# Quality analysis of point cloud change detection algorithms
# &
# SFC-based ultra-fast change detection

**Vitali Diaz**, Peter van Oosterom, Martijn Meijers,

Edward Verbree, Nauman Ahmed and Thijs van Lankveld

TU Delft

netherlands eScience center

nD-PointCloud

http://nd-pc.org/

October 30, 2023

# Outline

1. Introduction
2. Methodology
3. Results
4. Conclusions
5. Future work

# 1 Introduction

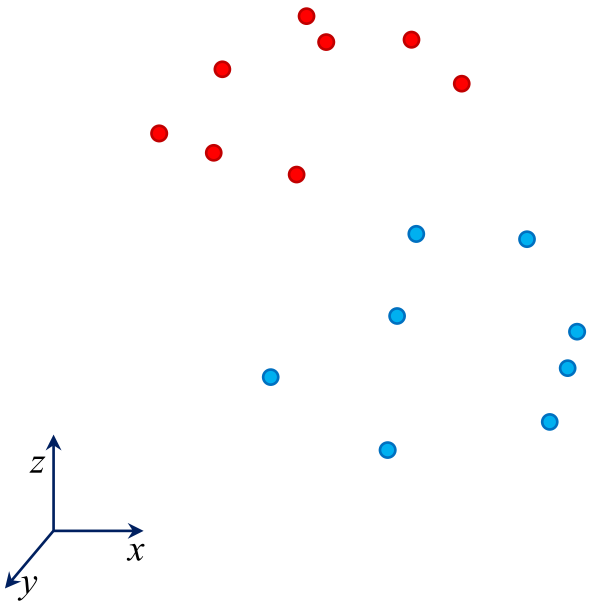
- In **point cloud change detection,** one of the initial stages is the performance of **cloud-to-cloud (C2C) distance calculation**.
- There are various methods for calculating the C2C distance between two corresponding point clouds.
- These methods can be classified from **simple** to **complex**, with more steps and calculations required for the latter.
- Generally, a more complex method is assumed to result in a more precise distance calculation, but this assumption is not always evaluated.
- **We assess the performance of eight commonly used methods for calculating the C2C distance with a controlled displacement test.**

# 1 Introduction

## Some definitions

reference cloud
time $t$

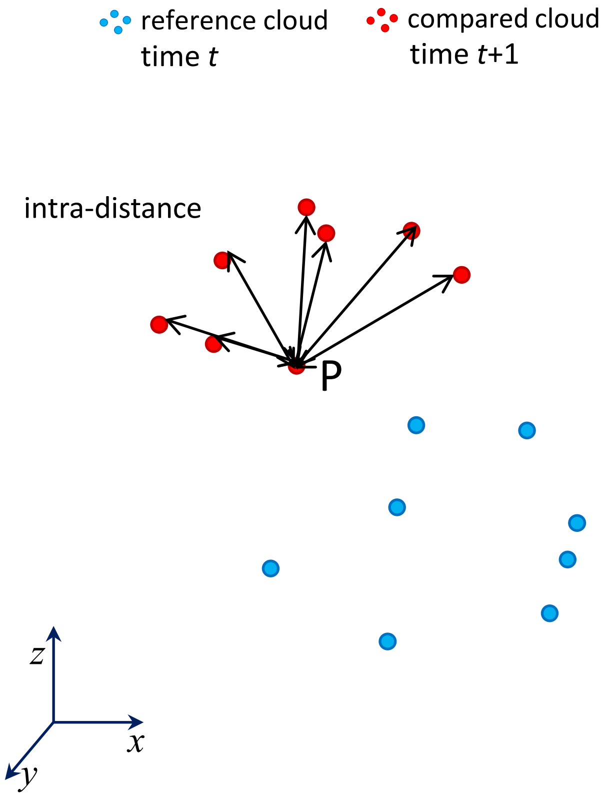
compared cloud
time $t$+1




netherlands eScience center

# 1 Introduction

## Some definitions



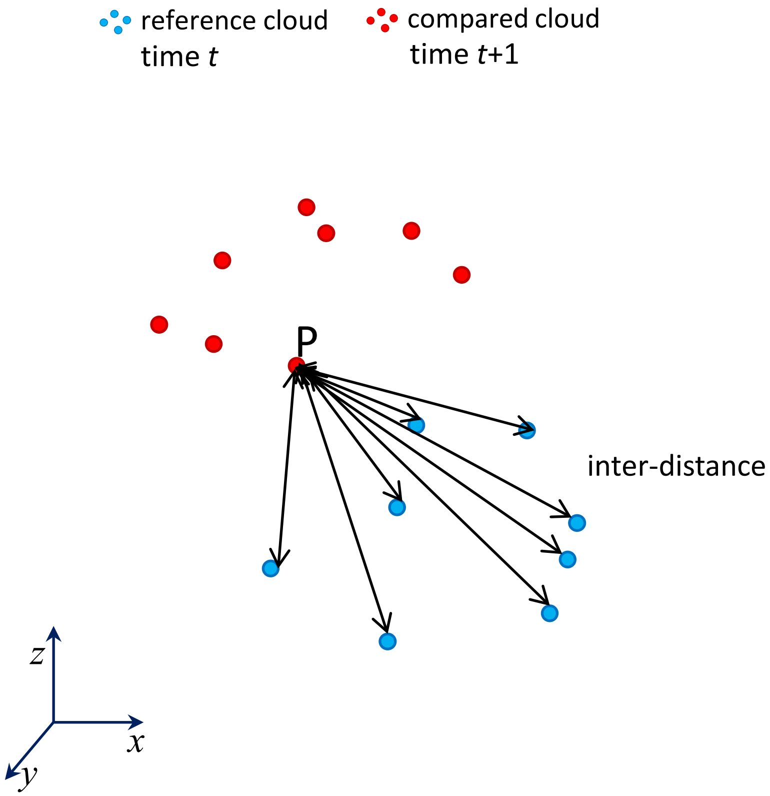

**Intra-distance:** distance between points within the same point cloud.

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

Intra-distance is calculated for each point.

# 1 Introduction

## Some definitions




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

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

Intra-distance is calculated for each point.

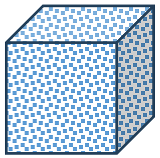
**Inter-distance:** distance between two corresponding point clouds taken at different epochs.

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 define/calculate it.

# 2 Methodology

**Controlled displacement test:**



(1) A specific point cloud is designated as the "reference cloud"

(2) The intra-distance is calculated for every point within the reference cloud and the average intra-distance.

(3) To explore a range of scenarios, artificial displacements are proposed based on the average intra-distance.

(4) These proposed displacements are applied to all points within the reference cloud, creating a "compared cloud" for each displacement scenario.

(5) The calculation of inter-point distance between the compared and the reference cloud takes place. Eight different methods were tested.

(6) Finally, each method is evaluated to determine its accuracy in capturing the applied artificial displacement.

# 2 Methodology

**Controlled displacement test:**

(1) A specific point cloud is designated as the "reference cloud"

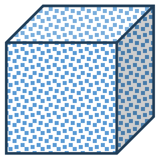
(2) The intra-distance is calculated for every point within the reference cloud and the average intra-distance.

(3) To explore a range of scenarios, artificial displacements are proposed based on the average intra-distance.

(4) These proposed displacements are applied to all points within the reference cloud, creating a "compared cloud" for each displacement scenario.

(5) The calculation of inter-point distance between the compared and the reference cloud takes place. Eight different methods were tested.

(6) Finally, each method is evaluated to determine its accuracy in capturing the applied artificial displacement.



intra-distance

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

nD-PointCloud

TU Delft netherlands eScience center

# 2 Methodology

**Controlled displacement test:**

(1) A specific point cloud is designated as the "reference cloud"

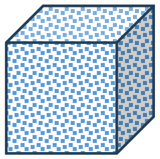
(2) The intra-distance is calculated for every point within the reference cloud and the average intra-distance.

(3) To explore a range of scenarios, artificial displacements are proposed based on the average intra-distance.

(4) These proposed displacements are applied to all points within the reference cloud, creating a "compared cloud" for each displacement scenario.

(5) The calculation of inter-point distance between the compared and the reference cloud takes place. Eight different methods were tested.

(6) Finally, each method is evaluated to determine its accuracy in capturing the applied artificial displacement.



intra-distance

intra-distance$_{avg}$

dx,dy,dz

nD-PointCloud

TU Delft

netherlands eScience center

# 2 Methodology

**Controlled displacement test:**

(1) A specific point cloud is designated as the "reference cloud"

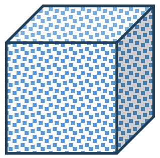
(2) The intra-distance is calculated for every point within the reference cloud and the average intra-distance.

(3) To explore a range of scenarios, artificial displacements are proposed based on the average intra-distance.

(4) These proposed displacements are applied to all points within the reference cloud, creating a "compared cloud" for each displacement scenario.

(5) The calculation of inter-point distance between the compared and the reference cloud takes place. Eight different methods were tested.

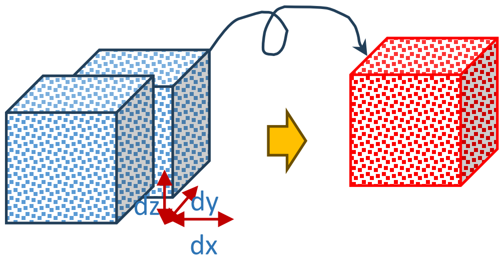
(6) Finally, each method is evaluated to determine its accuracy in capturing the applied artificial displacement.



intra-distance

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

dx,dy,dz



# 2 Methodology

**Controlled displacement test:**

(1) A specific point cloud is designated as the "reference cloud"

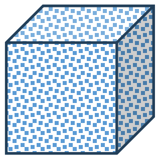
(2) The intra-distance is calculated for every point within the reference cloud and the average intra-distance.

(3) To explore a range of scenarios, artificial displacements are proposed based on the average intra-distance.

(4) These proposed displacements are applied to all points within the reference cloud, creating a "compared cloud" for each displacement scenario.

(5) The calculation of inter-point distance between the compared and the reference cloud takes place. Eight different methods were tested.

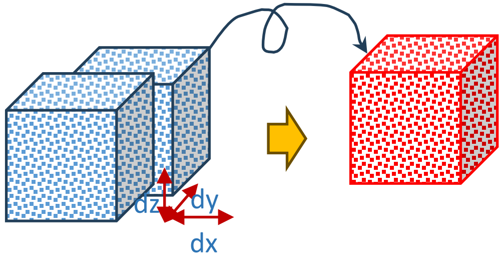
(6) Finally, each method is evaluated to determine its accuracy in capturing the applied artificial displacement.

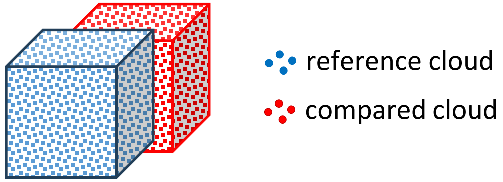


intra-distance

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

dx,dy,dz



# 2 Methodology

**Controlled displacement test:**

(1) A specific point cloud is designated as the "reference cloud"

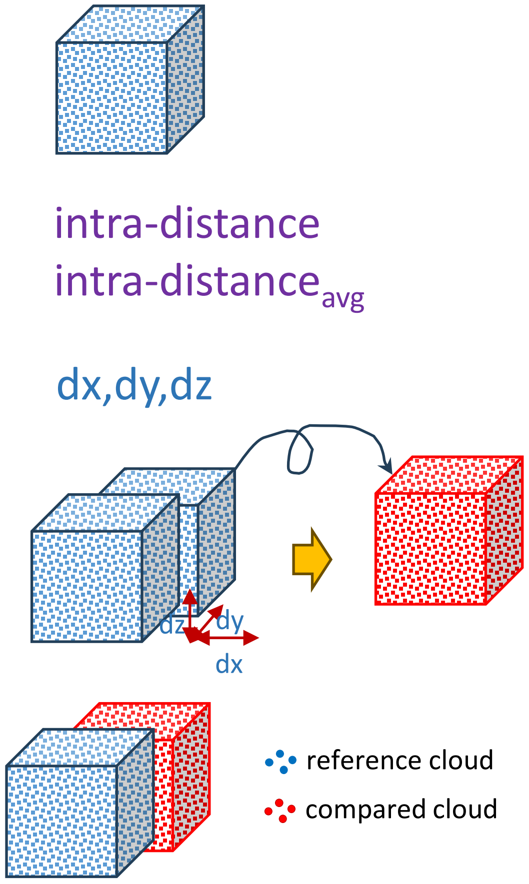
(2) The intra-distance is calculated for every point within the reference cloud and the average intra-distance.

(3) To explore a range of scenarios, artificial displacements are proposed based on the average intra-distance.

(4) These proposed displacements are applied to all points within the reference cloud, creating a "compared cloud" for each displacement scenario.

(5) The calculation of inter-point distance between the compared and the reference cloud takes place. Eight different methods were tested.

(6) Finally, each method is evaluated to determine its accuracy in capturing the applied artificial displacement.



# 2 Methodology

## Inter-distance (C2C distance)

### Simple approach

- The nearest neighbor

### Weighted methods

- Natural Neighbor Interpolation (NNI)
- Inverse Distance Weight (IDW)

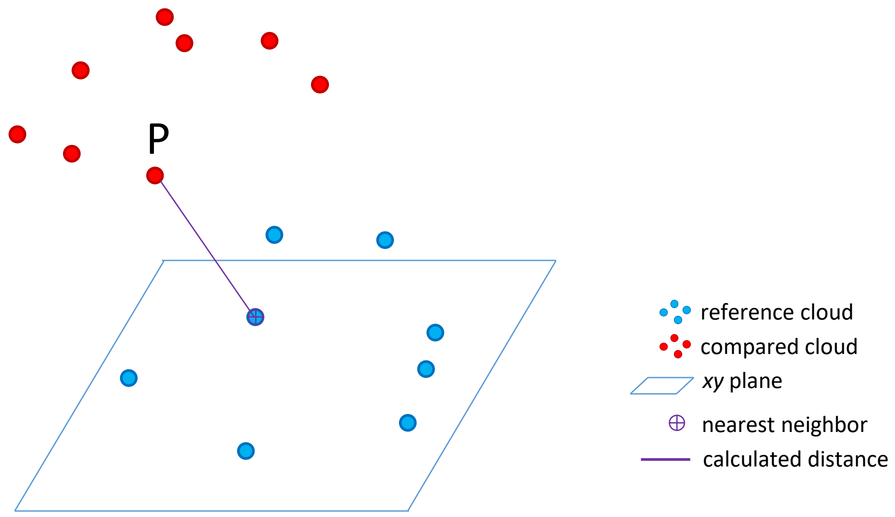
### Local modelling

- **Point-Model**
  - Least squares plane
  - Linear interpolation
  - 2.5D triangulation
  - Quadratic (height function)
- **Model-Model**
  - Multiscale Model to Model Cloud Comparison (M3C2)

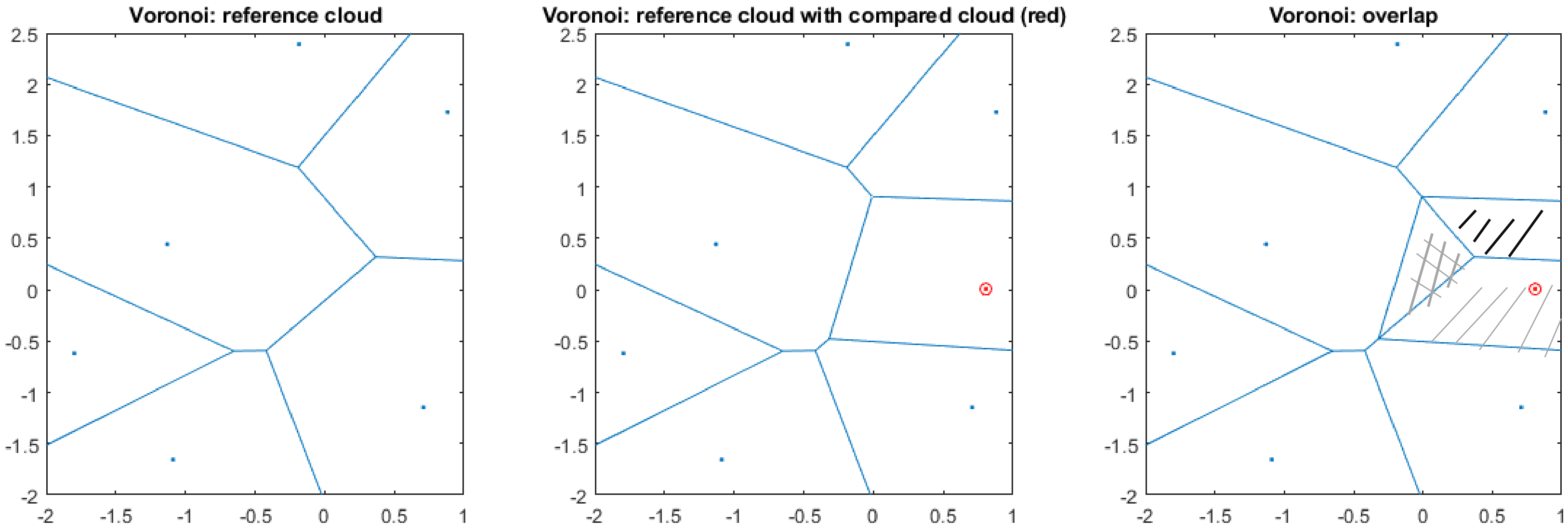
increasing accuracy
and also time cost

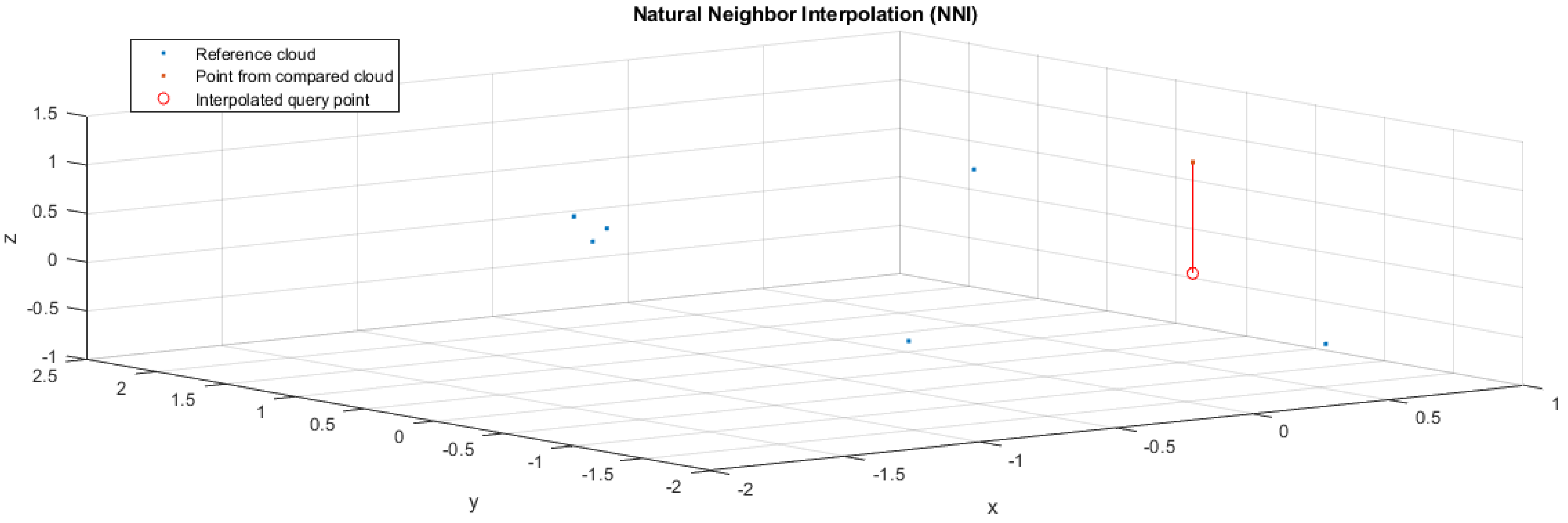
# 2 Methodology

**The nearest neighbor**

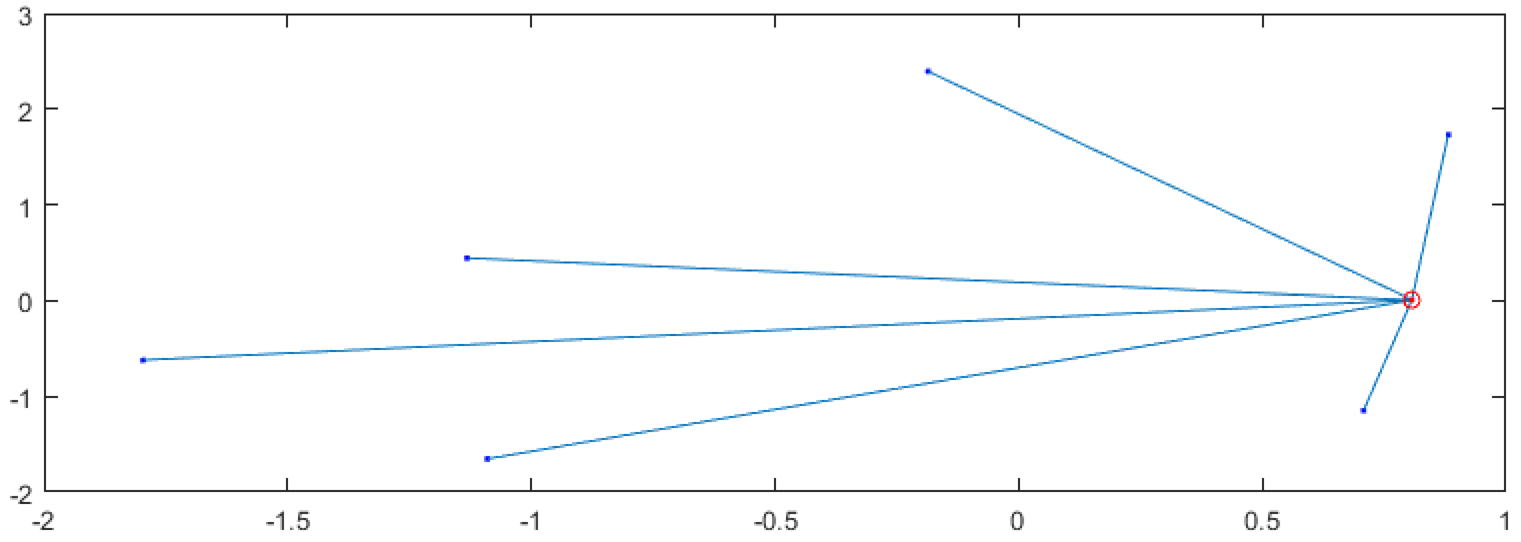


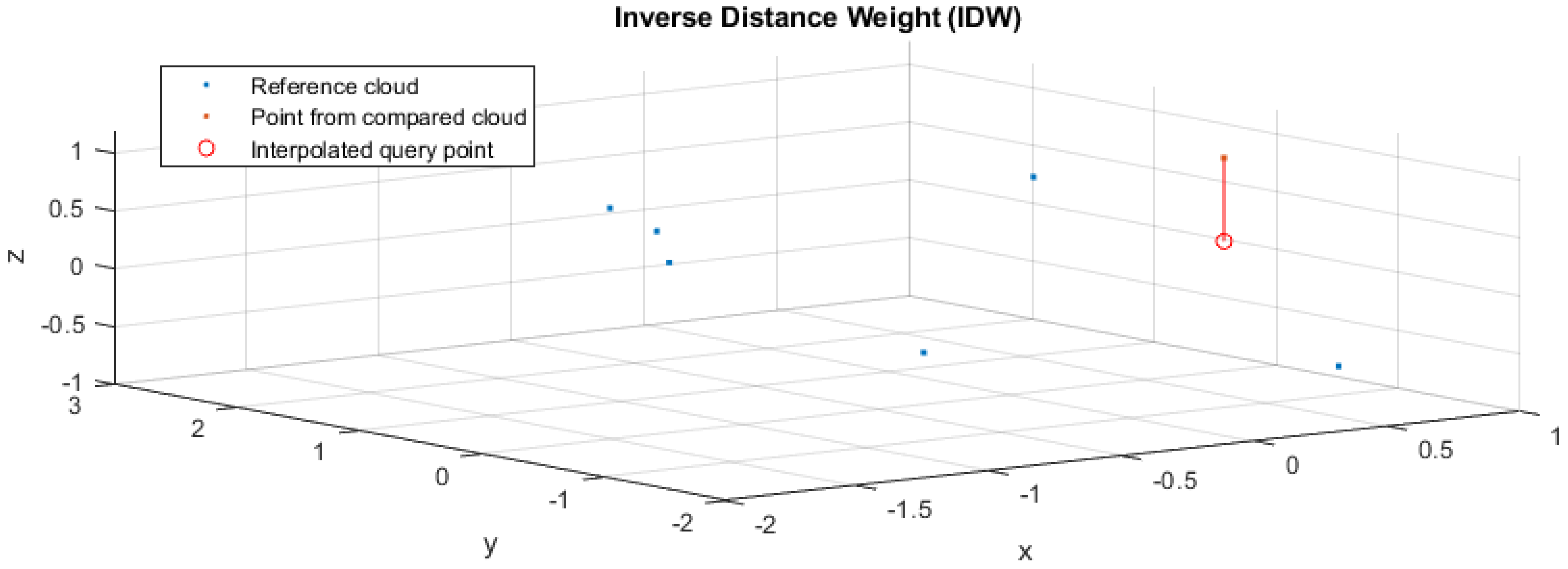
# Natural Neighbor Interpolation (NNI)



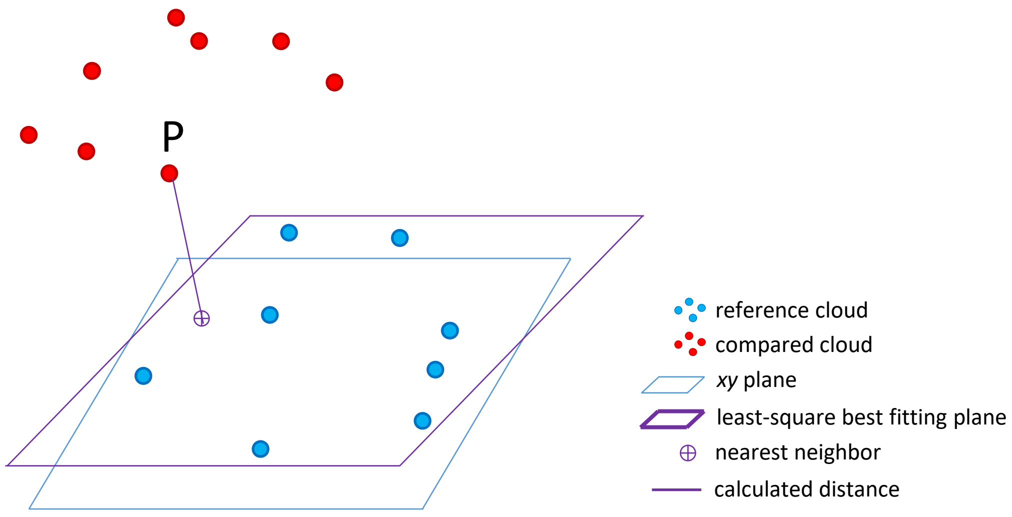
# Inverse Distance Weight (IDW)



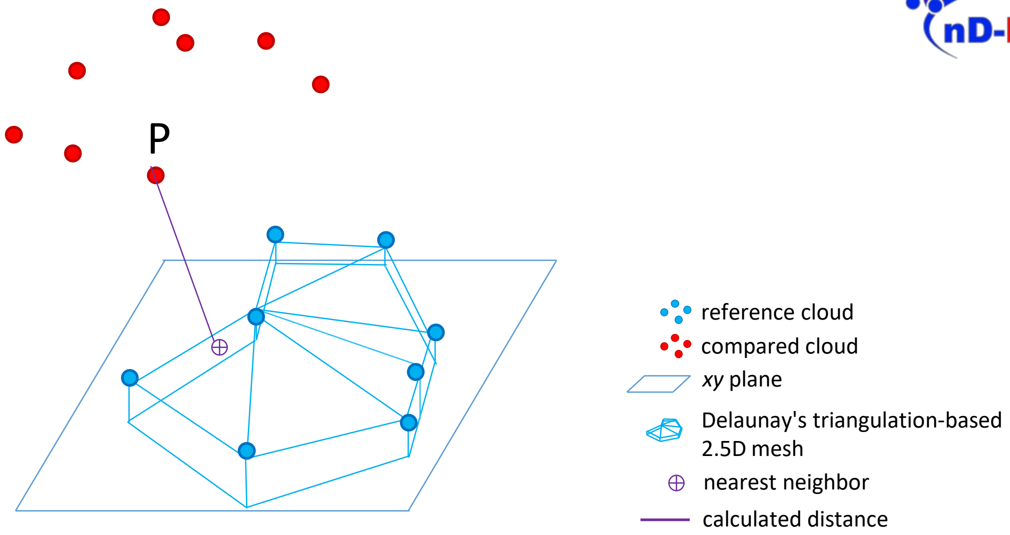



# Local modelling

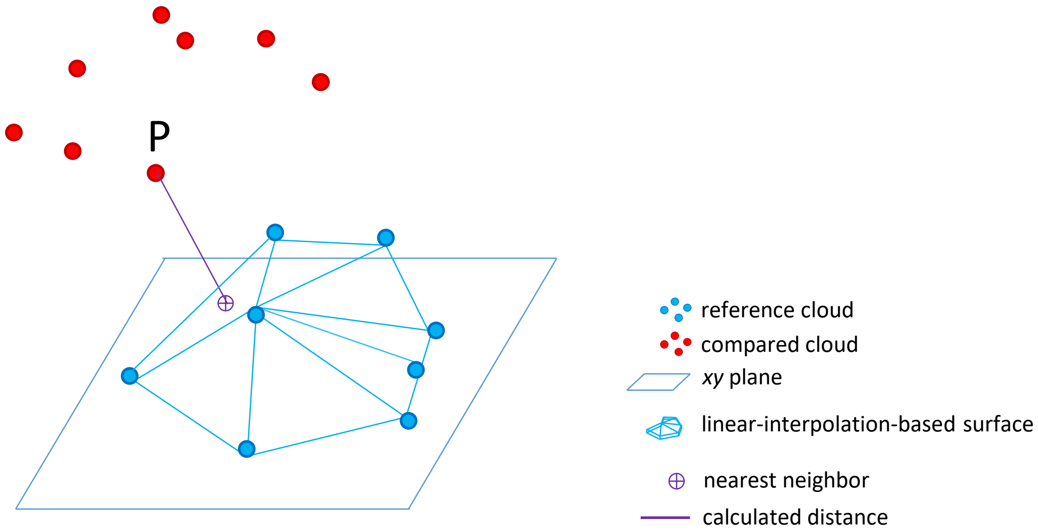
## Least squares plane



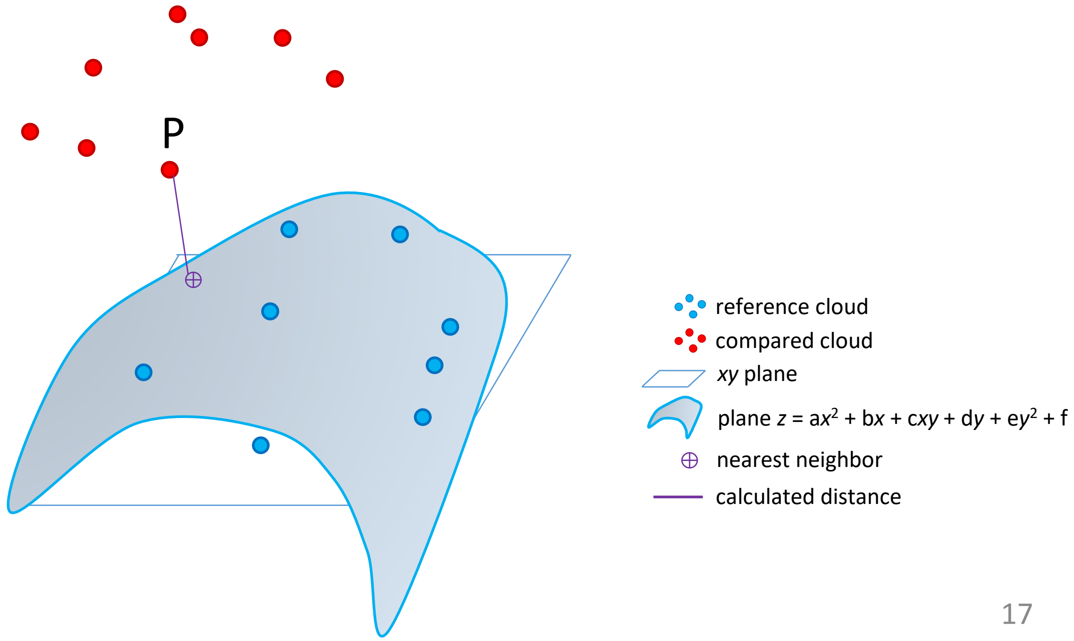

## 2.5D triangulation




## Linear interpolation

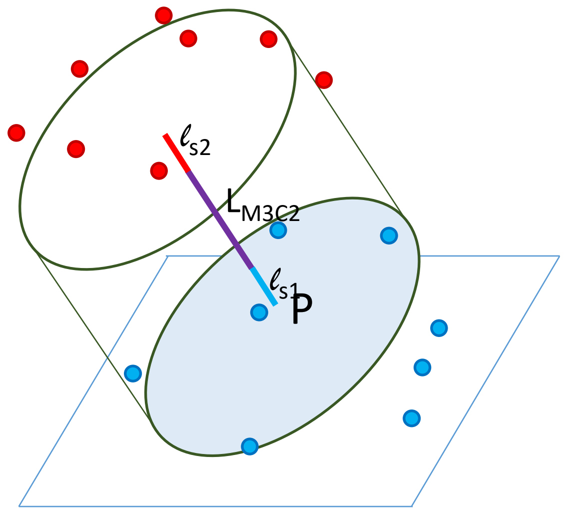



## Quadratic (height function)

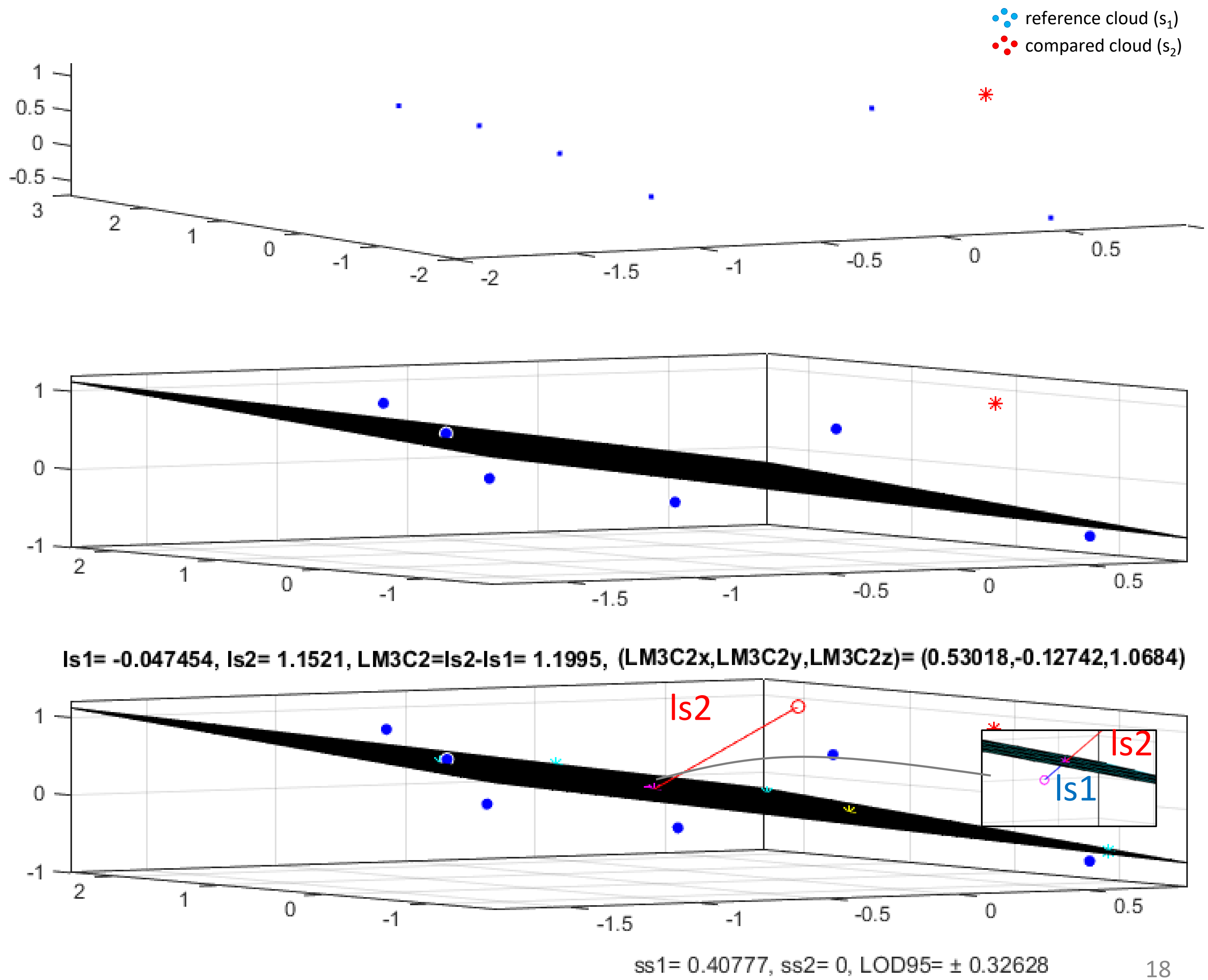


# Model-Model

## 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



# 2 Methodology

## 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** | |
| • Multiscale Model to Model Cloud Comparison (M3C2) | dx,dy,dz |

increasing accuracy
and also time cost

# 2 Methodology

Quality analysis of point cloud change detection algorithms

$$deviation = \frac{|applied\ displacement - calculated\ displacement|}{applied\ displacement}$$

Six intervals were considered for assessing the deviation:

- 0 to 10,
- 10 to 20,
- 20 to 30,
- 30 to 40,
- 40 to 50,
- and greater than 50%

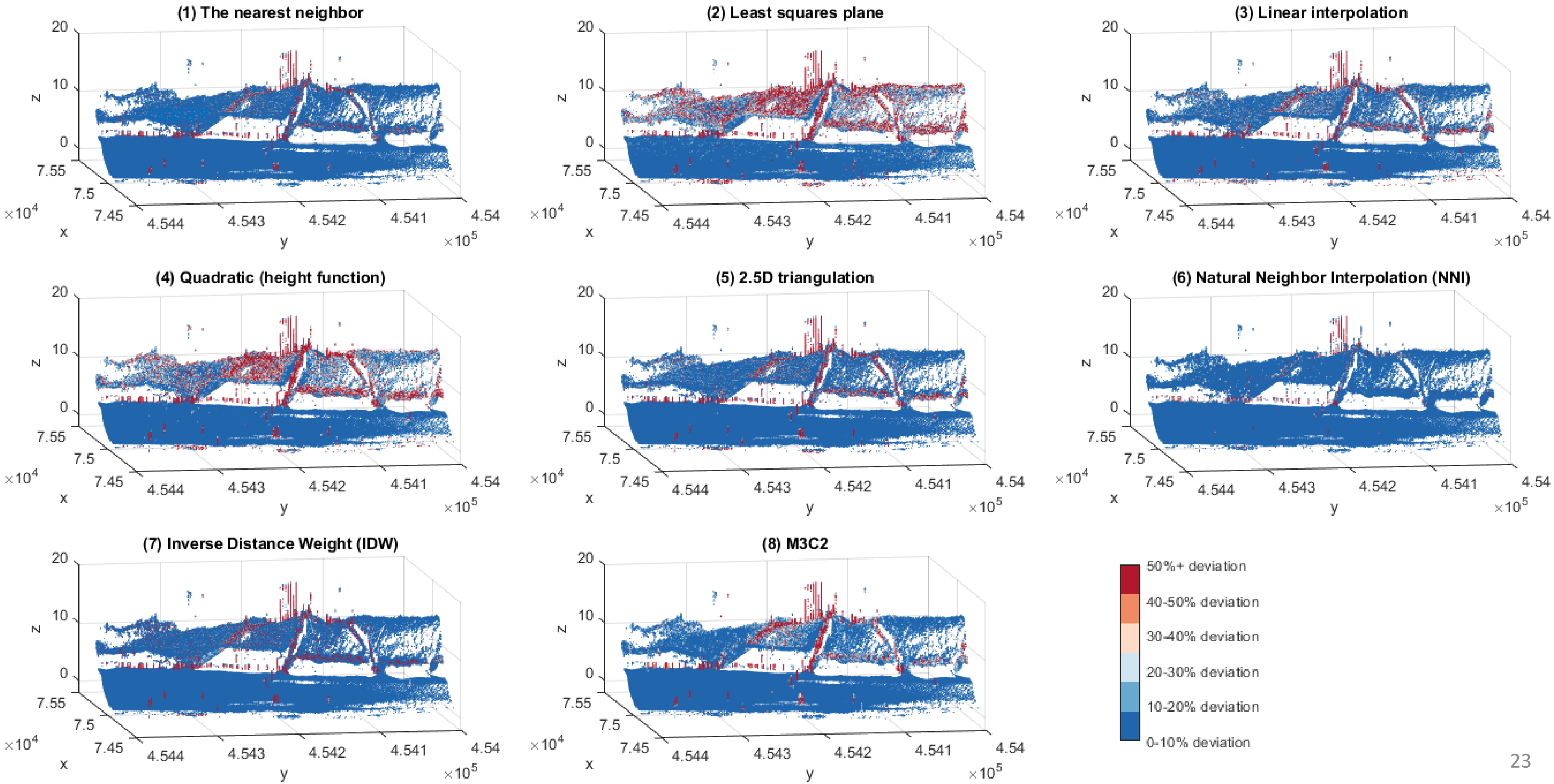
# 3 Results

a. Testing 4 different datasets (bunny, lake, CostScan and AHN)
b. 3 different artificial displacements: vertical, horizontal, and diagonal
c. 8 different methods

# CoastScan

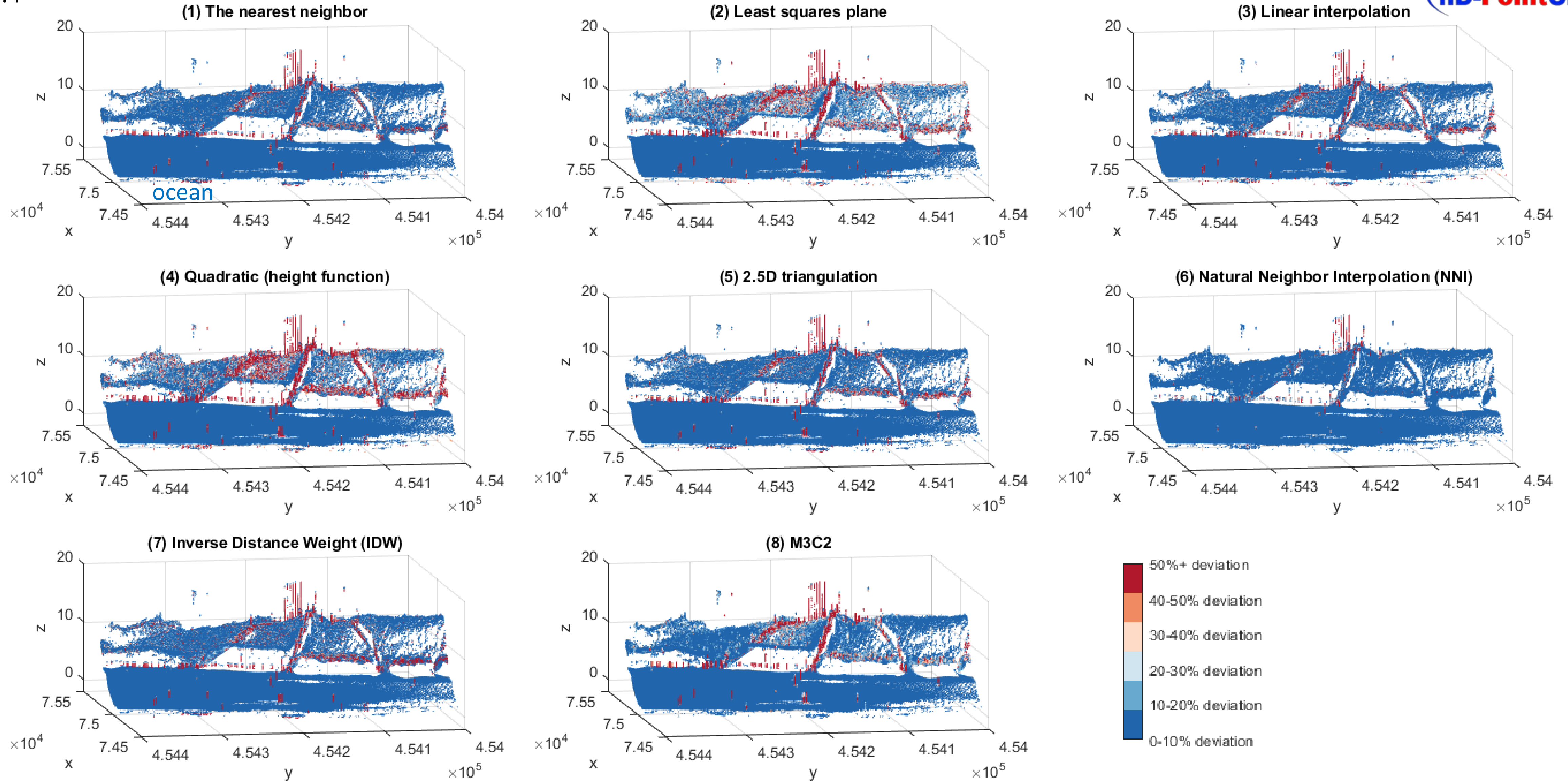
dx=0
dy=0
dz=0.2

Average intra-distance = 0.396 m



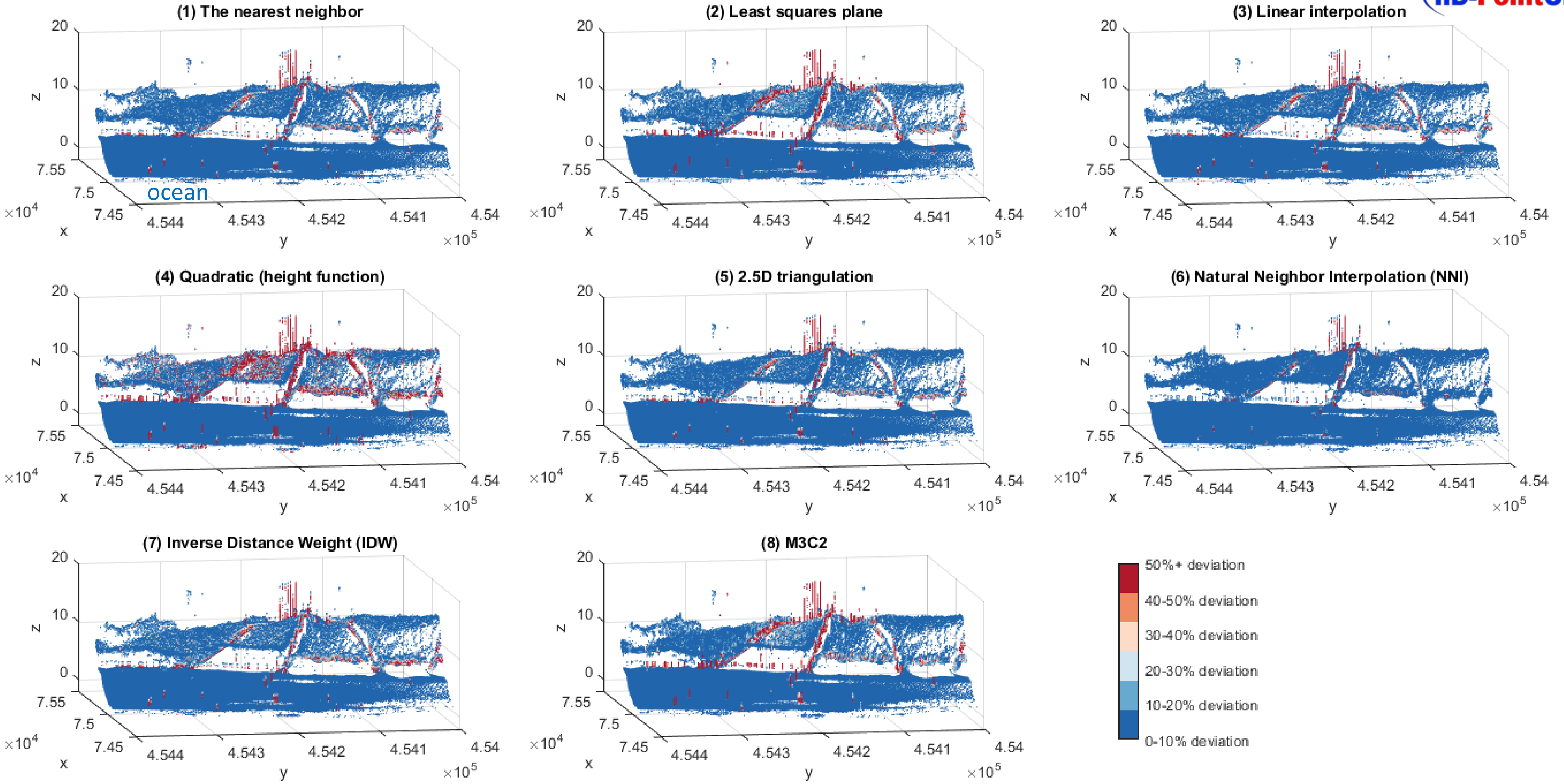
dx=0
dy=0
dz=0.4

Average intra-distance = 0.396 m



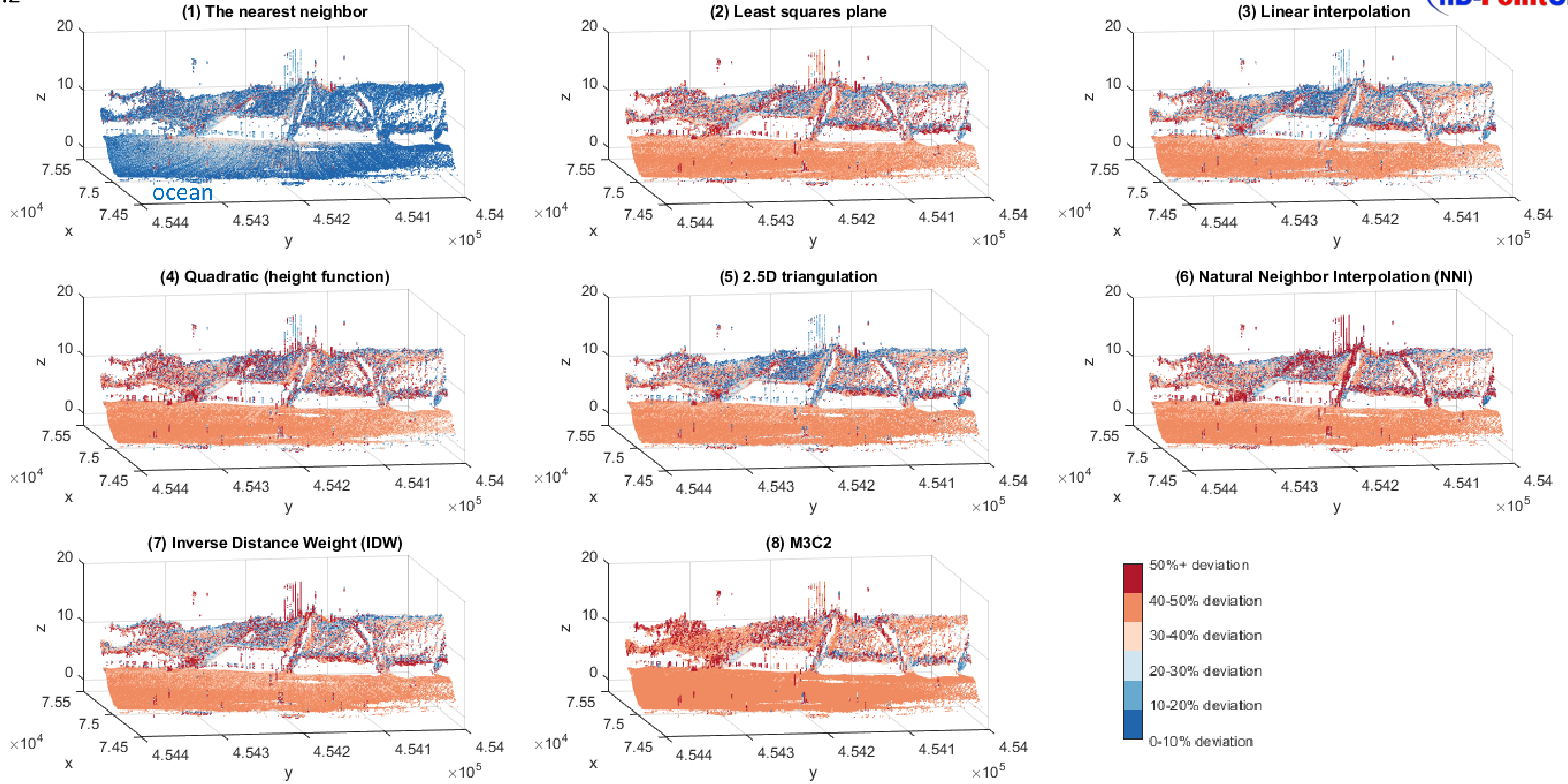
dx=0
dy=0
dz=1

Average intra-distance = 0.396 m



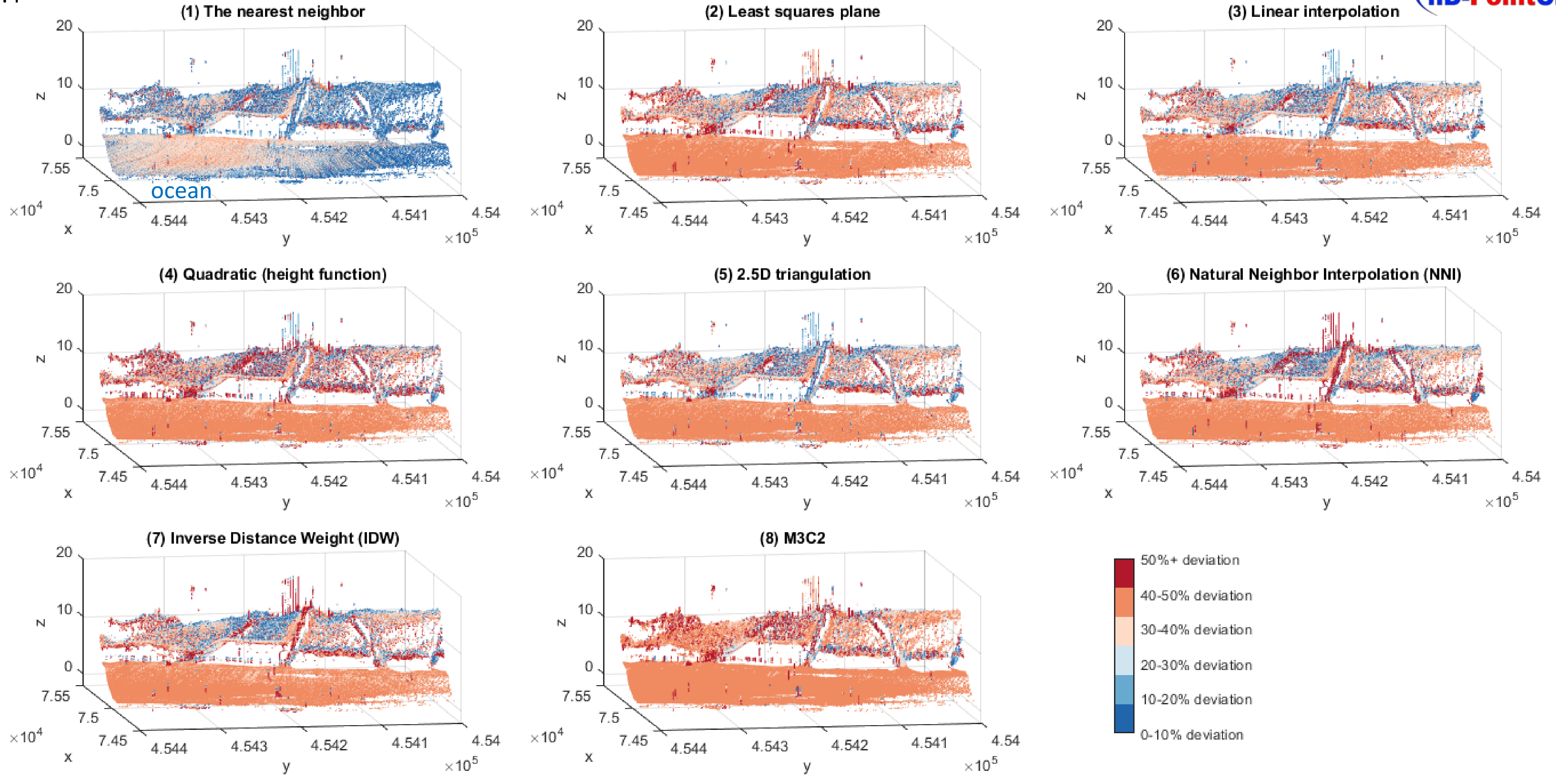
dx=0.2
dy=0.2
dz=0.2

Average intra-distance = 0.396 m



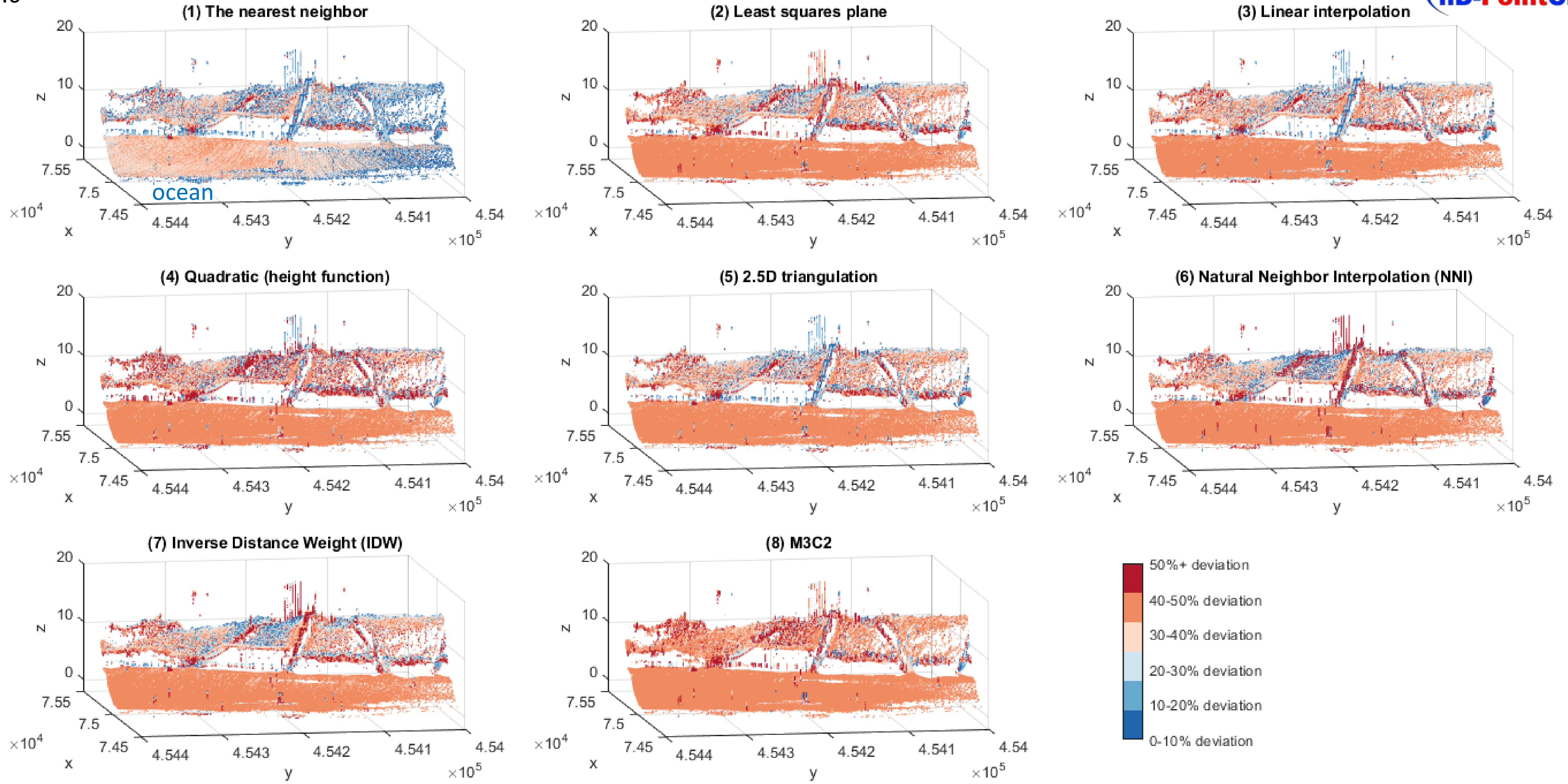
dx=0.4
dy=0.4
dz=0.4

Average intra-distance = 0.396 m



dx=0.6
dy=0.6
dz=0.6

Average intra-distance = 0.396 m



# 4 Conclusions

**1)** In the case of the beach, most methods perform similarly for the dz (displacement in z). This is because for most points of the compared cloud, the neighbors of the reference cloud represent the same section of the beach very well.

When applying a horizontal offset, most methods fail; only the Nearest neighbor behaves better.

**2)** In the case of an object (bunny), Nearest neighbor, Natural Neighbor Interpolation, and Inverse Distance Weight are the ones that best capture the applied dz. In the case of horizontal displacement, Natural Neighbor Interpolation performs better than the rest of the methods.

It is observed that the results are sensitive to the displacement direction, i.e. if displacement is applied in z, for the points close to the top, the distance calculated in said points is closer to the displacement applied; in horizontal displacement, the points located on the sides are the ones that best capture the applied displacement.

# 4 Conclusions

**3)** In the case of lake database, most methods capture well the vertical displacement in the terrain. Regarding the trees, the nearest neighbor, Natural Neighbor Interpolation, and Inverse Distance Weight stand out.

On horizontal displacement, most of the methods fail; only Nearest neighbor and Natural Neighbor Interpolation are good.

In horizontal offsets, the results show that in objects (trees), the displacement is better captured in the points close to the sides. Regarding terrain, only the Nearest neighbor seems to show better results.

**4)** Finally, it is observed that the results depend on whether the displacement is less or greater than the intra-distance, being better in the first case.

# 5 Future work

- We tested artificial displacement (dx,dy,dz), this only considers the translation effect; other transformations (e.g. rotation) can be tested in the future.

- We are analyzing the AHN database (the lidar data for the whole Netherlands).

- We are analyzing the results to select the 'best method', the results suggest that Nearest Neighbor is the suitable method.

- We are working on implementing Nearest Neighbor efficiently in a Database Management System (DBMS) using the Space Filling Curve (SFC) key for the whole AHN2-3-4.

# Preliminary Results

## SFC-based ultra-fast change detection

### Morton key-based

**Preparation (performed on the reference cloud)**

1 Scaling and offsetting

2 Morton key calculation

3 Sorting Morton key column

In progress…

**Batch Processing for Nearest Neighbor Euclidean Distance Calculation (performed on the compared cloud with Preparation's output)**

1 Scaling a given point x,y

2 Morton key calculation for x_scaled,y_scaled

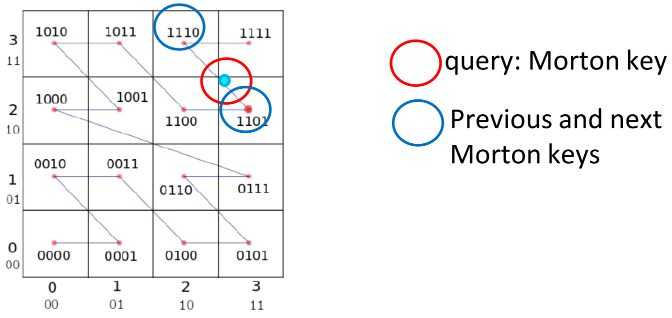
3 Searching the previous and next Morton key of the calculated Morton key

4 Decoding these two Morton keys

5 Unscaling both scaled points (previous and next)

6 Finding the nearest neighbor based on Euclidean distance calculation

Output d and dx,dy,dz

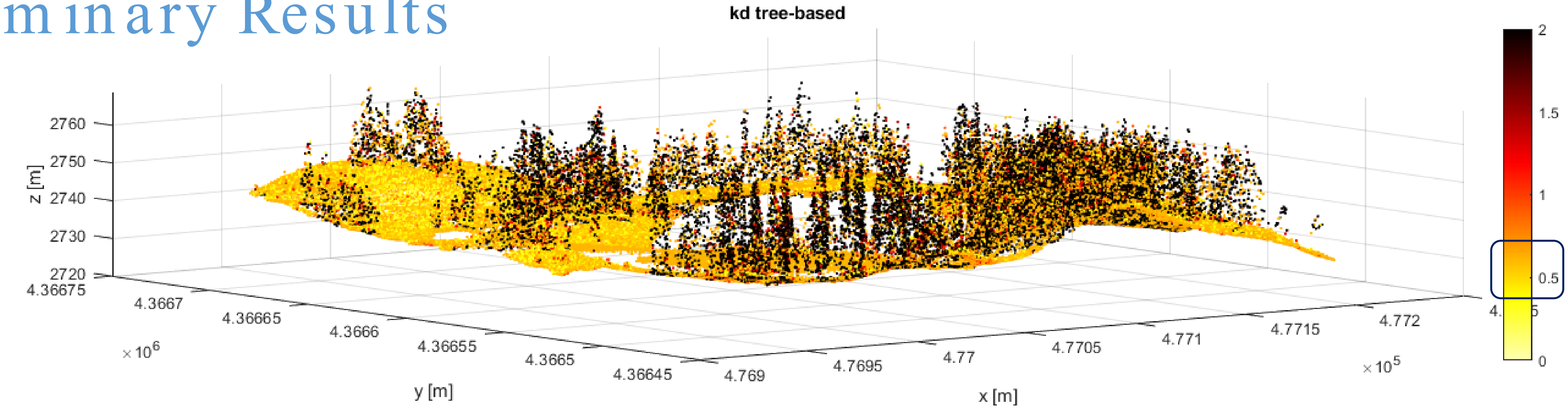


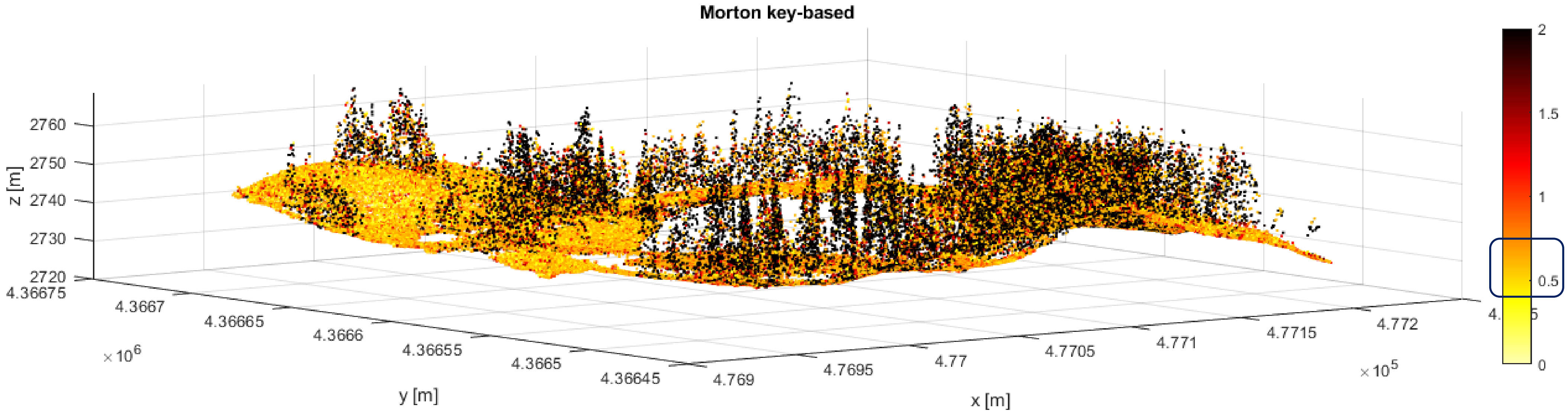

Liu, H. (2022)

# Preliminary Results

Artificial displacement
dx=-0.25
dy=-0.25
dz=-0.5

d=0.6124

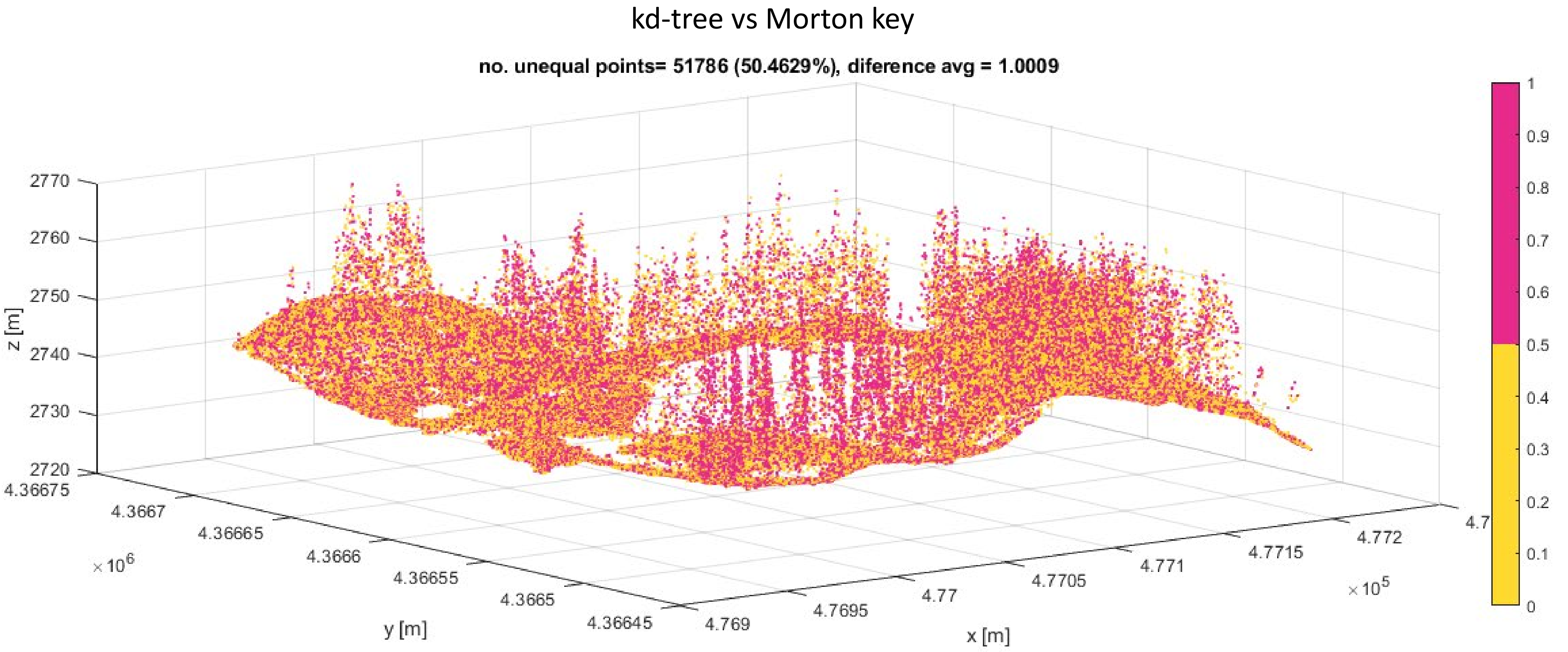


# Preliminary Results

Artificial displacement
dx=-0.25
dy=-0.25
dz=-0.5

d=0.6124

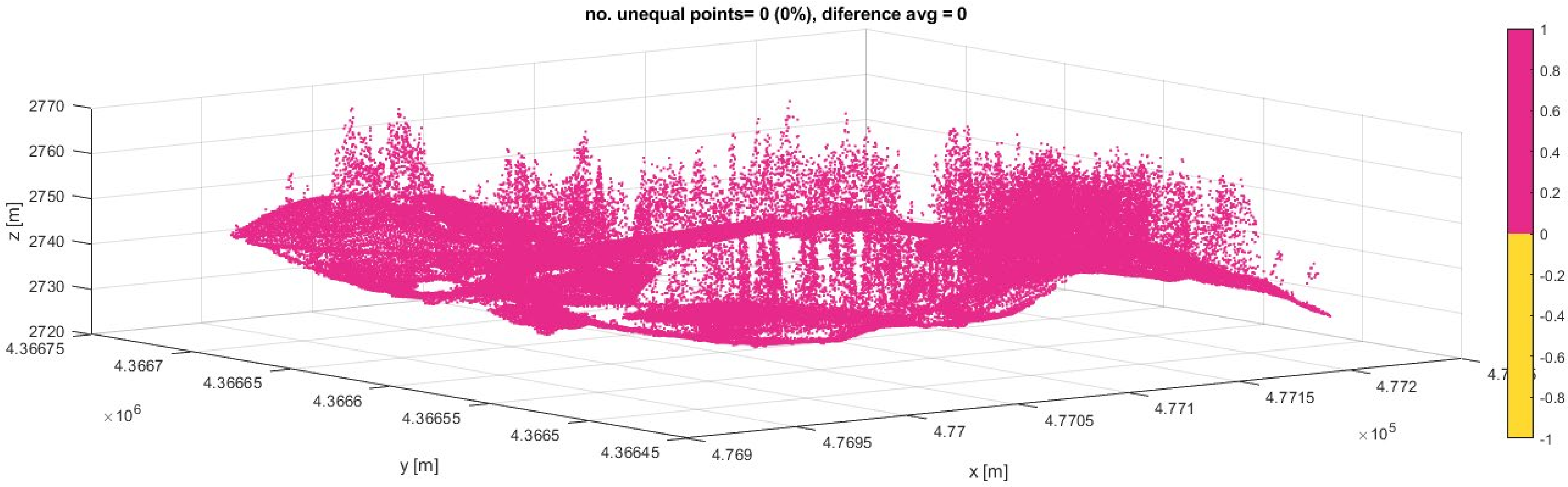


# Preliminary Results

Artificial displacement
dx=0
dy=0
dz=1

d=1

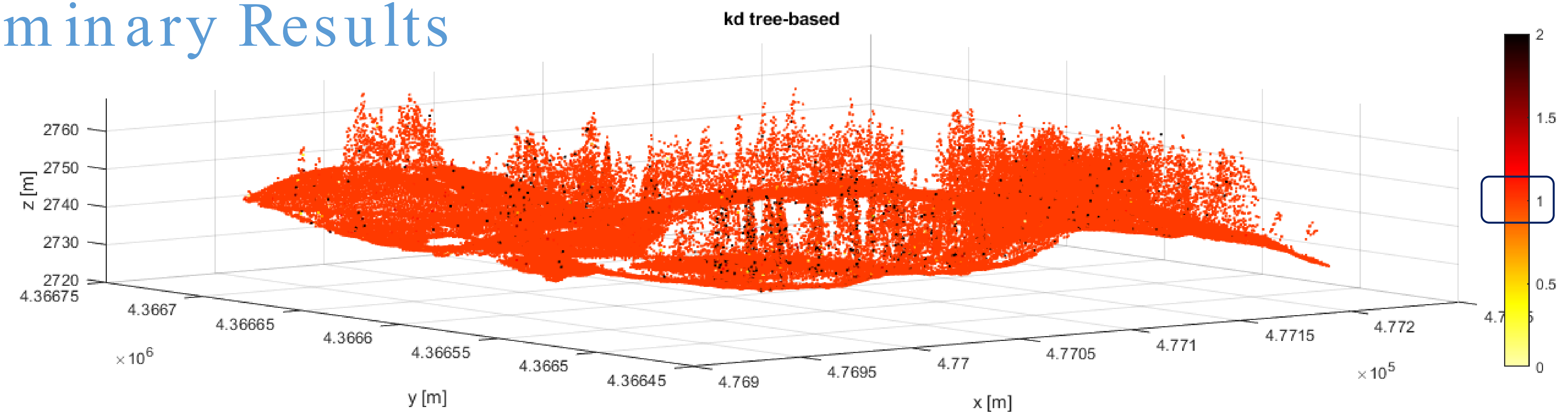
kd-tree vs Morton key

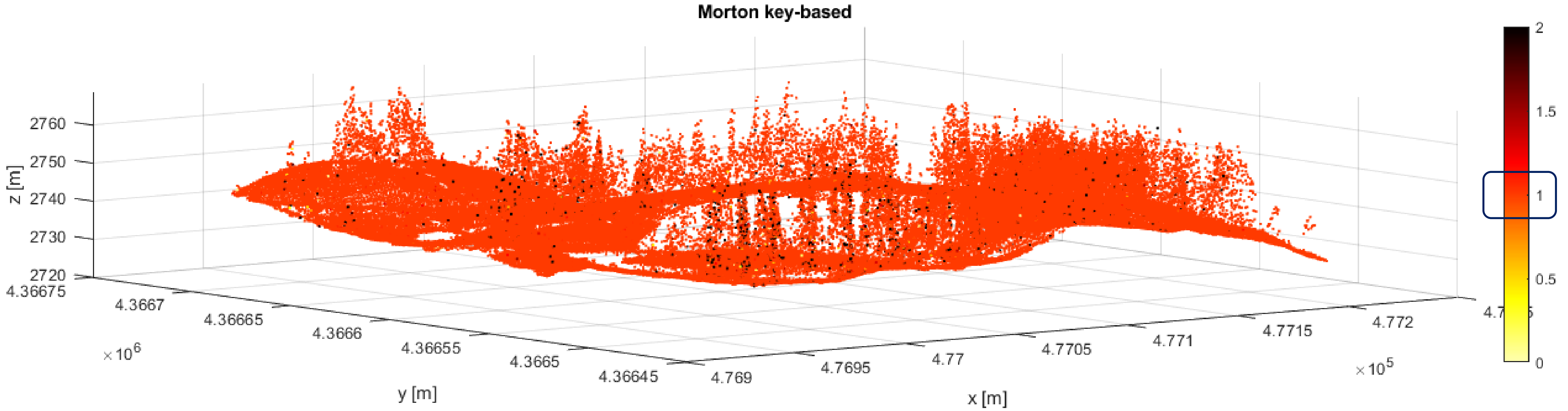


# Preliminary Results

Artificial displacement
dx=0
dy=0
dz=1

d=1



# Quality analysis of point cloud change detection algorithms & SFC-based ultra-fast change detection

Vitali Diaz
v.diazmercado@tudelft.nl

nD-PointCloud

http://nd-pc.org/

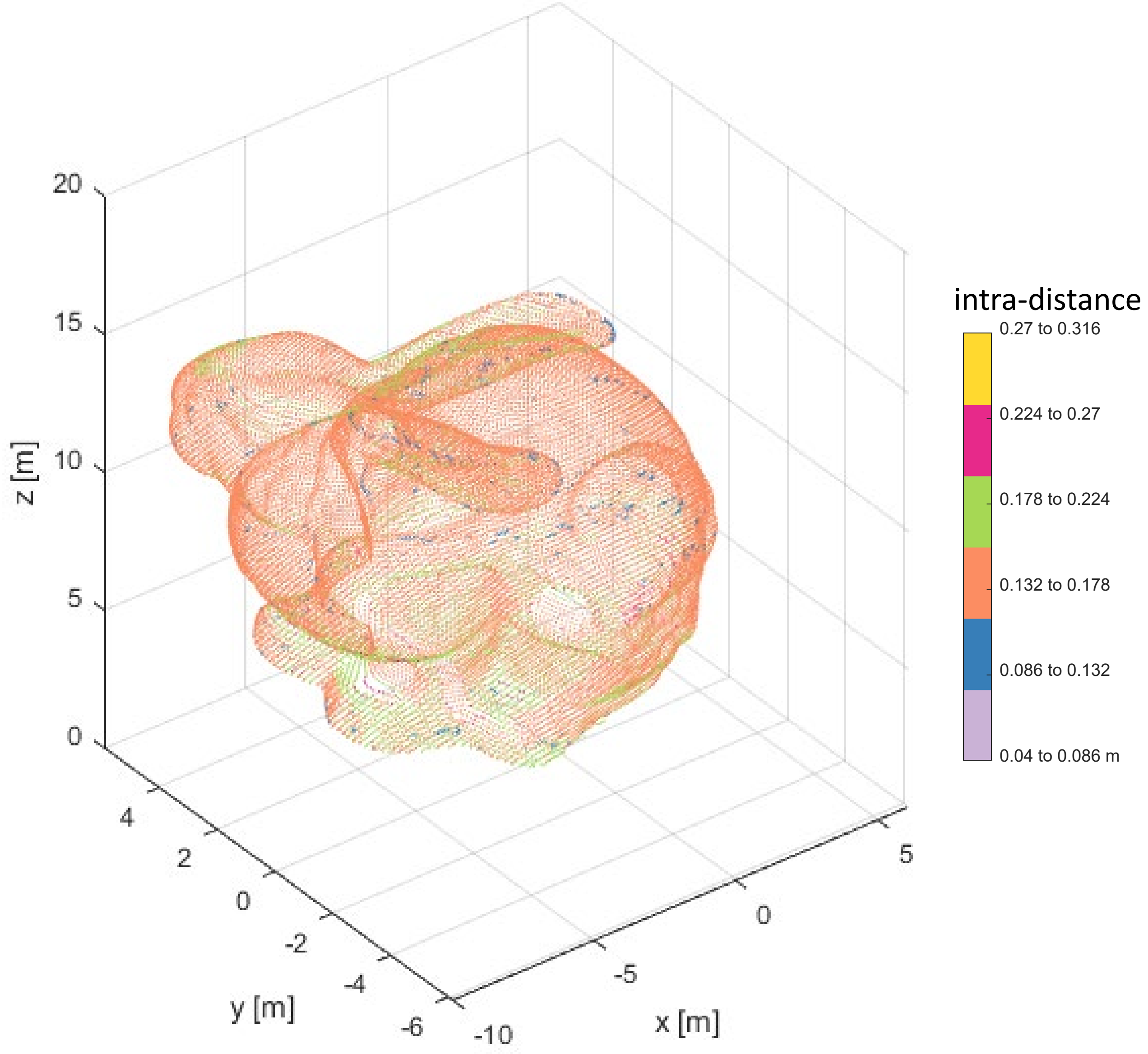
http://www.gdmc.nl/publications/

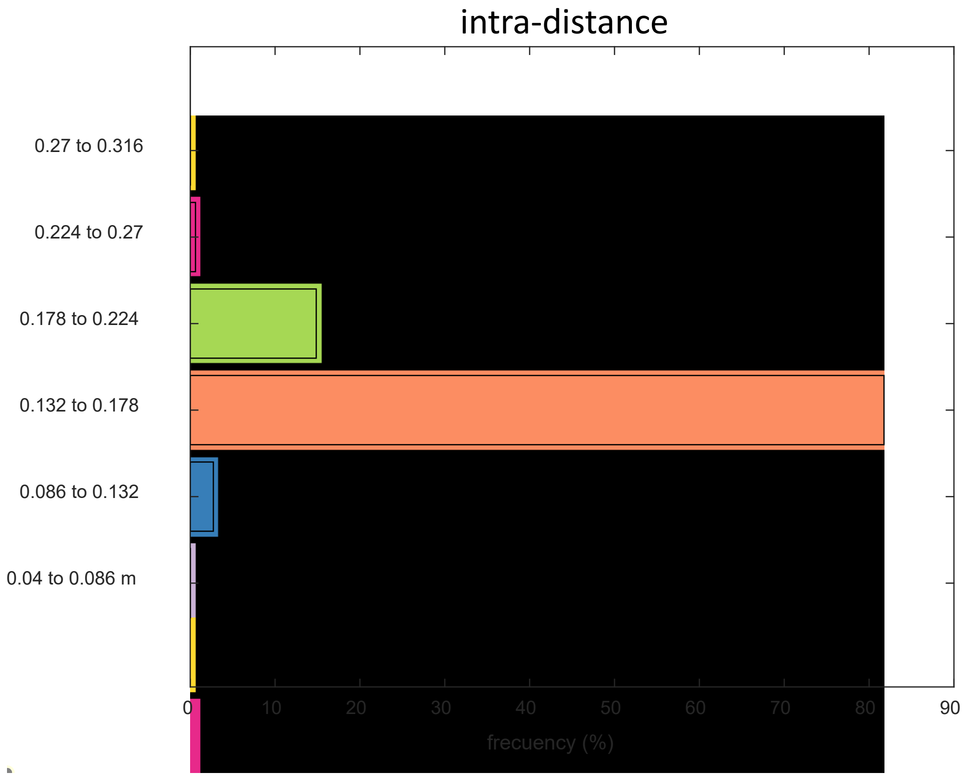
TUDelft

netherlands eScience center

# Appendix

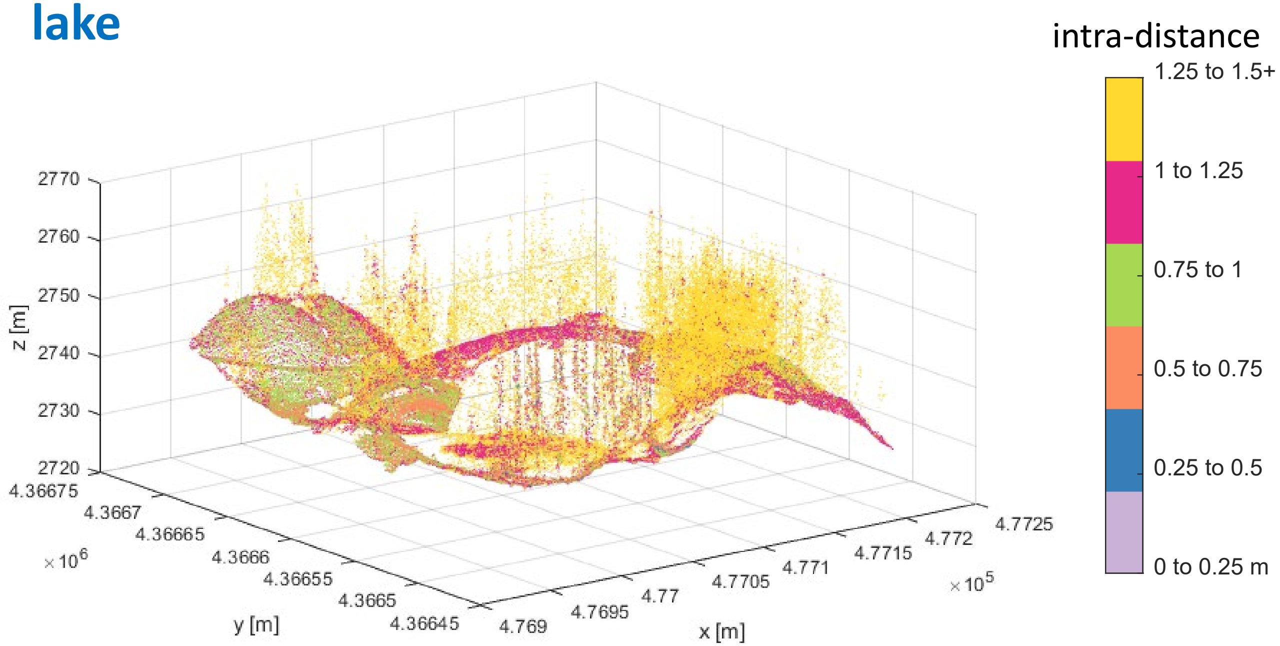
# bunny scaled

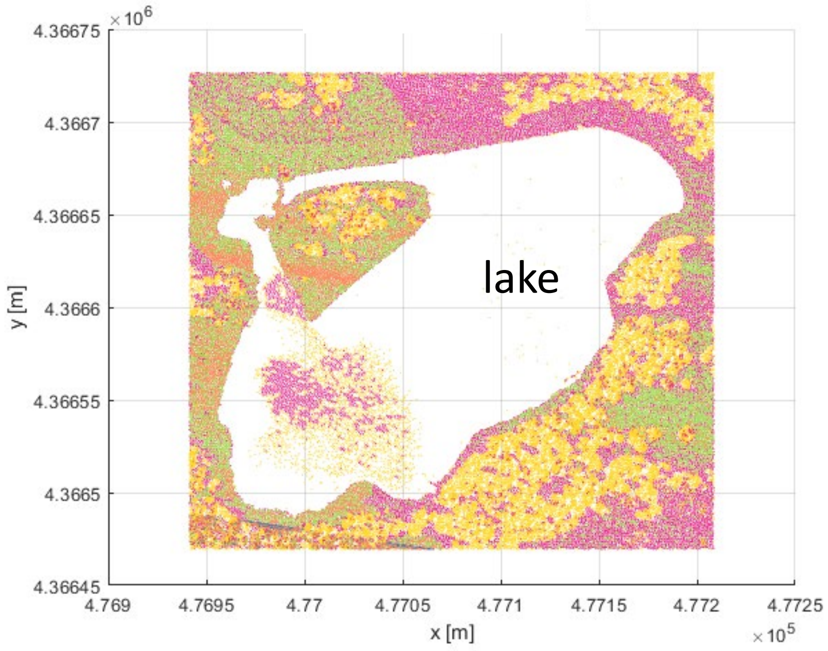


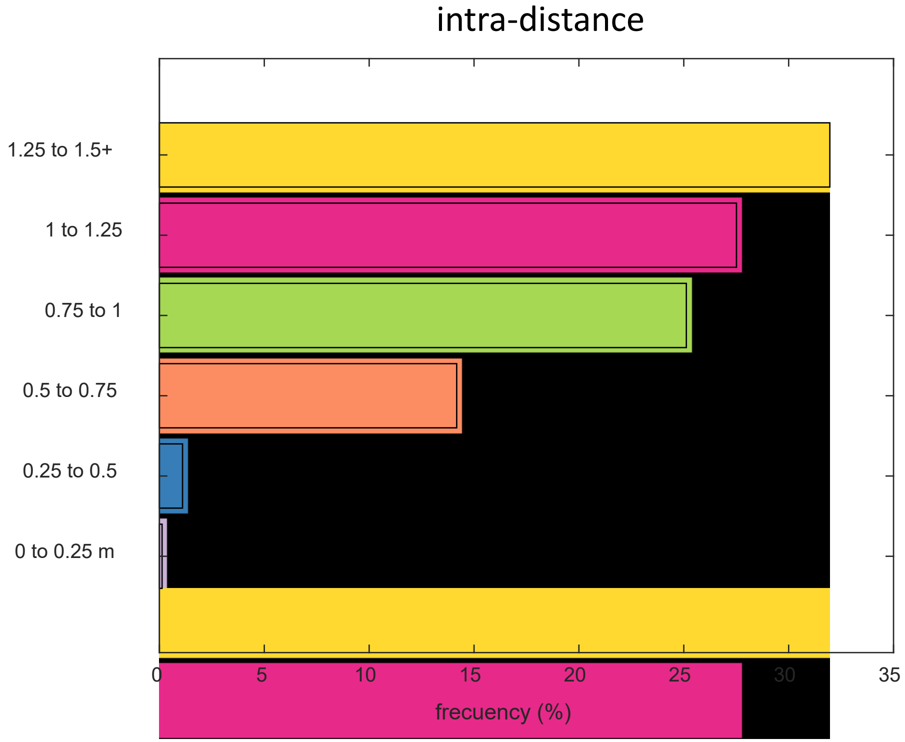




intra-distance avg = 0.16 m

# lake

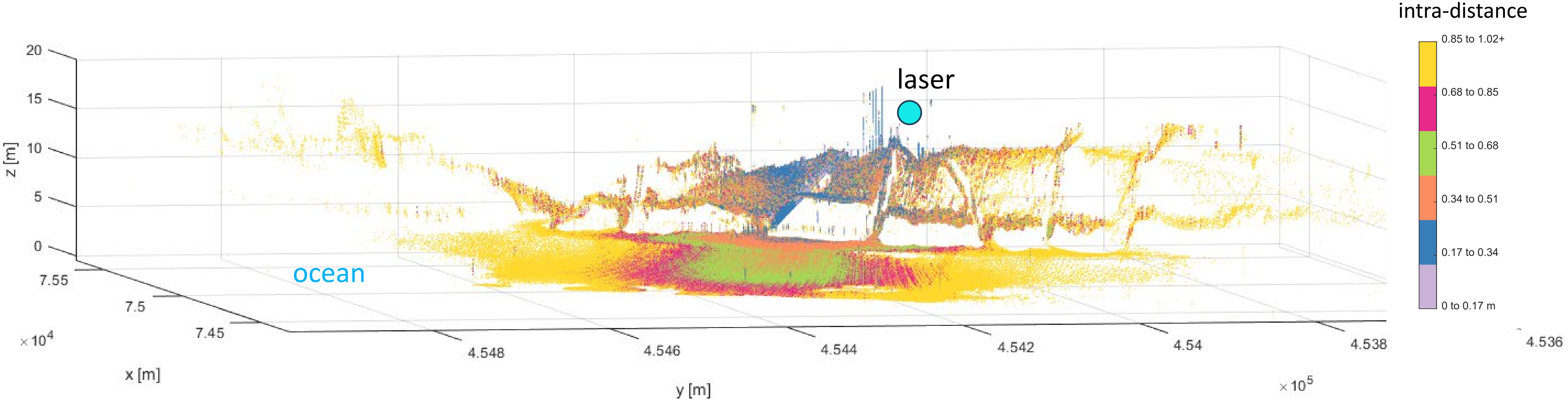


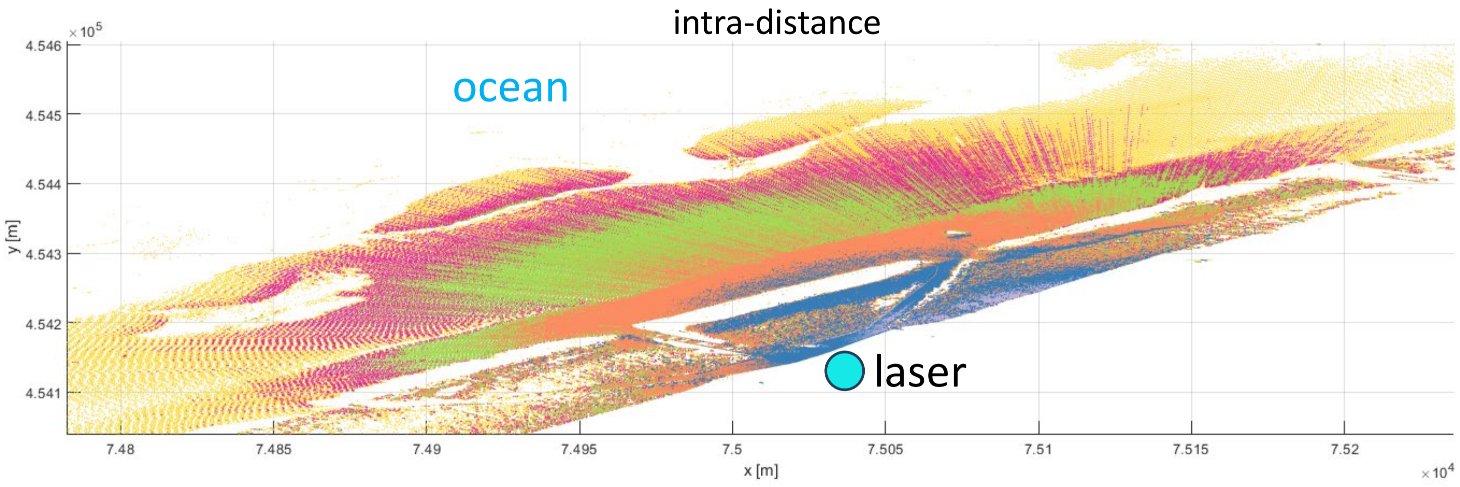



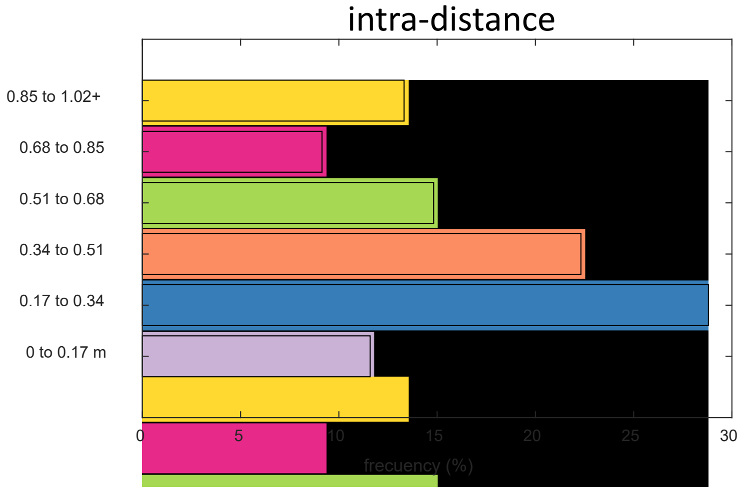




intra-distance avg = 1.06 m

# coastscan

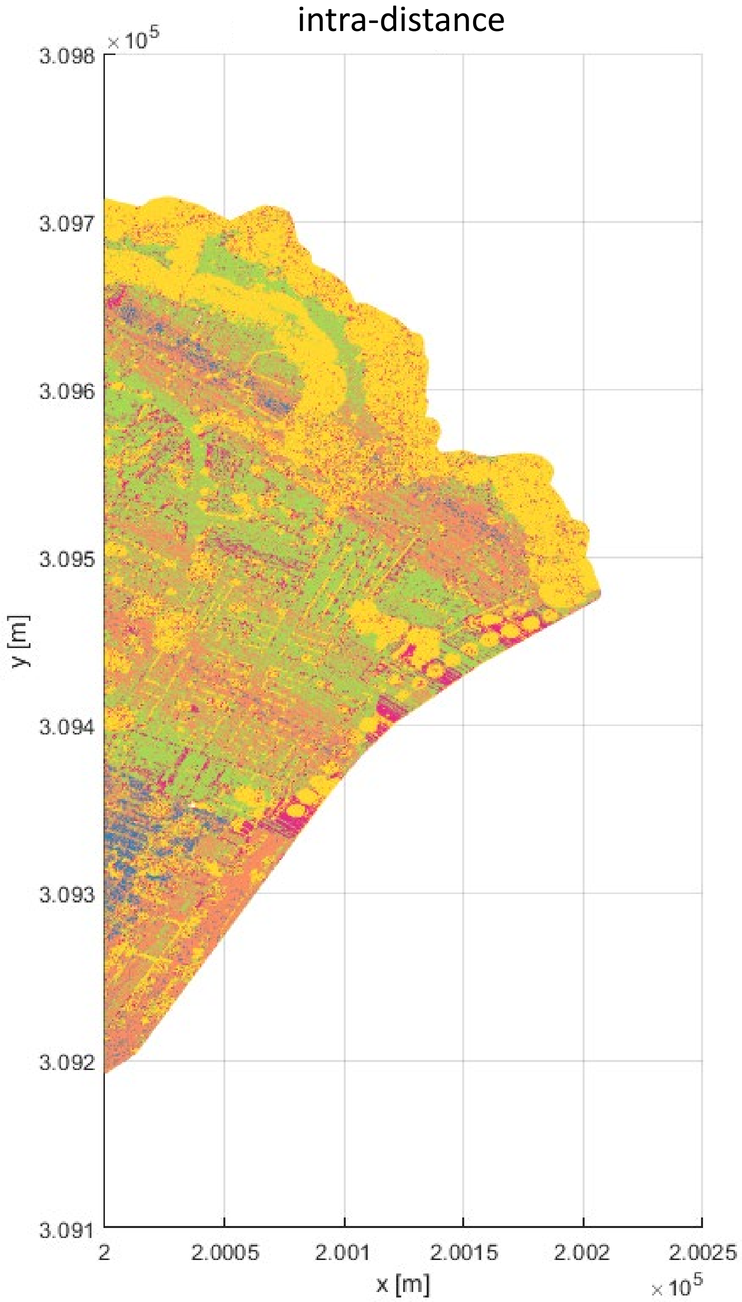


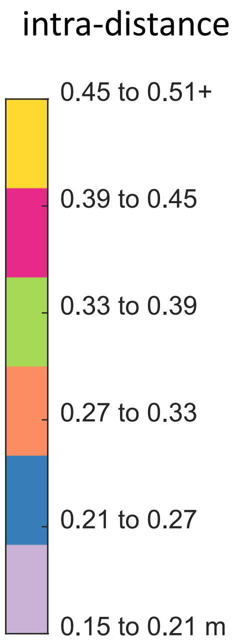



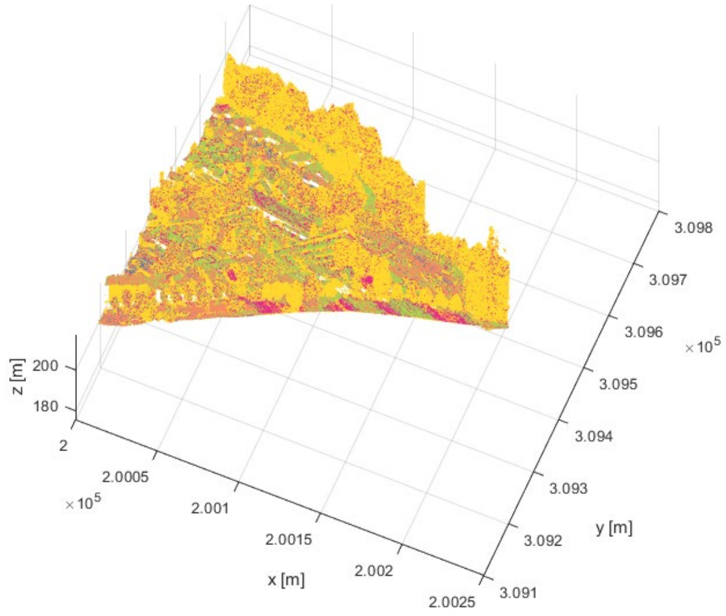



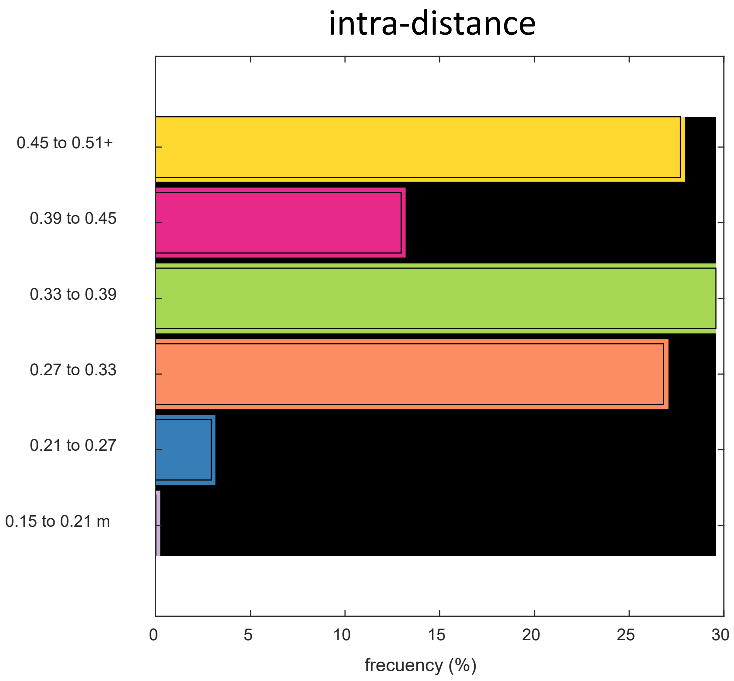

intra-distance avg = 0.396 m

# AHN3

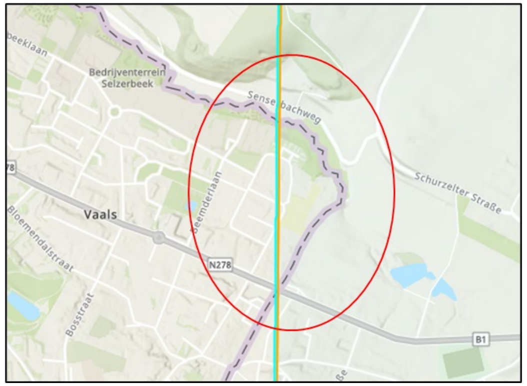


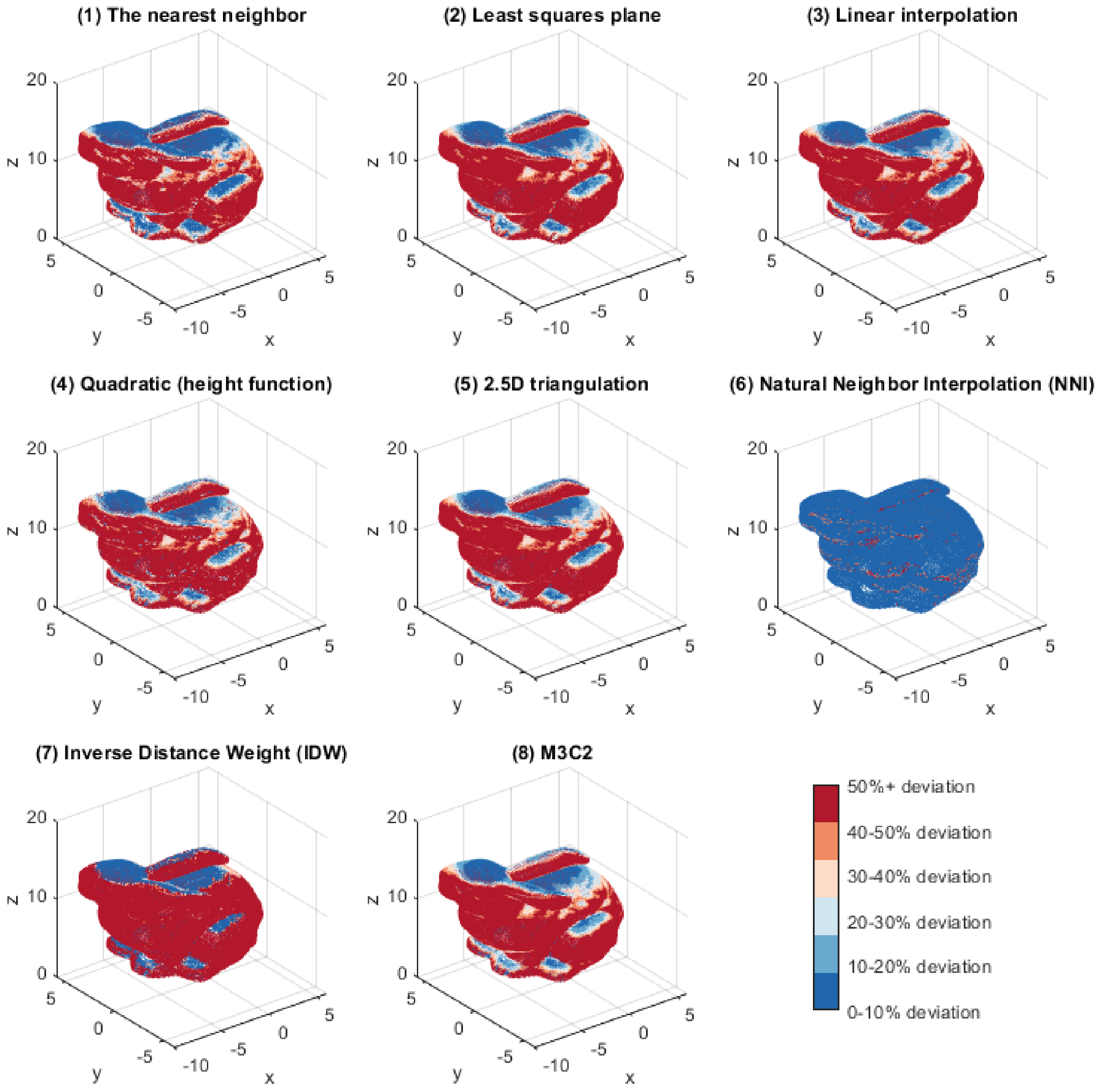

intra-distance avg = 0.363 m



# Bunny

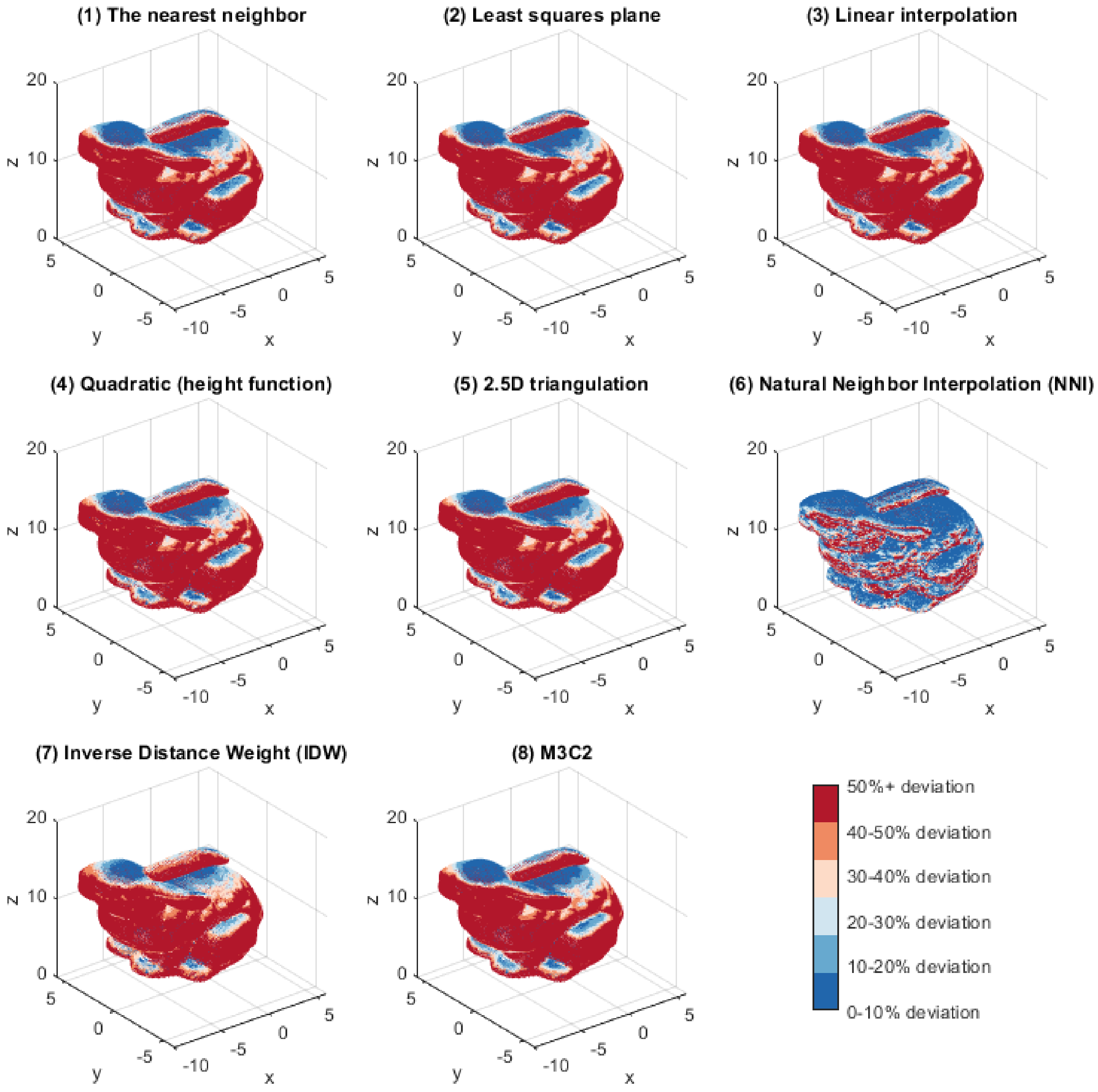
*dx*=0
*dy*=0
*dz*=0.15

Average intra-distance = 0.16 m



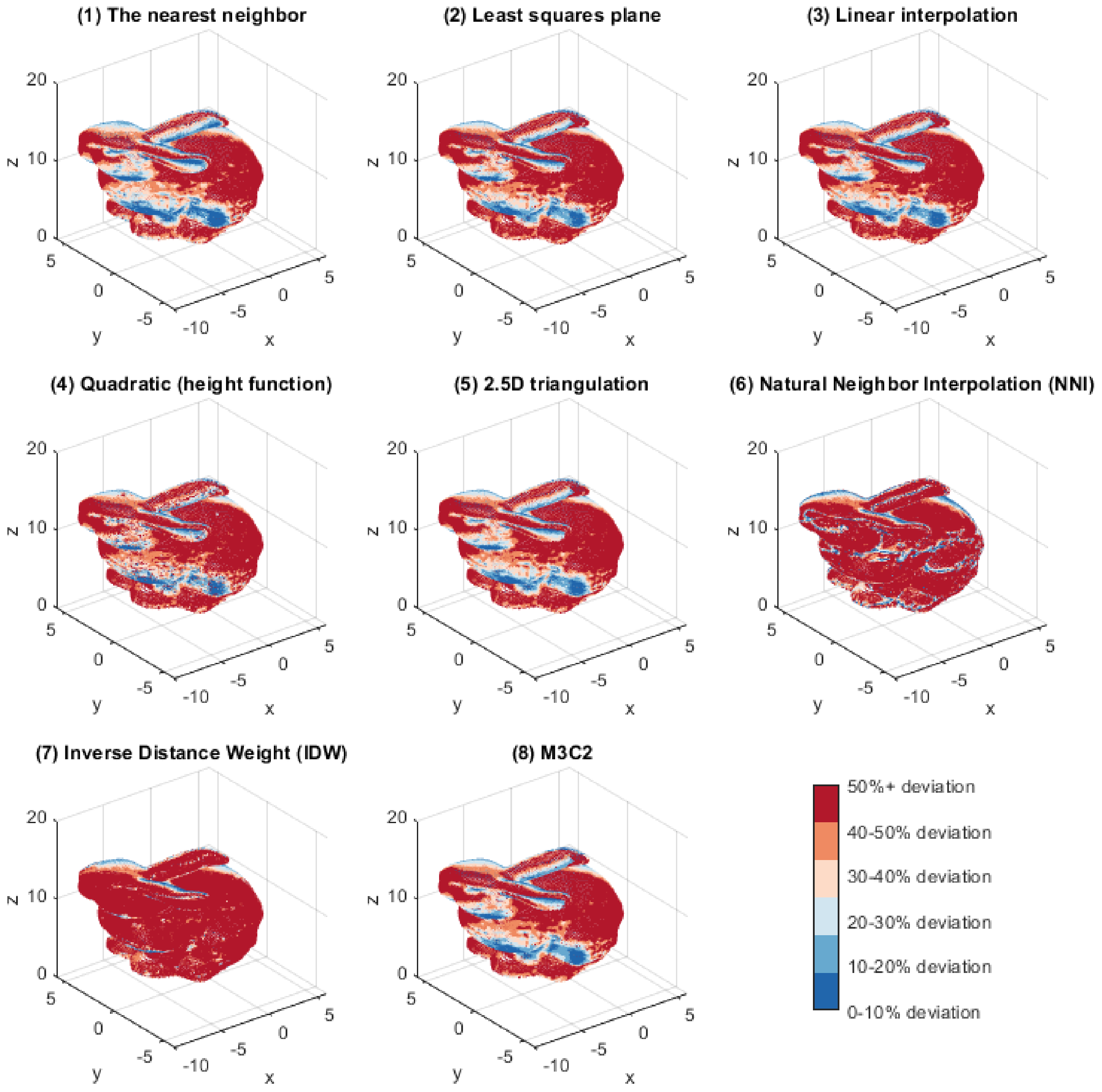
dx=0
dy=0
dz=0.4

Average intra-distance = 0.16 m



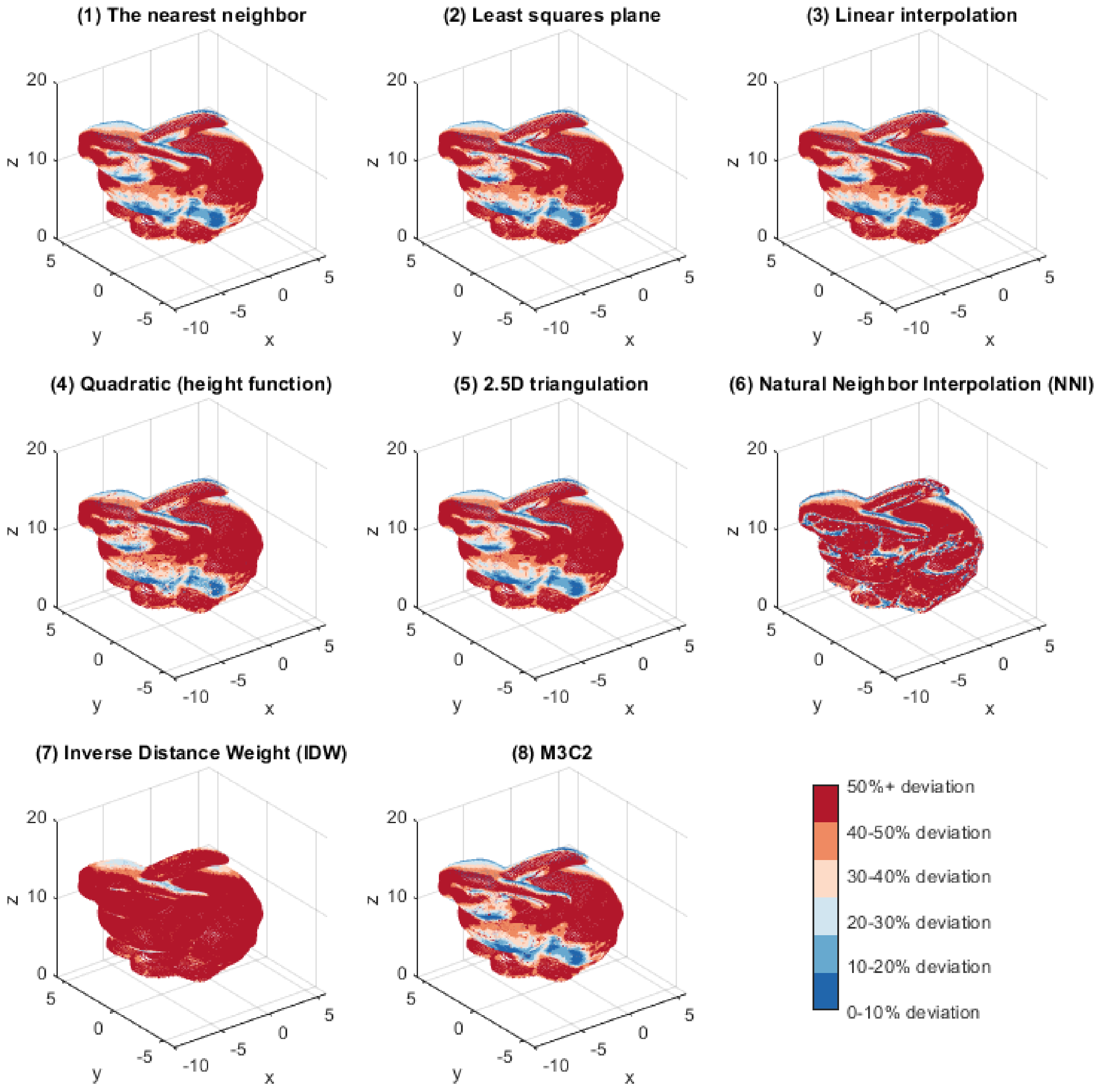
dx=0.15
dy=0.15
dz=0.15

Average intra-distance = 0.16 m



dx=0.4
dy=0.4
dz=0.4

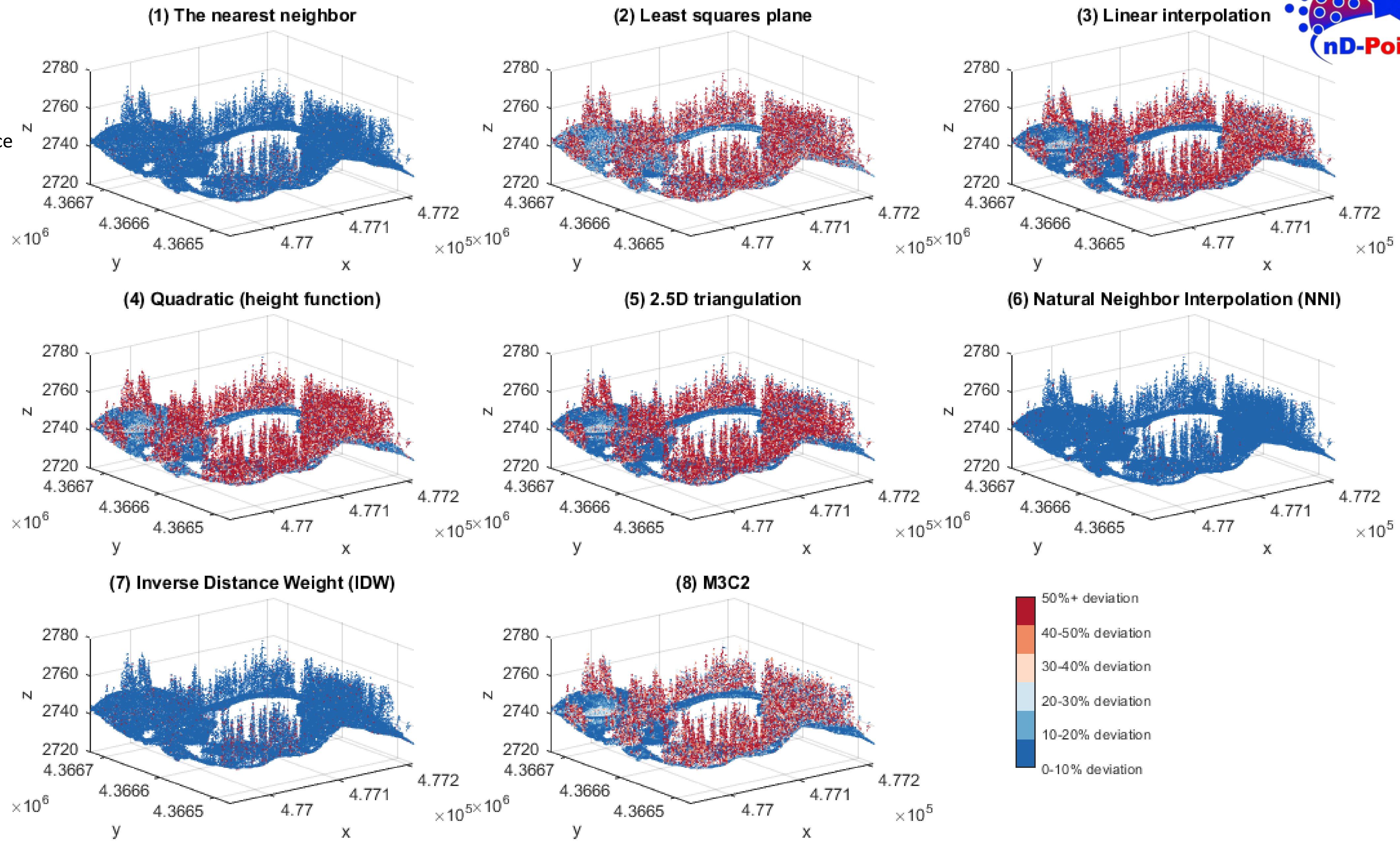
Average intra-distance = 0.16 m



# Lake

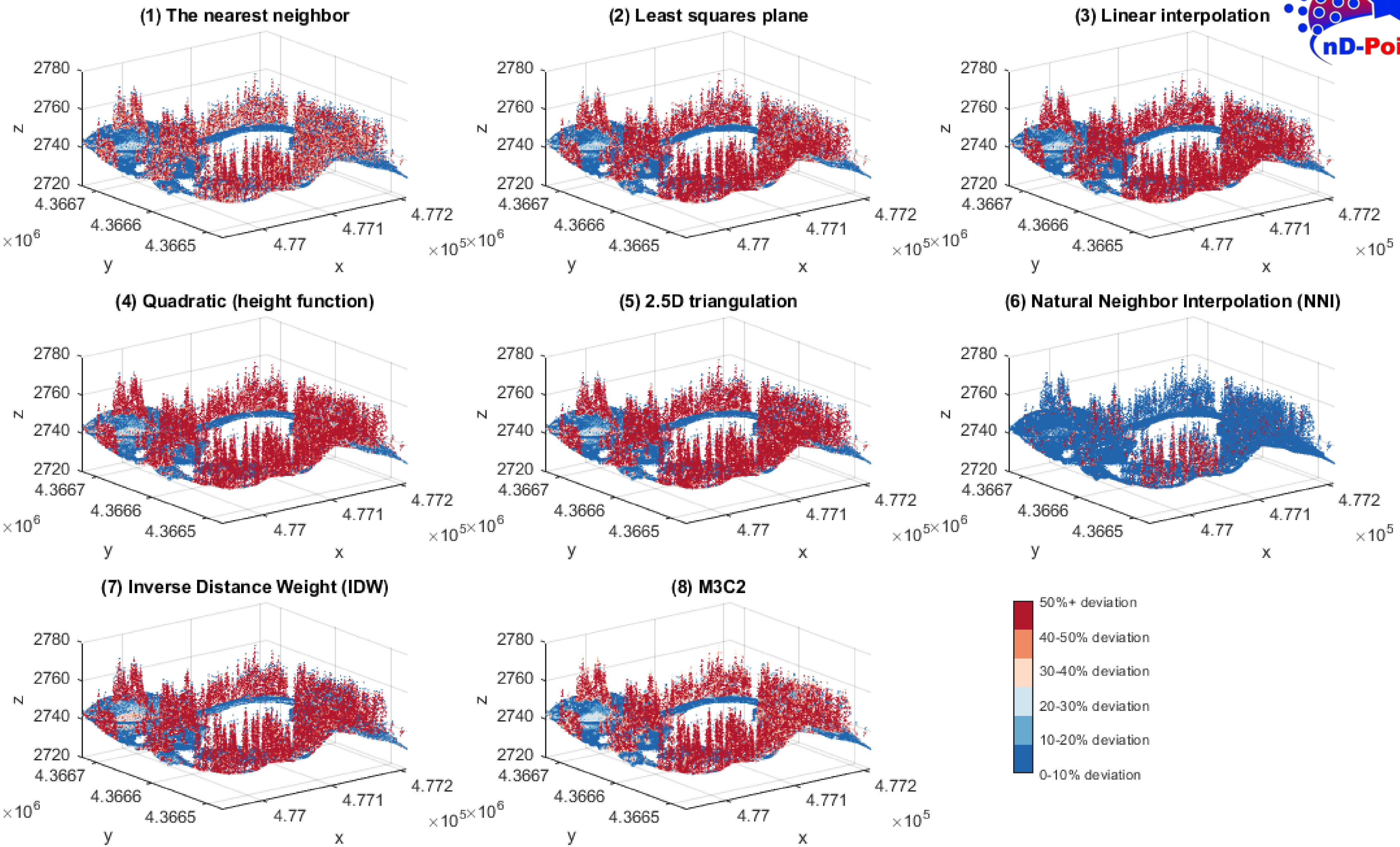
*dx*=0
*dy*=0
*dz*=0.5

Average intra-distance = 1.06 m



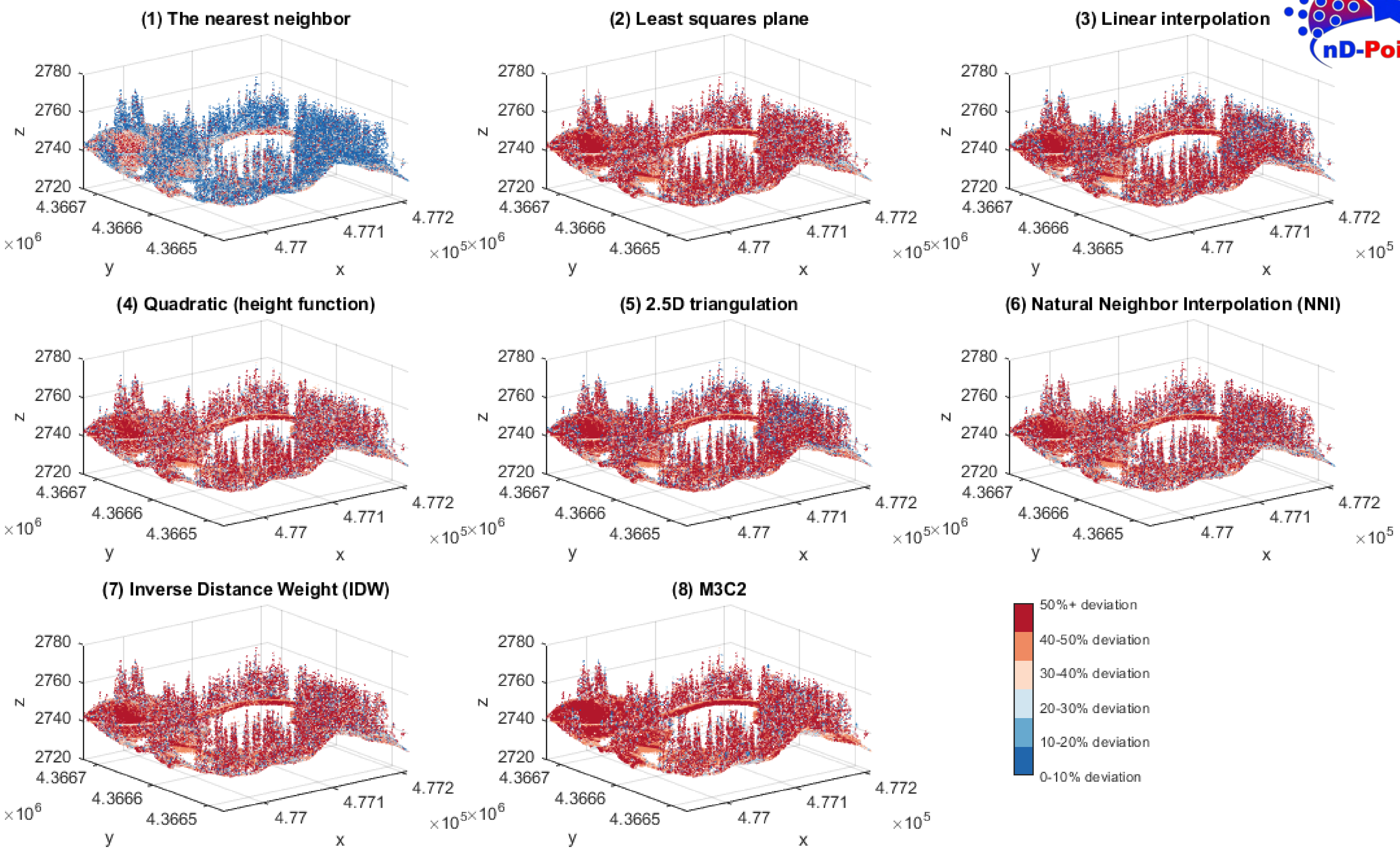
dx=0
dy=0
dz=2

Average intra-distance = 1.06 m



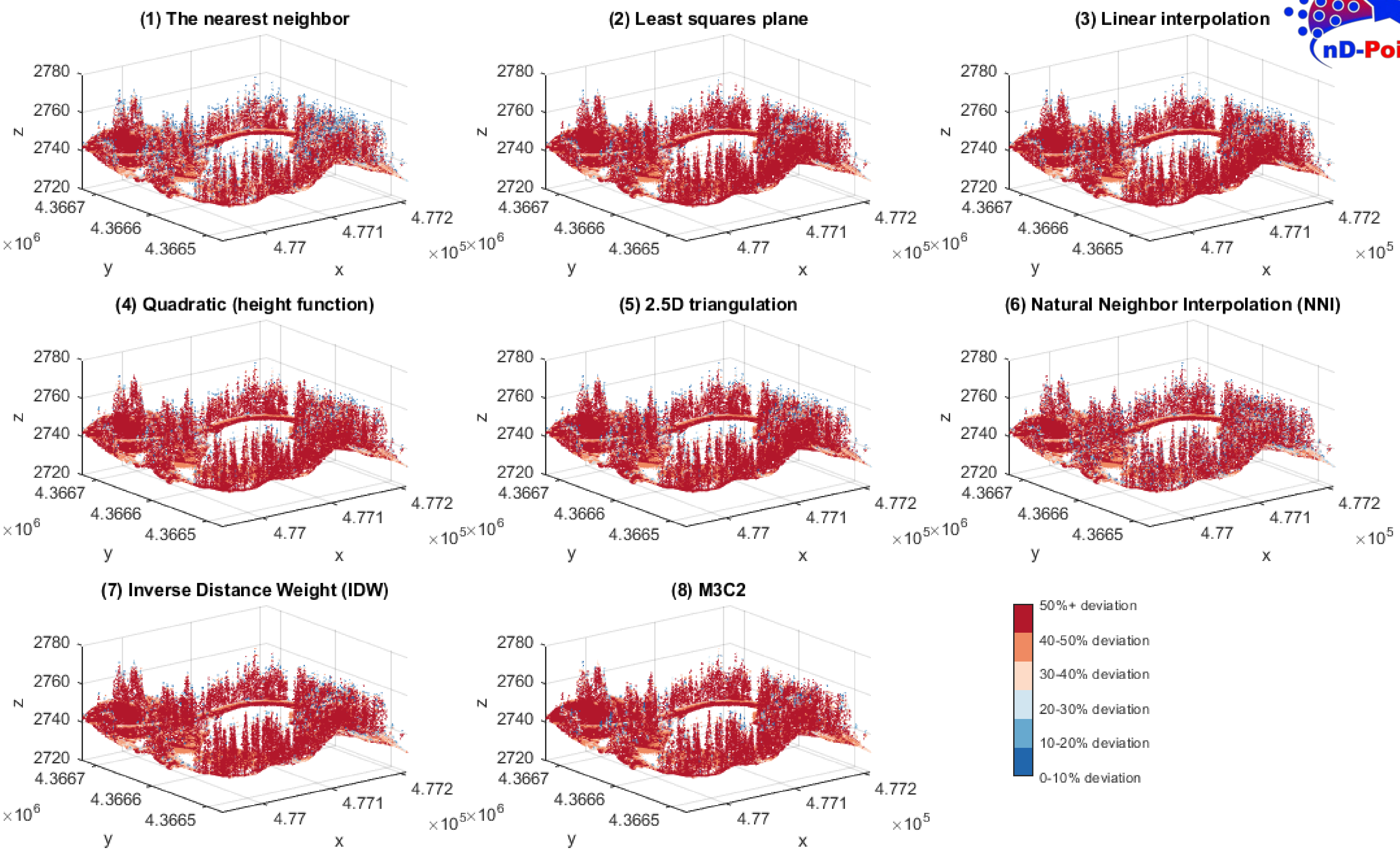
dx=0.5
dy=0.5
dz=0.5

Average intra-distance = 1.06 m



dx=2
dy=2
dz=2

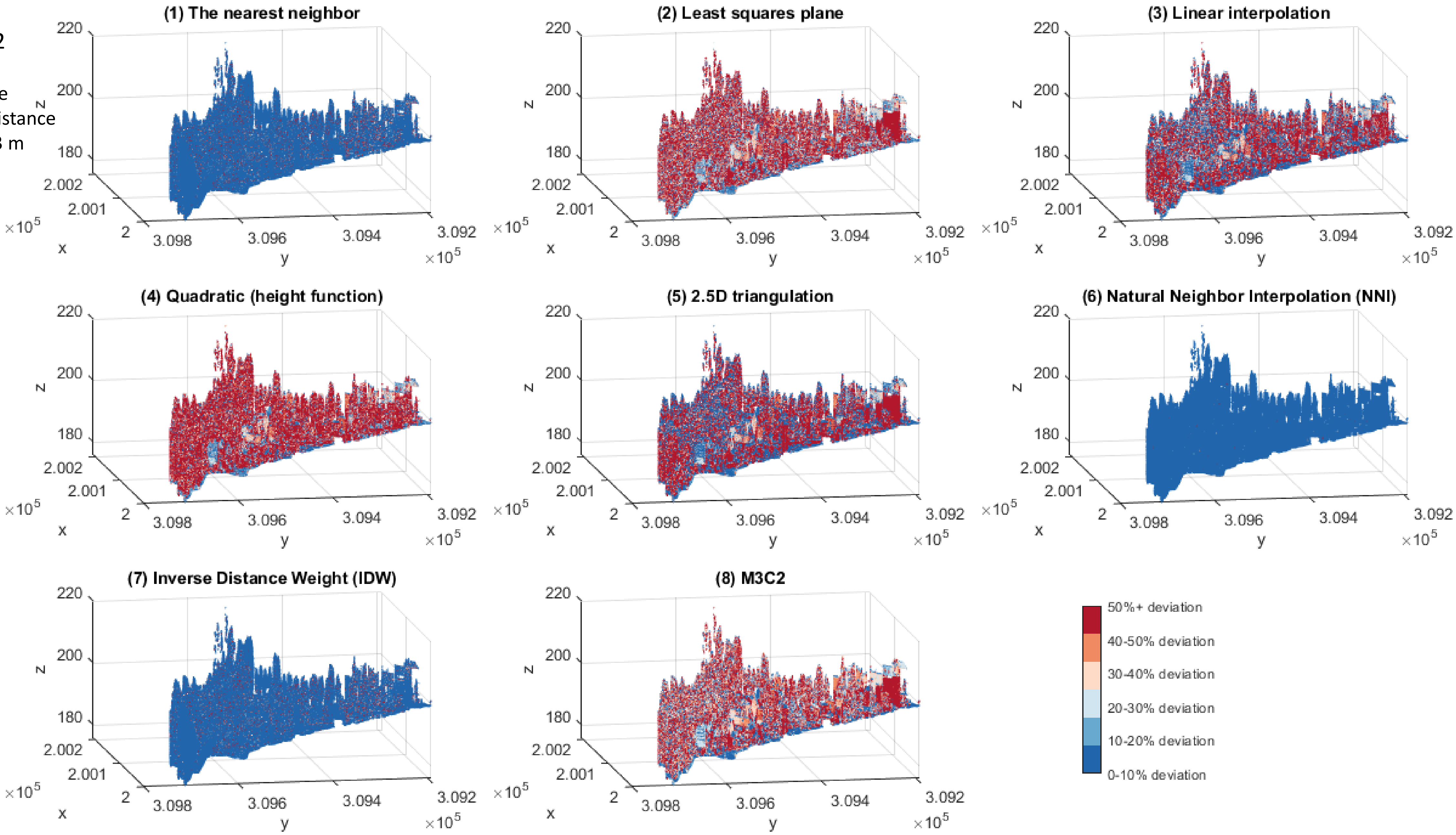
Average intra-distance = 1.06 m



# AHN3

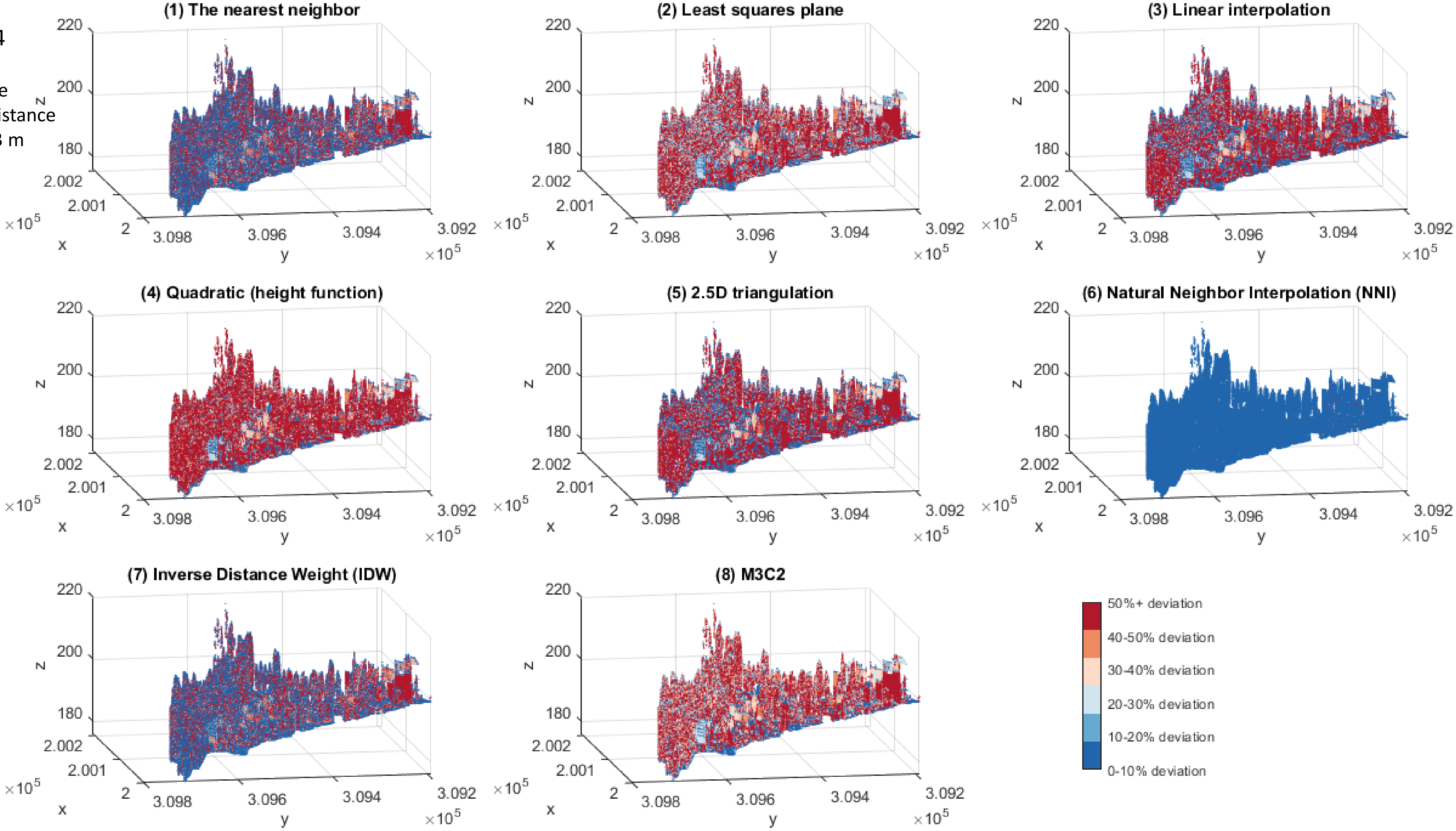
dx=0
dy=0
dz=0.2

Average intra-distance = 0.363 m



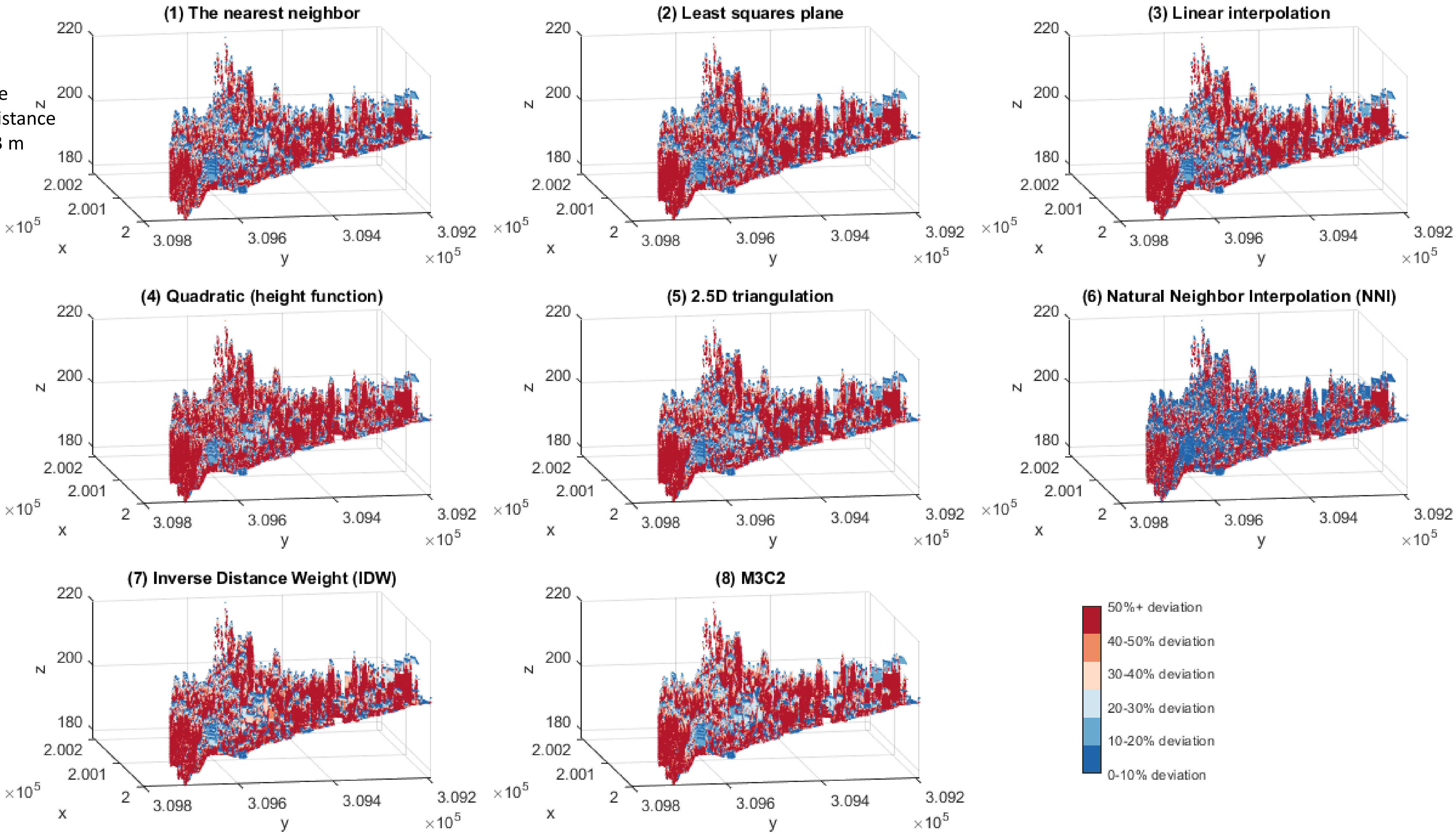
dx=0
dy=0
dz=0.4

Average intra-distance = 0.363 m



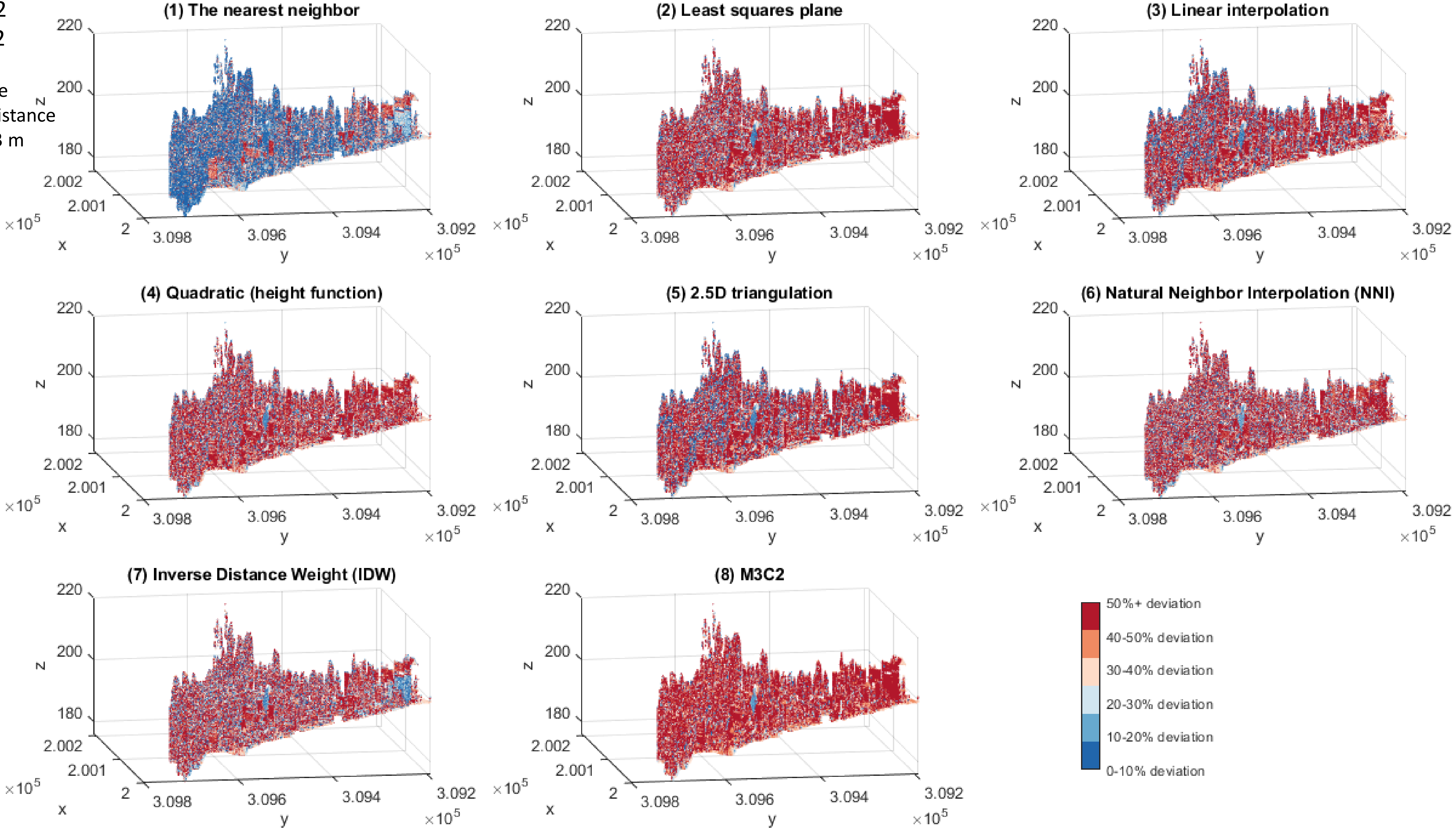
dx=0
dy=0
dz=2

Average intra-distance = 0.363 m



dx=0.2
dy=0.2
dz=0.2

Average intra-distance = 0.363 m



dx=0.4
dy=0.4
dz=0.4

Average intra-distance = 0.363 m