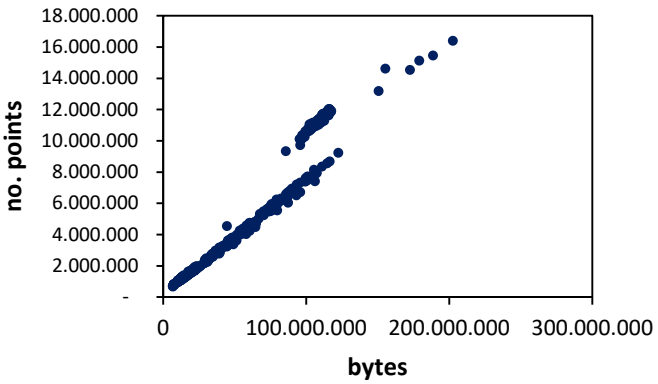
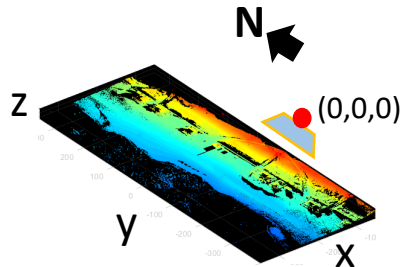
(C) The location of Kijkduin (52.07°N, 4.22°E) in The Netherlands. (Vos et al., 2022)

Stationary terrestrial laser scanner (TLS) (Riegl VZ-2000)

- Period: Six months, 2016/11/11 to 2017/05/26
- Time resolution: hourly (~1 million points each)
- Coordinate reference system: local Cartesian coordinates with the TLS sensor as origin (0,0,0)

3. Data preparation

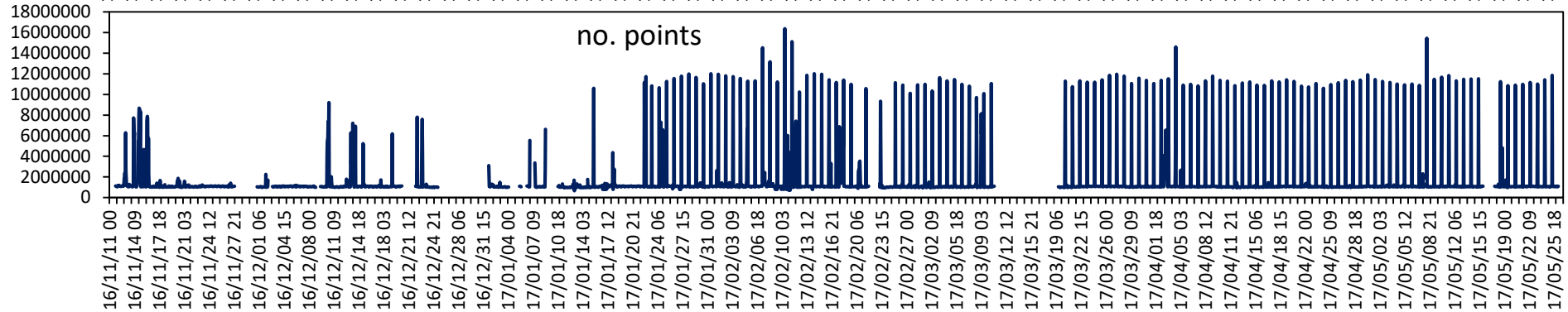
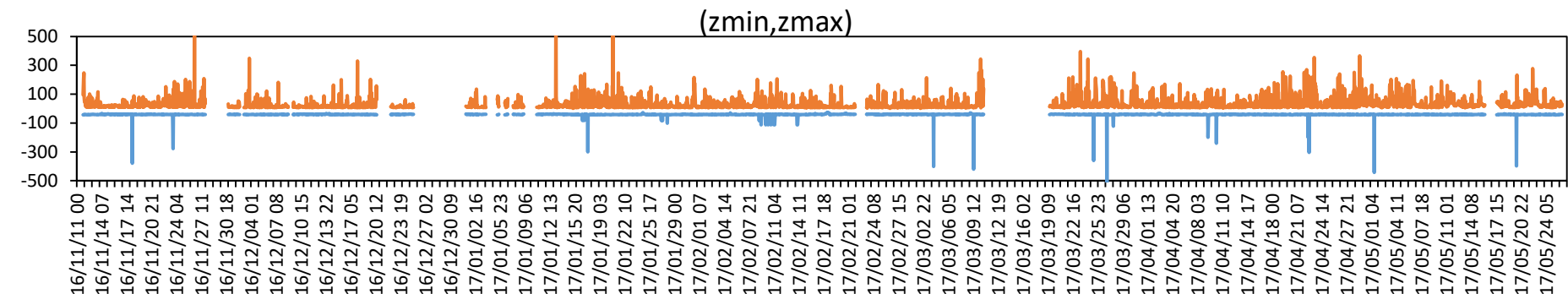
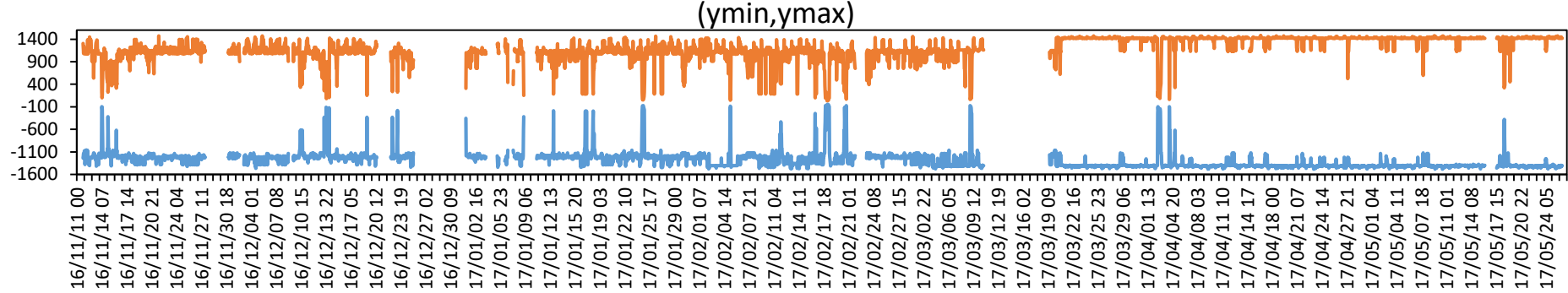
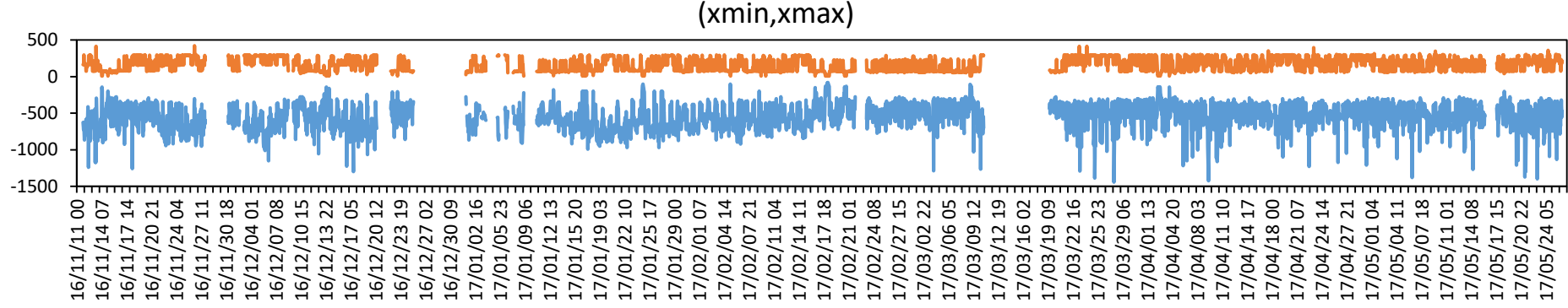
Some statistics



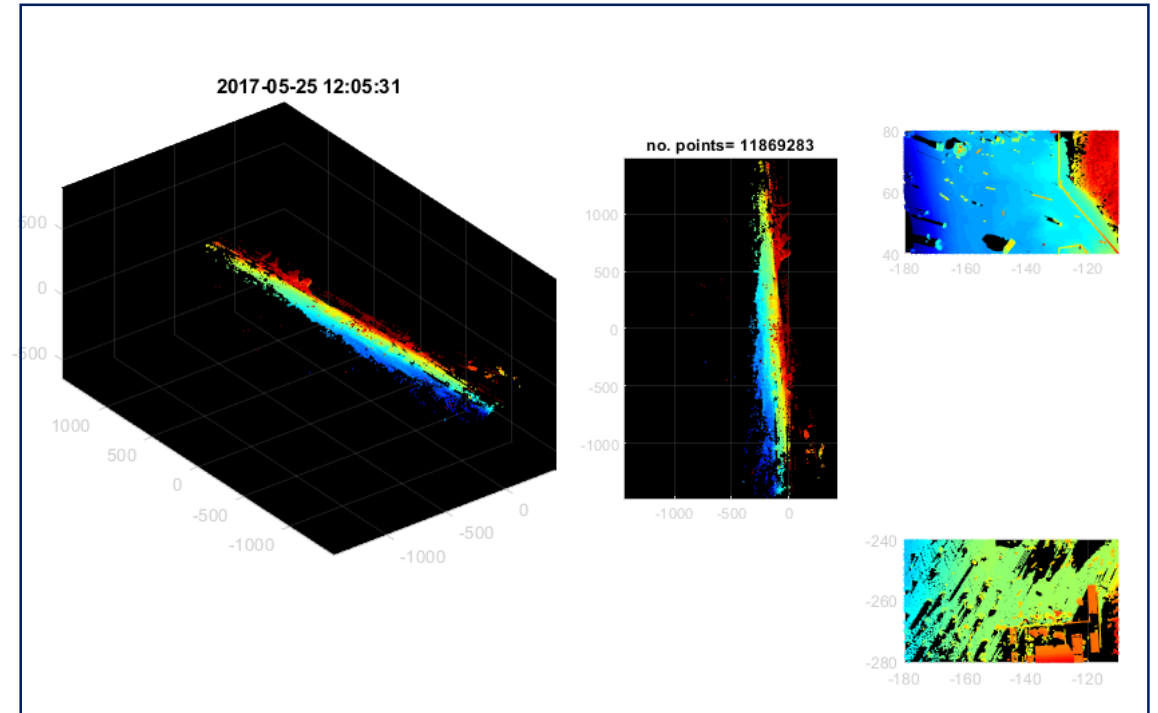
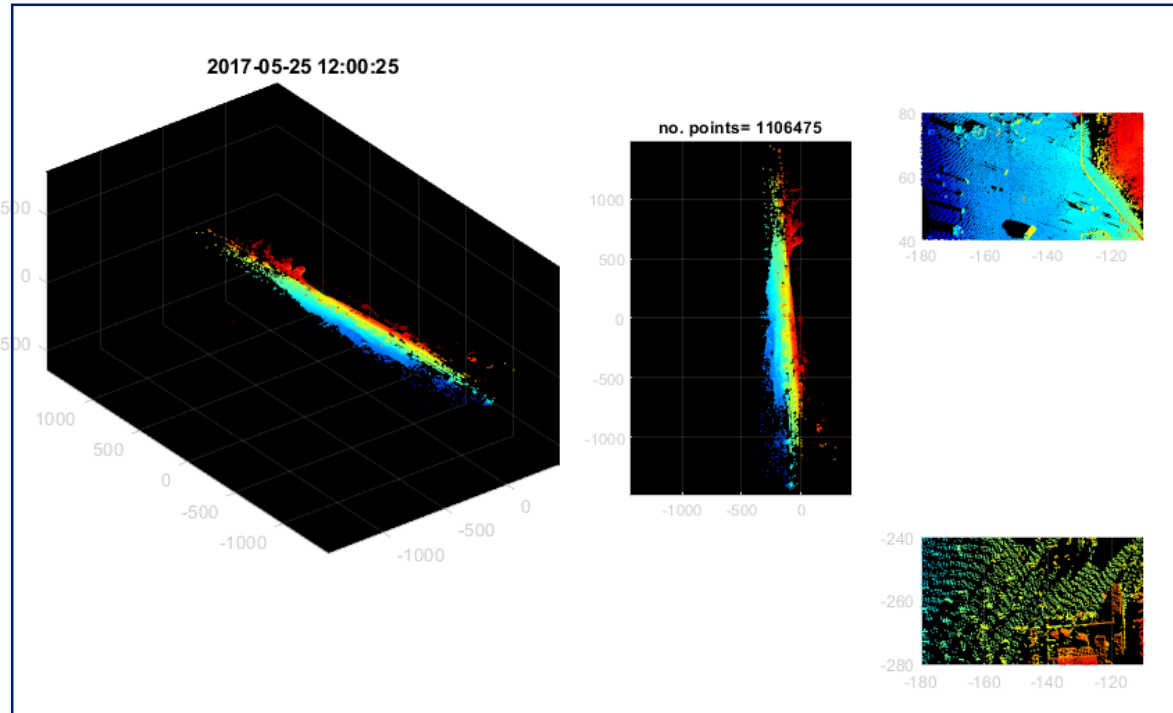
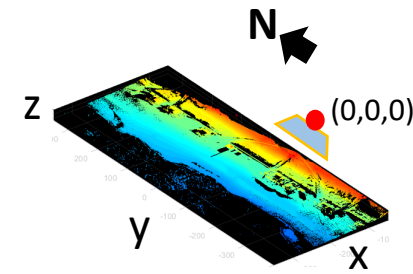
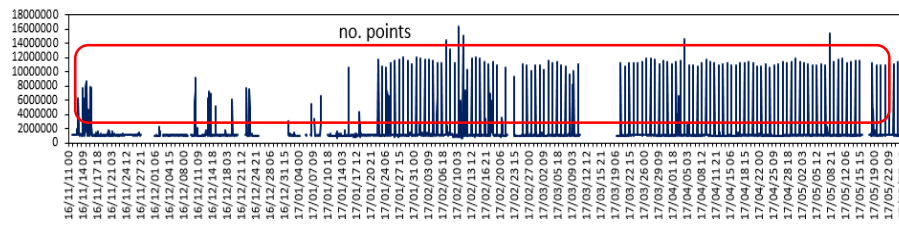
12/11/2016 to 26/05/2017

4,082 LAZ files
3,950 low resolution
132 high resolution

Saved in 178 folders
24 LAZ files per folder



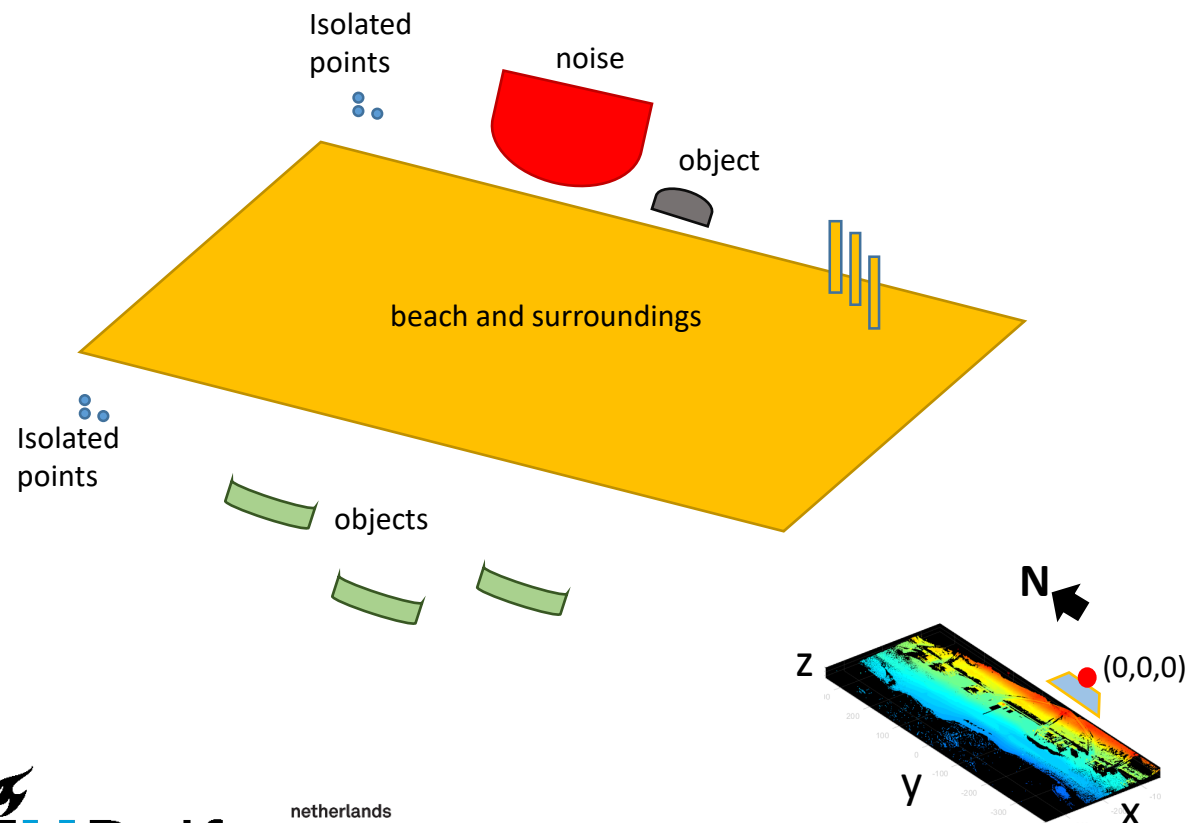
Two samples taken at the same hour
The first with the least amount is taken first



Every 24 hours at 12:00 more points are collected with a second scan.
More or less 10 times more points.
So, a 12 o'clock, two scans are available.

Data filtering/cleaning

Conceptual model



The hypothesis is that the clustering of the points allows the identification of the different groups/objects (including the noise) and, with this, select those that refer to the terrain to calculate the changes.

Data filtering/cleaning

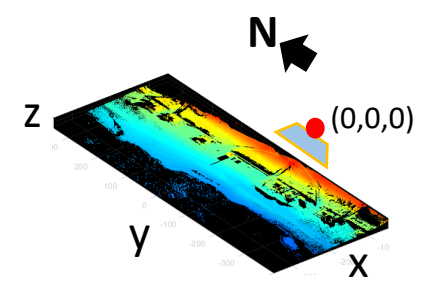
Methodology

- Cluster calculation, using the distance (d) between points as the inclusion/exclusion parameter

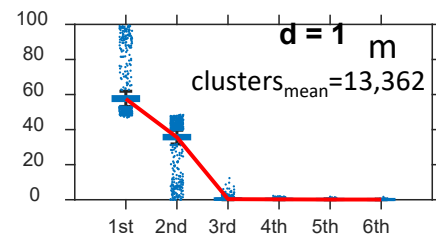
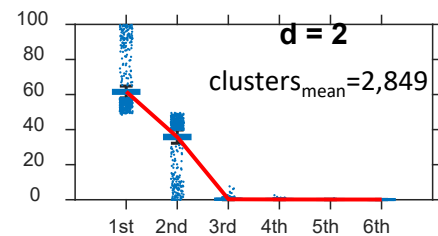
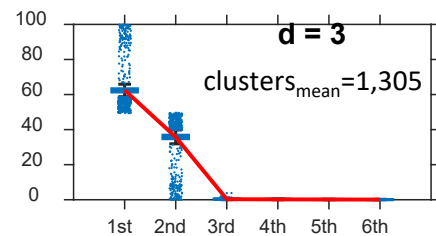
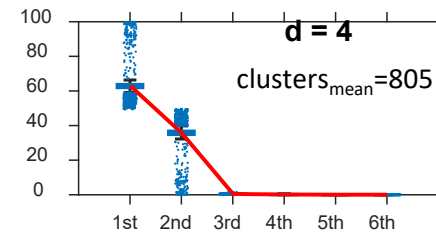
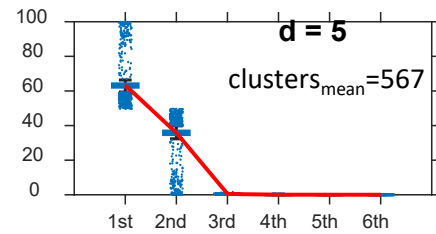
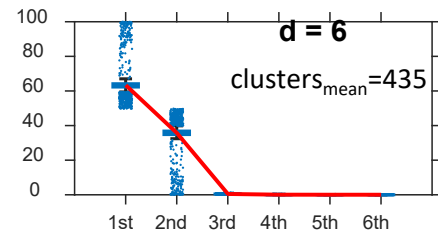
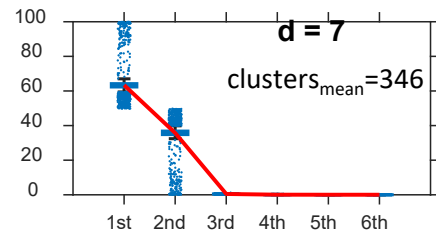
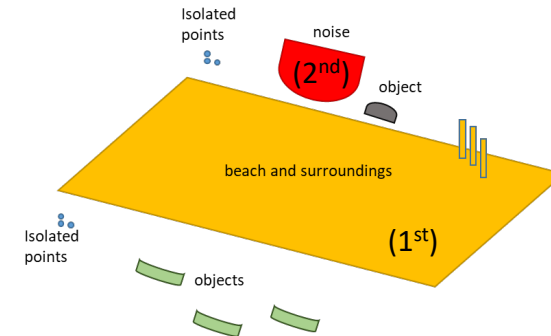
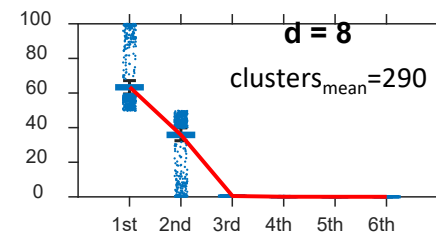
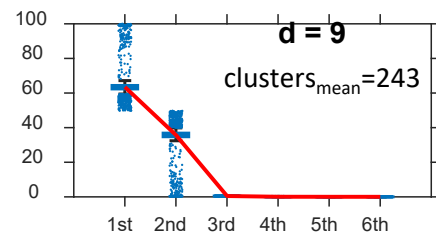
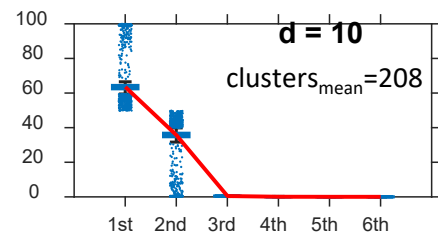
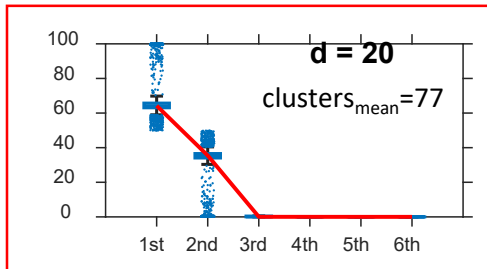
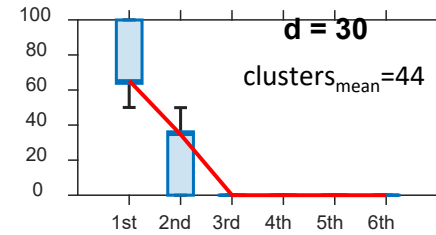
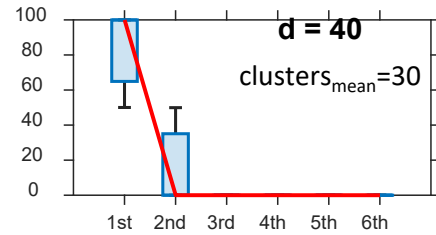
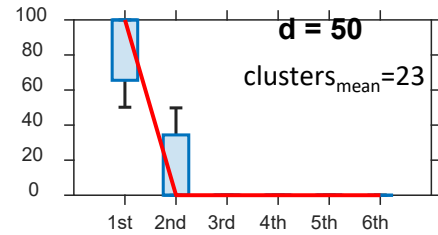
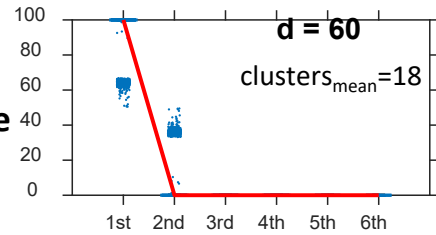
$d=100,90,80,70,60,50,40,30,20,10,9,8,7,6,5,4,3,2,1$ m

- For each cluster, the following was calculated
 - number of points
 - points percentage
 - xmin, xmax
 - ymin, ymax
 - zmin, zmax
 - centroid (x,y,z)
 - time cost

Data filtering/cleaning

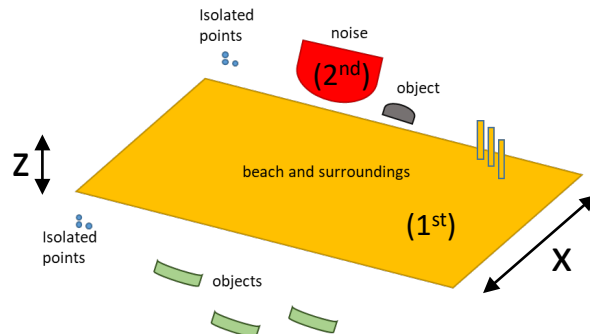
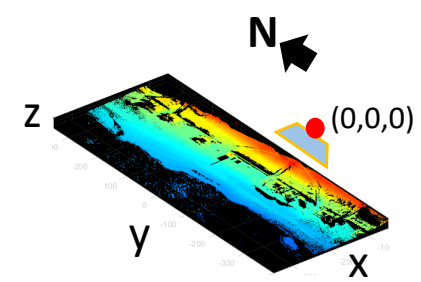


Points percentage



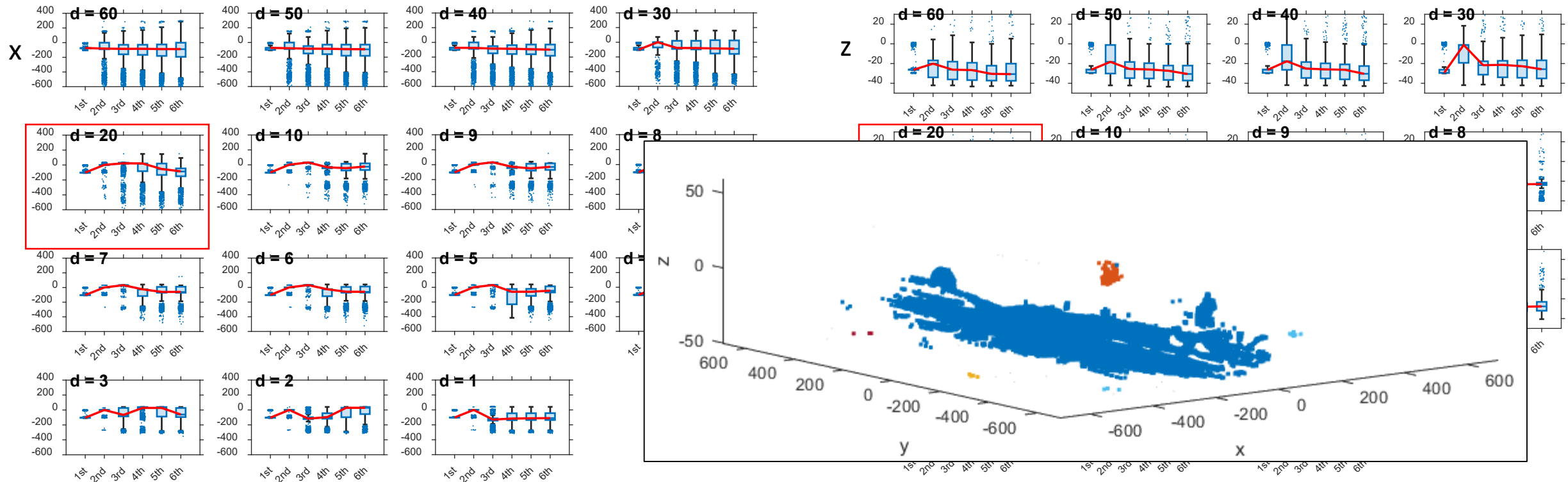
Cluster size

Data filtering/cleaning



X_{centroid}

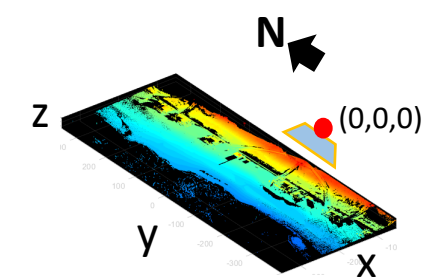
Z_{centroid}



Cluster size

Cluster size

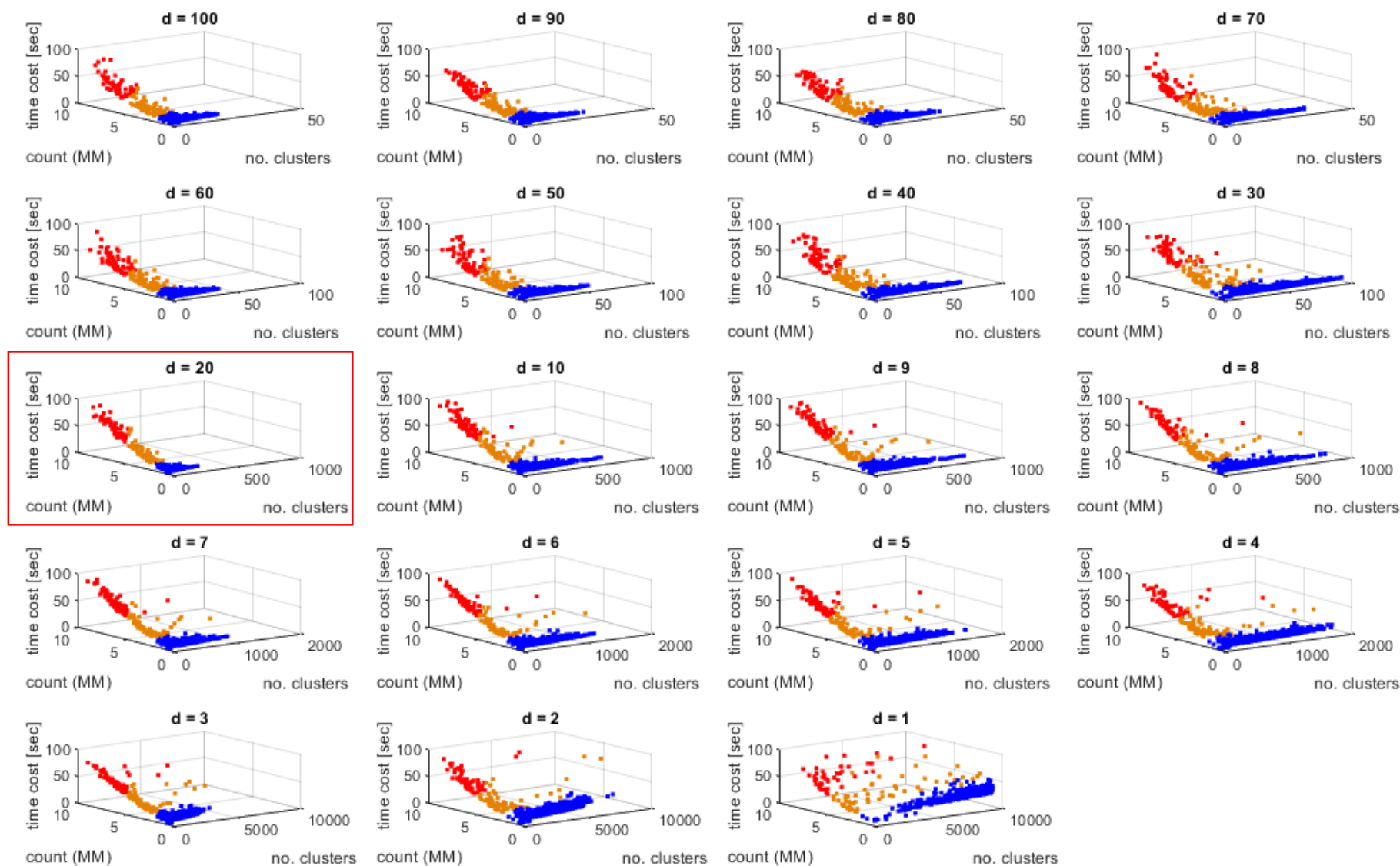
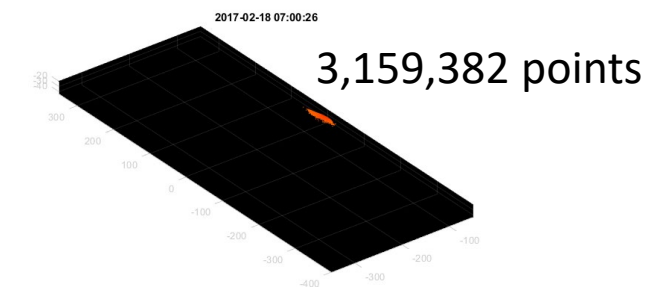
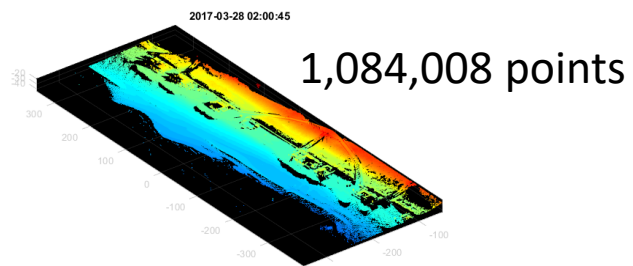
Data filtering/cleaning



Every point is 1 LAZ File

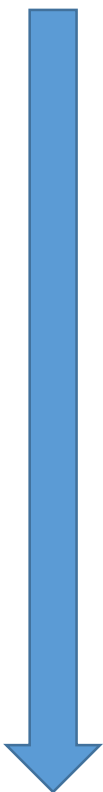
- with little noise 3811 (96.8%)
- with moderate noise 67 (1.7%)
- with much noise 61 (1.5%)

example



4. Change detection algorithms

Cloud to cloud distance (C2CD)

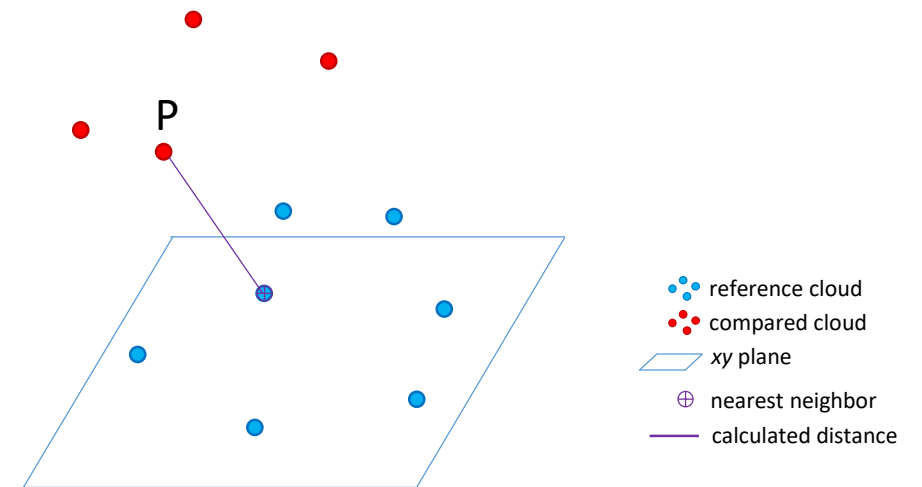
- 
- A large blue arrow pointing downwards, indicating increasing accuracy and time cost.
- **Simple approach**
 - the nearest neighbor's distance
 - **Local modelling**
 - **Point-Model**
 - Least squares plane
 - 2.5D triangulation
 - Quadratic (height function)
 - **Model-Model**
 - Multiscale Model to Model Cloud Comparison (M3C2)

increasing accuracy
and also time cost

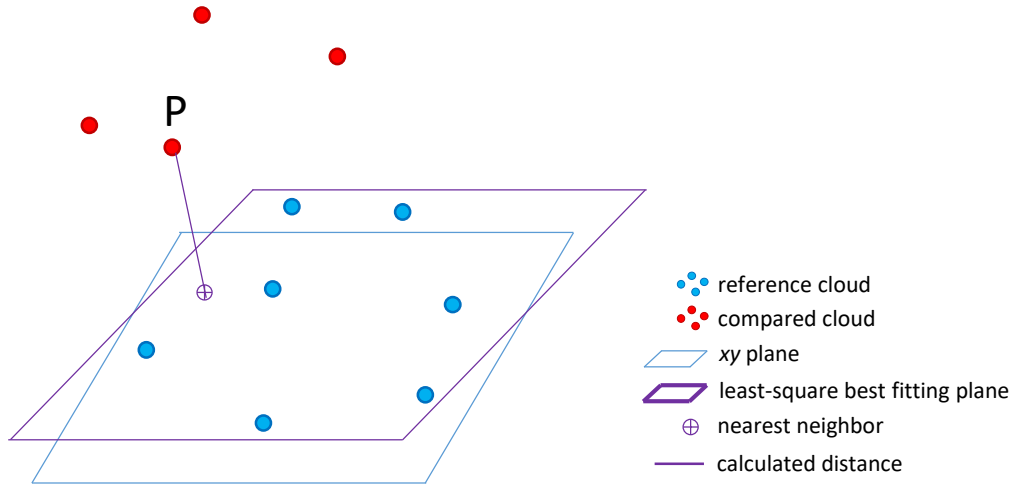
Cloud to cloud distance (C2CD)

- **Simple approach**
 - the nearest neighbor's distance
- Local modelling
 - Point-Model
 - Least squares plane
 - 2D1/2 triangulation
 - Quadratic (height function)
 - Model-Model
 - Multiscale Model to Model Cloud Comparison (M3C2)
 - Dominant movement direction (DMD)

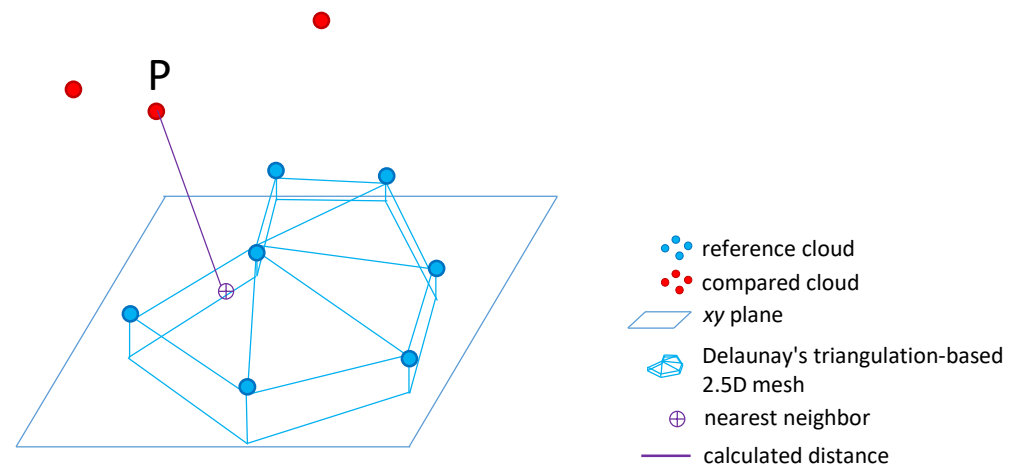
The nearest neighbor



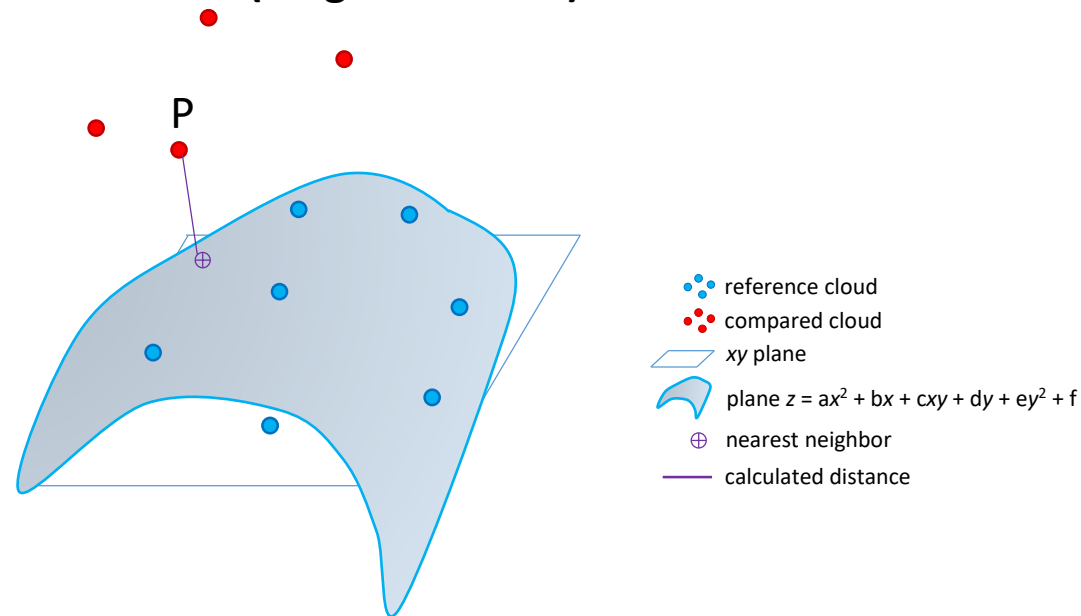
Least squares plane



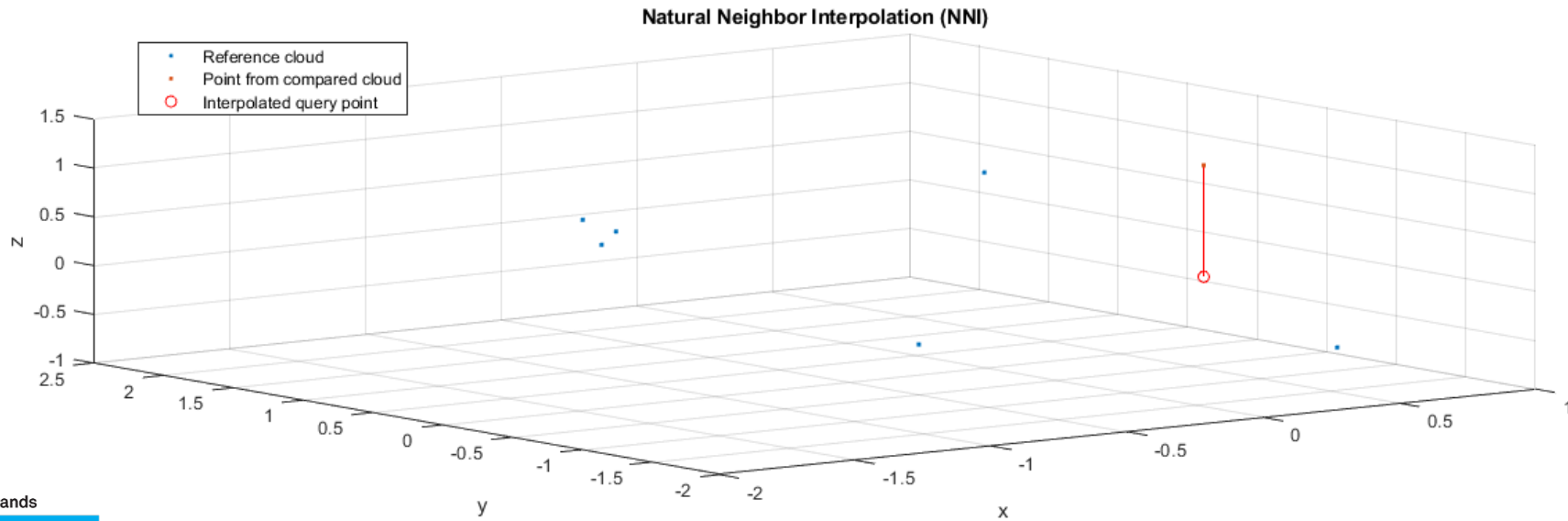
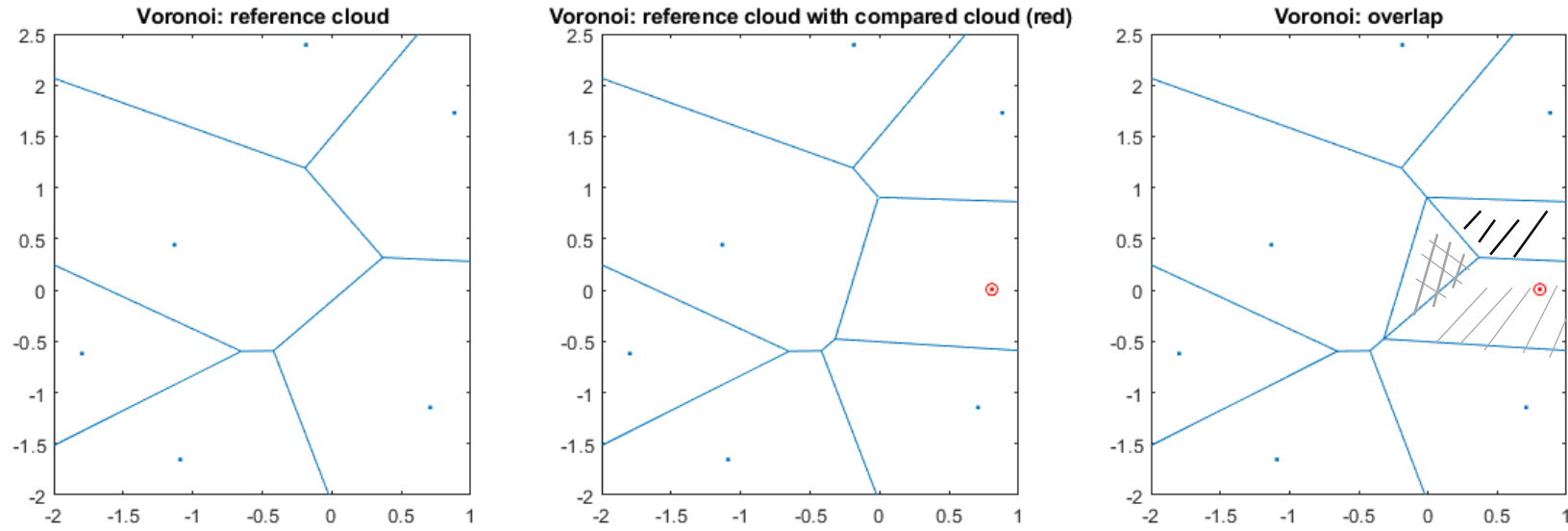
2.5D triangulation



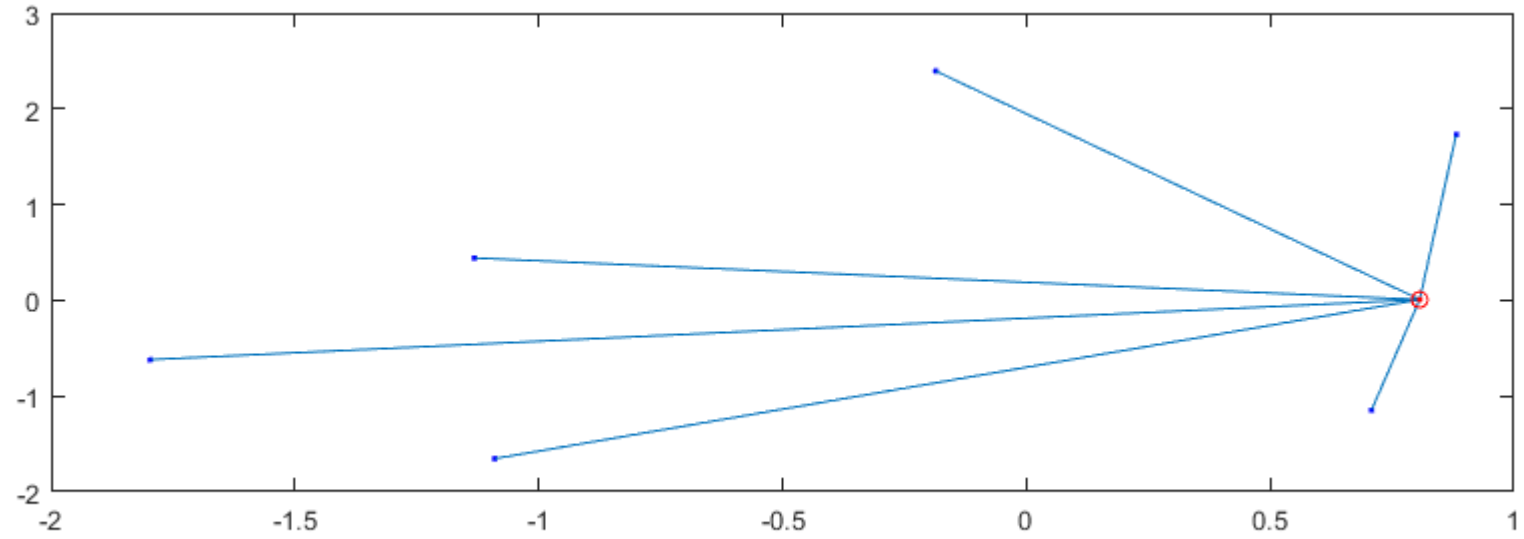
Quadratic (height function)



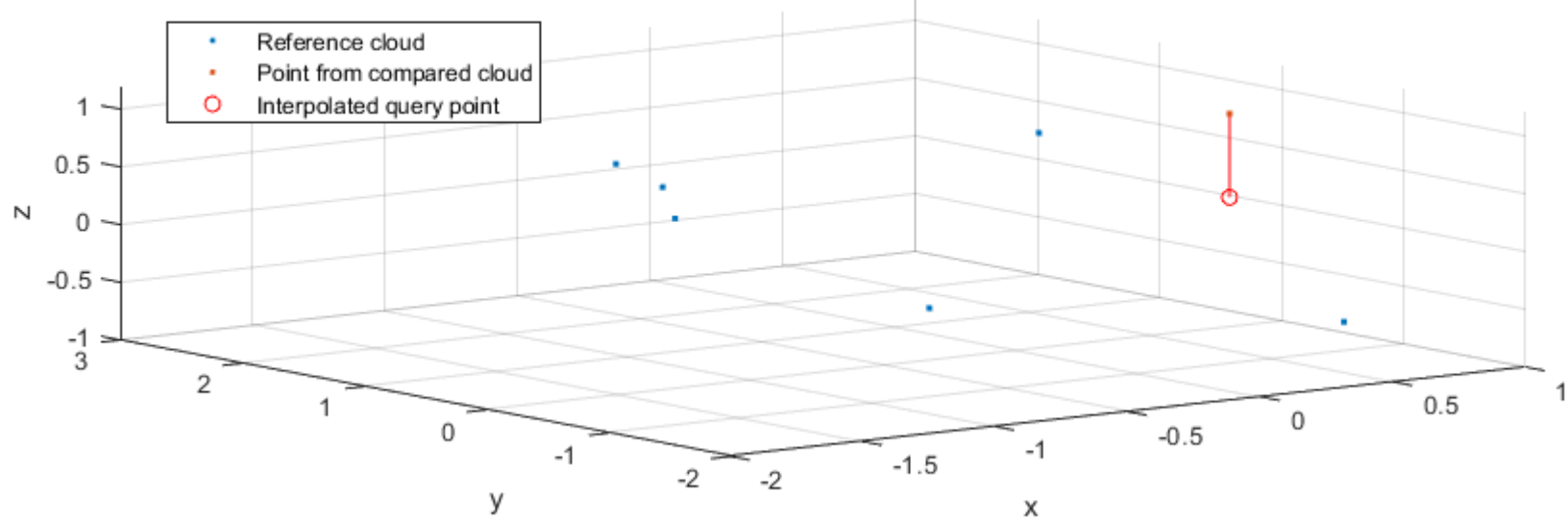
Natural Neighbor Interpolation (NNI)



Inverse Distance Weight (IDW)

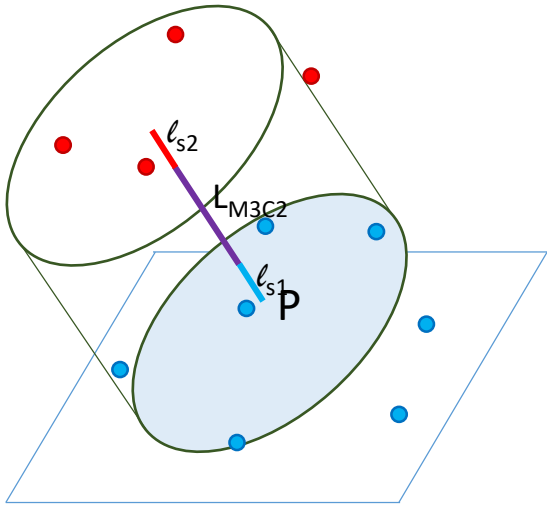


Inverse Distance Weight (IDW)

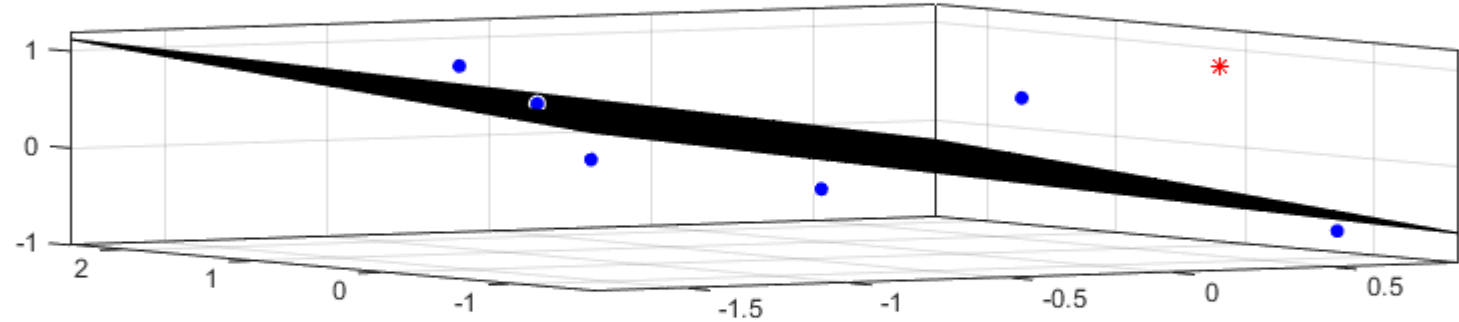
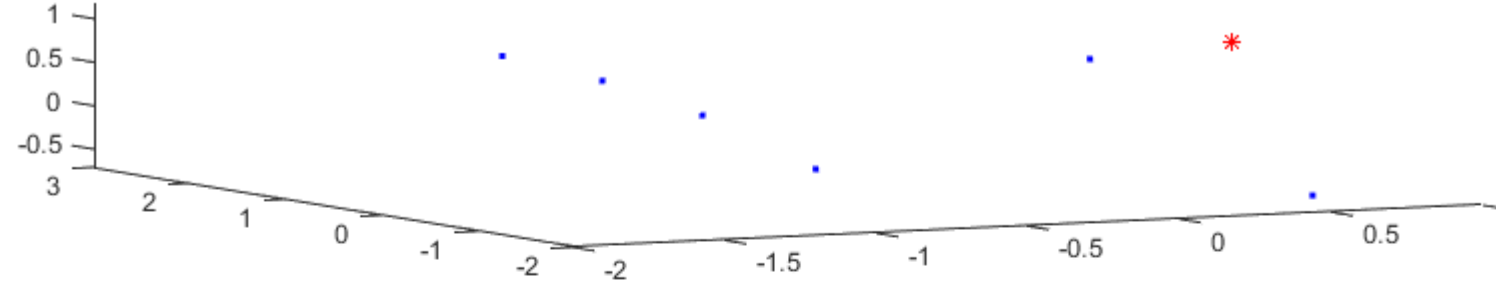


Model-Model

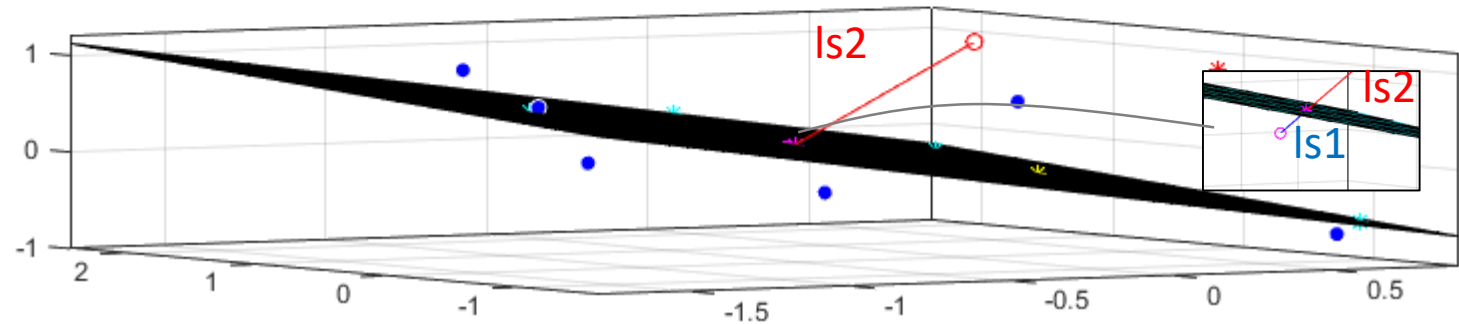
Multiscale Model to Model Cloud Comparison (M3C2)



- reference cloud (s_1)
- compared cloud (s_2)
- xy plane
- cylinder whose base is normal to the neighborhood of P
- calculated distance



$ls_1 = -0.047454$, $ls_2 = 1.1521$, $LM3C2 = ls_2 - ls_1 = 1.1995$, $(LM3C2x, LM3C2y, LM3C2z) = (0.53018, -0.12742, 1.0684)$

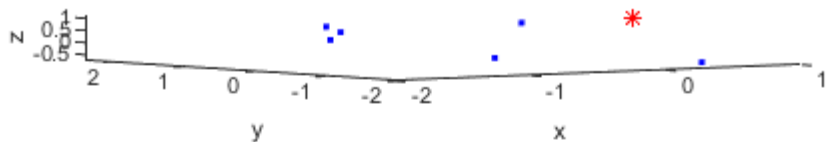


$ss_1 = 0.40777$, $ss_2 = 0$, $LOD95 = \pm 0.32628$

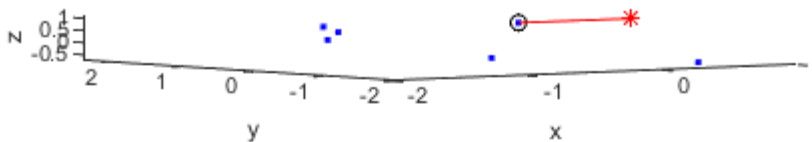
8 options for cloud-to-cloud distance computation



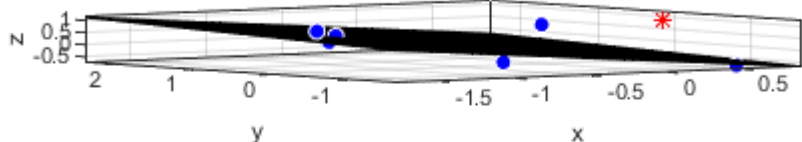
(1) The nearest neighbor



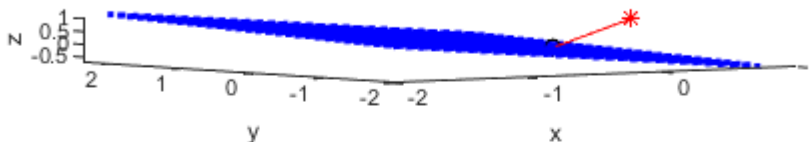
$d = 1.8058, (dx,dy,dz) = (-0.074838, -1.7234, 0.53426), P = (0.88158, 1.7305, 0.30543)$



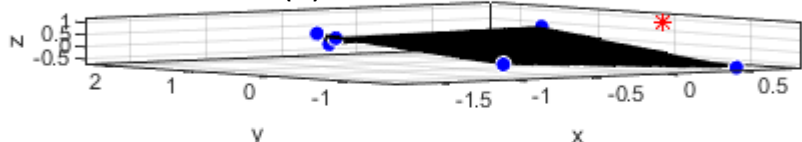
(2) $z = p00 + p10*x + p01*y$



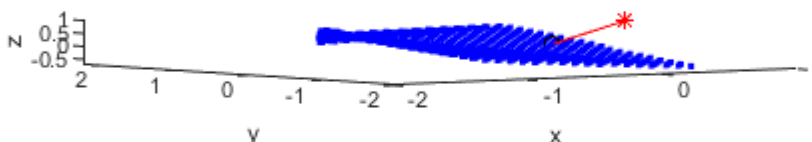
$d = 1.1521, (dx,dy,dz) = (0.50921, -0.12238, 1.0262), P = (0.29754, 0.12949, -0.18647)$



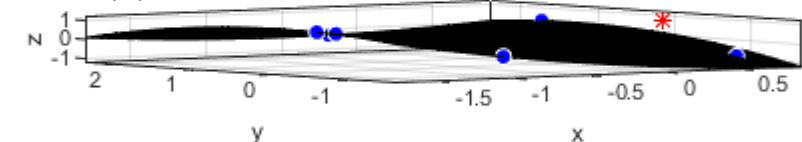
(3) Linear interpolation



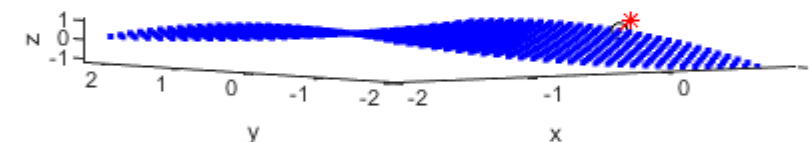
$d = 1.0128, (dx,dy,dz) = (0.38689, -0.29294, 0.88896), P = (0.41986, 0.30006, -0.049269)$



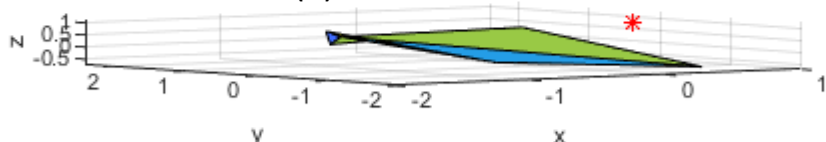
(4) $z = p00 + p10*x + p01*y + p20*x^2 + p11*x*y + p02*y^2$



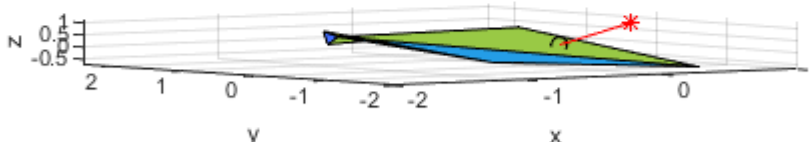
$d = 0.74553, (dx,dy,dz) = (-0.074838, -0.29294, 0.68147), P = (0.88158, 0.30006, 0.15822)$



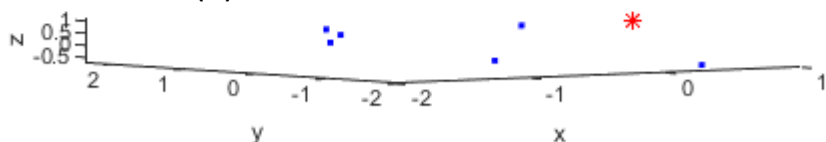
(5) 2.5D triangulation



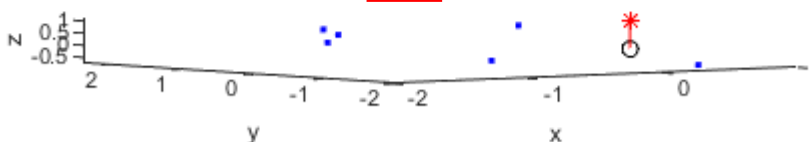
$d = 1.0113, (dx,dy,dz) = (0.35095, -0.33347, 0.88795), P = (0.45579, 0.34058, -0.048256)$



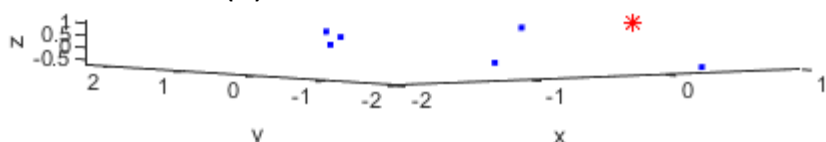
(6) Natural Neighbor Interpolation (NNI)



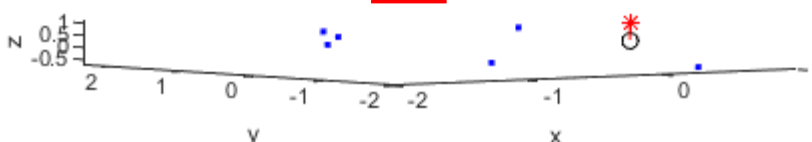
$d = 1.1442, (dx,dy,dz) = (0,0,1.1442), P = (0.80675, 0.0071106, -0.30454)$



(7) Inverse Distance Weight (IDW)

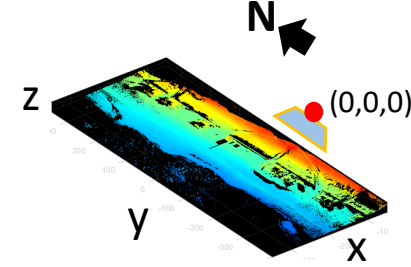


$d = 0.72417, (dx,dy,dz) = (0,0,0.72417), P = (0.80675, 0.0071106, 0.11552)$



Methodology for the implementation

- t1 -Retrieve the reference/compared cloud sample
- t2 -Model the reference-cloud local surface
- t3 -Find the nearest point in the local surface to the compared point(s)
- t4 -Calculate the minimum distance
- t5 -Update the distance as an attribute in the compared cloud sample

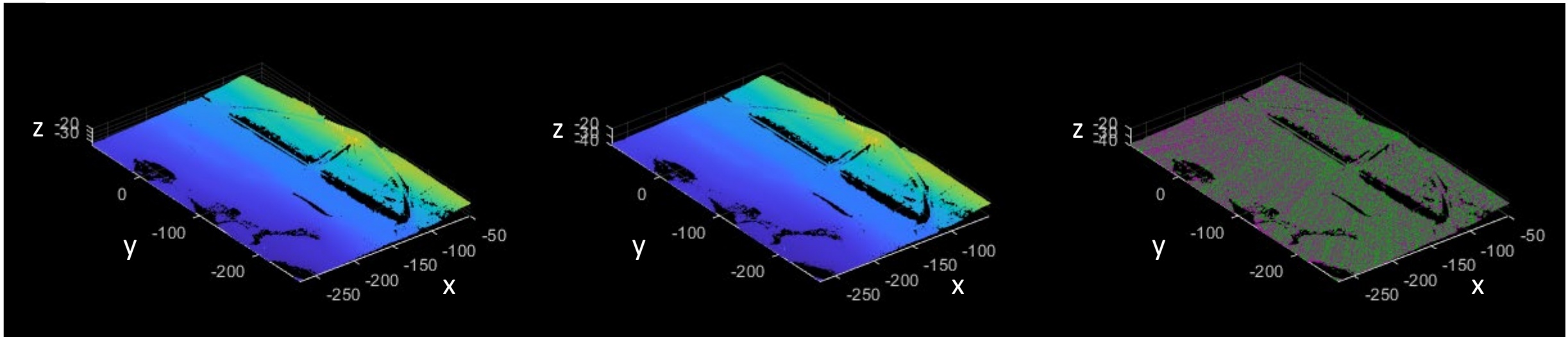


5. Preliminary results

Reference cloud

Compared cloud

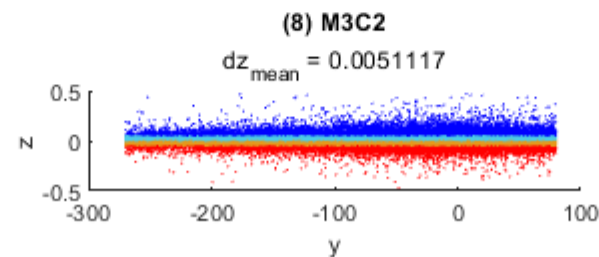
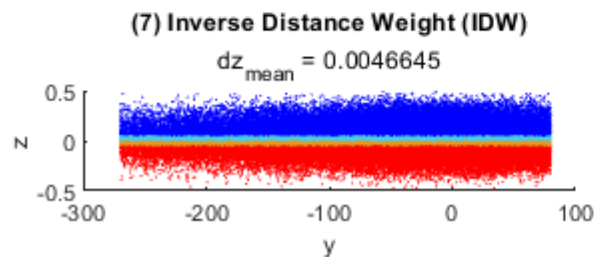
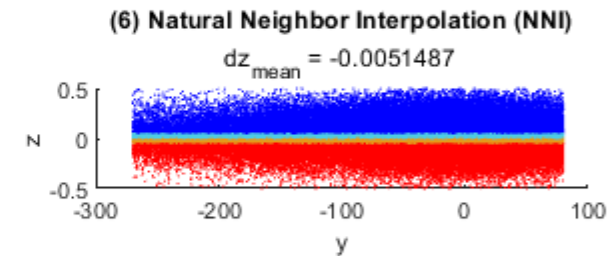
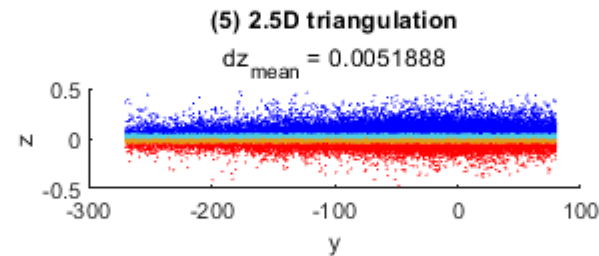
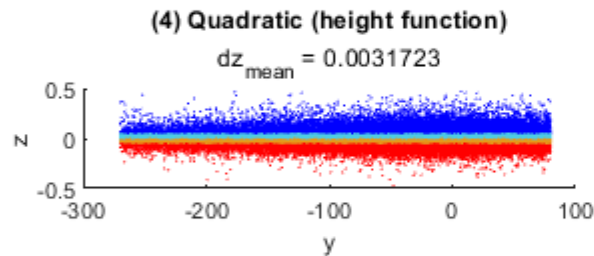
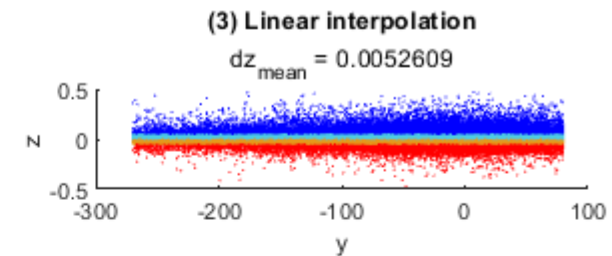
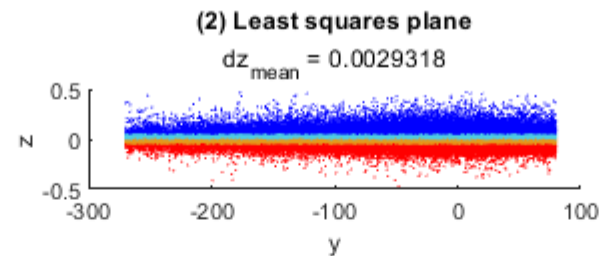
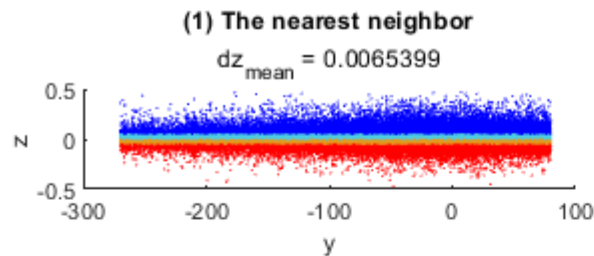
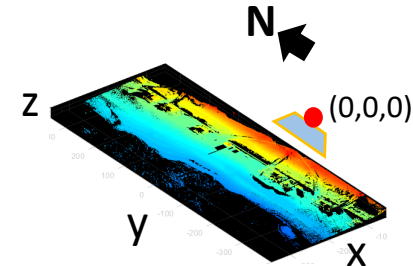
Comparison



- reference cloud
- compared cloud

5. Preliminary results

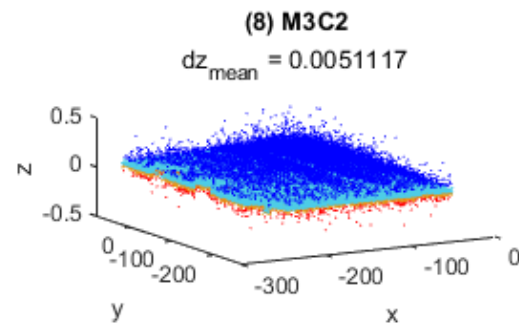
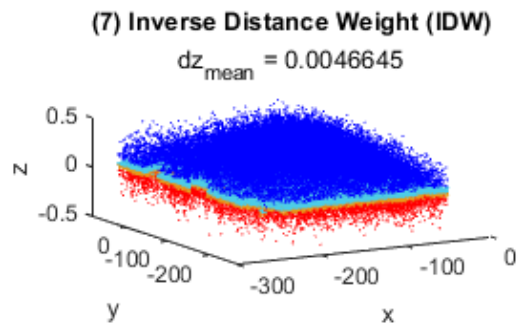
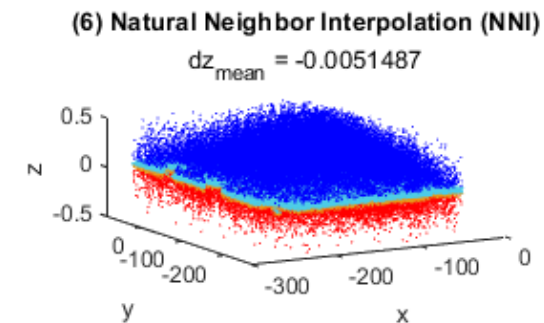
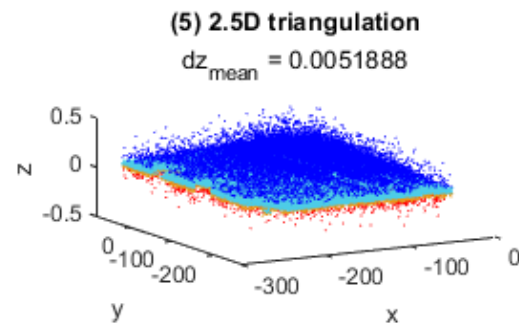
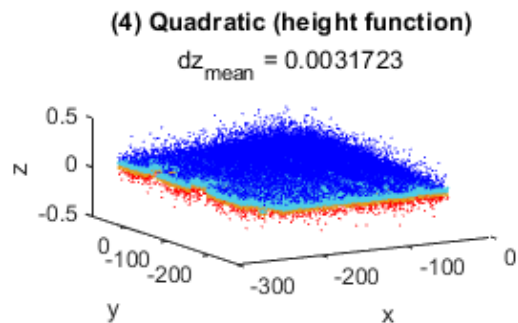
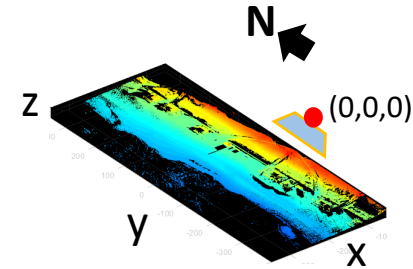
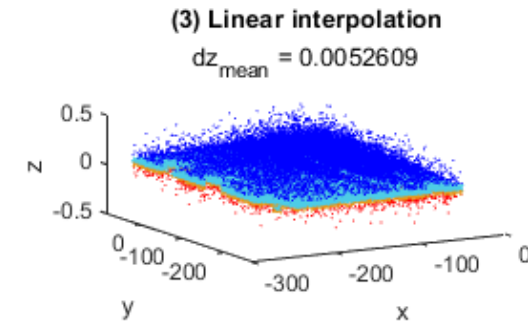
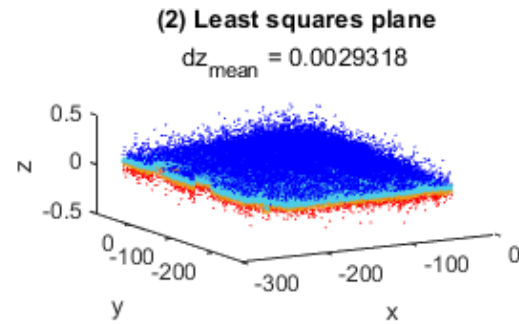
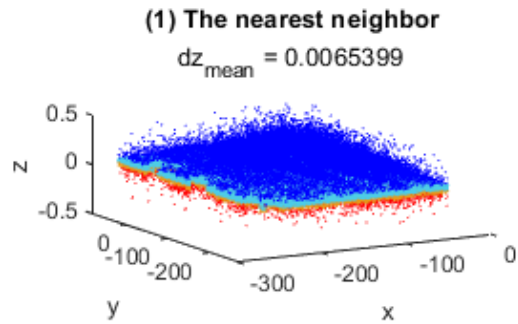
Vertical distance (dz)



- $dz > 0$ "deposition"
- $dz < 0$ "erosion"

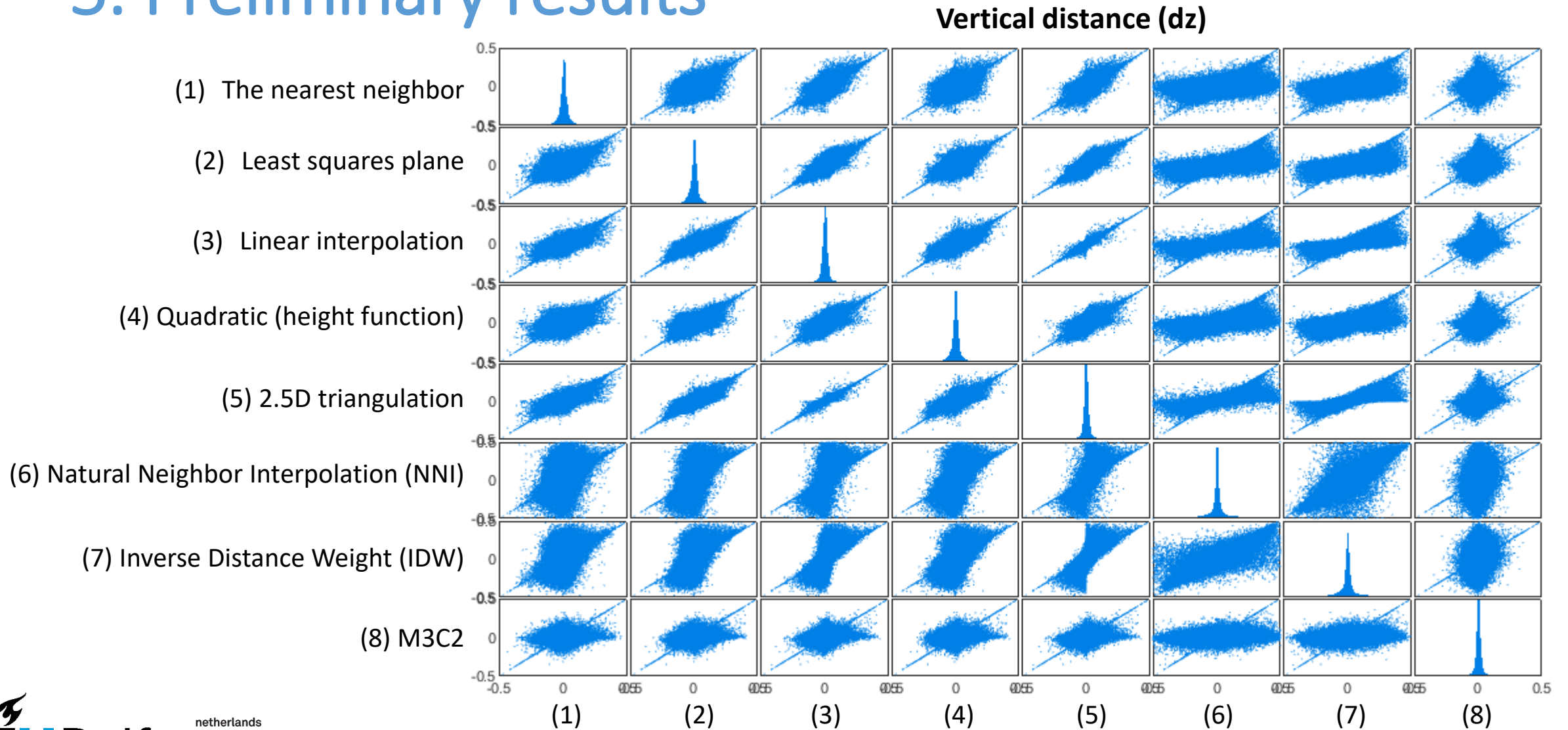
5. Preliminary results

Vertical distance (dz)

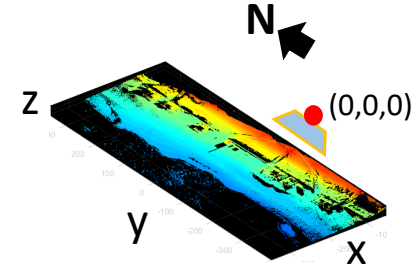
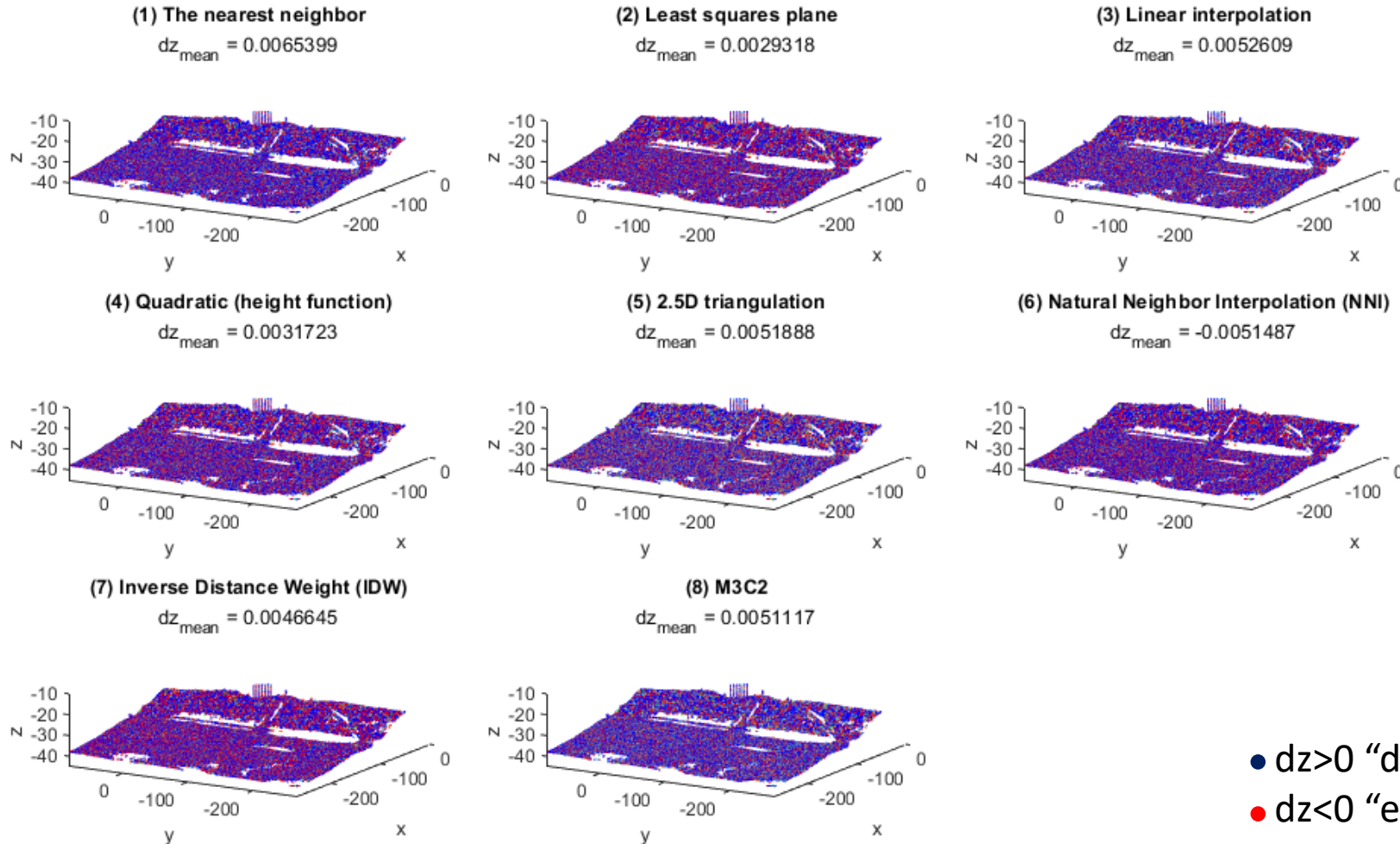


- $dz > 0$ "deposition"
- $dz < 0$ "erosion"

5. Preliminary results



5. Preliminary results



- $dz > 0$ “deposition”
- $dz < 0$ “erosion”

6. Conclusions and future steps

1. CoastScan data was explored
2. Data filtering/cleaning approach tested
3. Explored cloud-to-cloud distance (C2CD) methods
 - Complete analysis of C2CD methods in the entire database (including the higher-resolution database)
 - Explore the database approach to performing queries, calculation
 - Test the C2CD methods with the AHN database



Data cleaning CoastScan data
& change detection in
spatio-temporal point clouds

v.diazmercado@tudelft.nl



Thanks!

