

Pole Tagger

# Labeling Guide

15 May 2023



# General Goal



The City of Amsterdam is responsible for managing and maintaining all street lights in the city. Using point cloud data, we help automate this process.



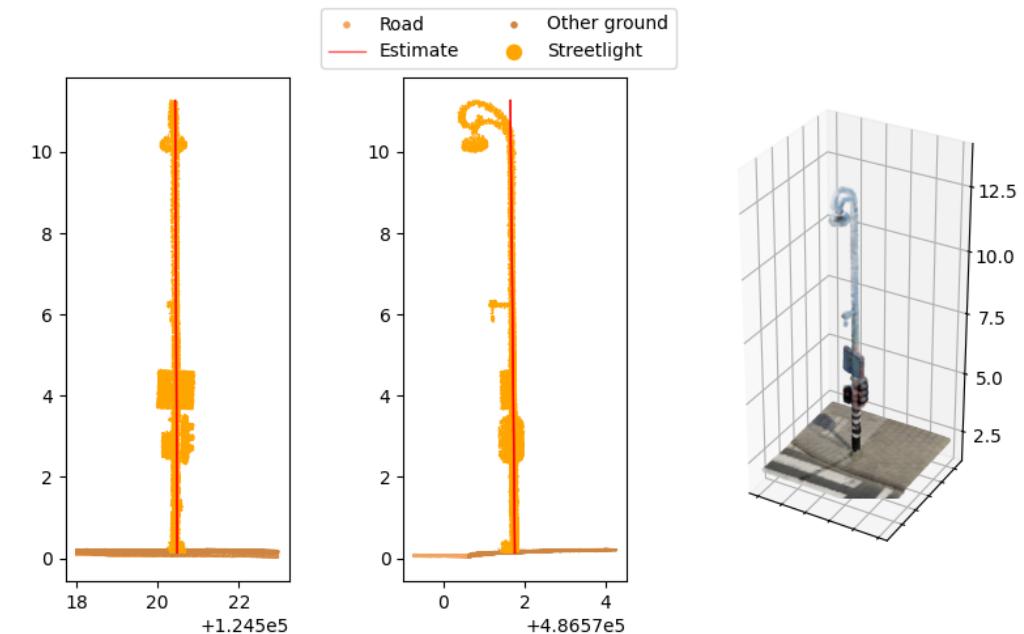
An initial pipeline finds objects which are likely to be street lights. It also calculates the location, height and angle of each potential street light.



In this next step, we aim to validate the extracted objects and refine these most important statistics so that everything is correct.



This is done by iteratively inspecting all object based on 2 different perspectives, as well as a bird-eye view of the point cloud around the object.



# Expected results



## Location

Within the registers, the municipality is allowed a maximum of 15cm deviation from the actual location of each light. We consider predictions correct if the bottom point of the fitted red line is anywhere within the base of the light (not necessarily the perfect center).



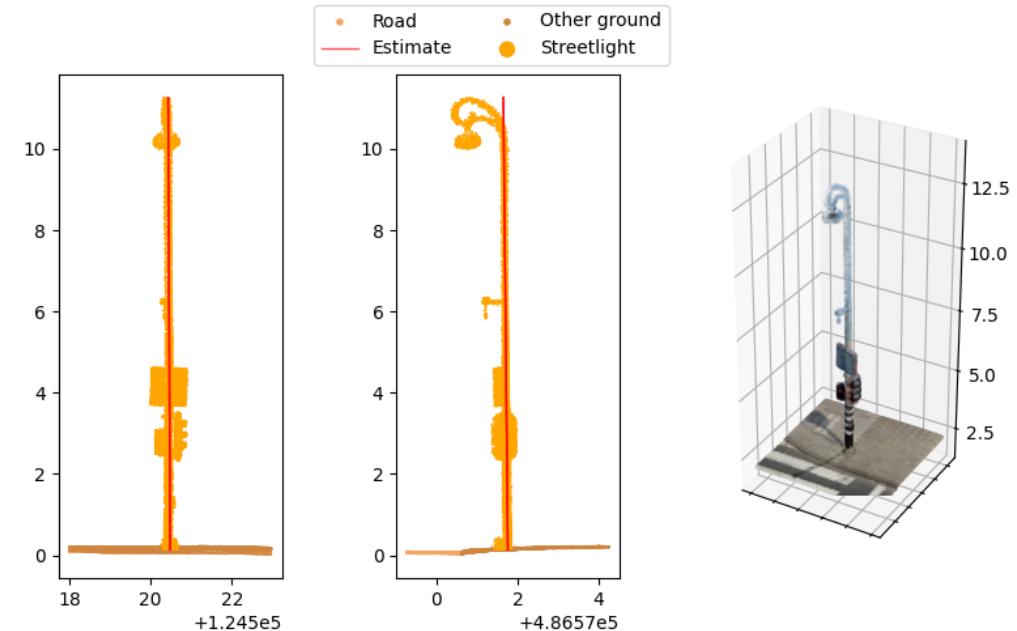
## Height

We measure until the highest point, including attachments. We only need to be able to distinguish between different height categories, so predictions which deviate by at most 25cm from the real (visible) height do not need to be corrected further.



## Angle

To calculate the correct angle of a pole, we need the red line to be (more or less) within the actual object for both perspectives.



# Annotation Setup: 3 Tasks



## Task 1: Validate poles and fits

For all predicted lights, indicate whether they are a street light with a correct fit, a street light which needs refitting, not a street light at all or it is unclear



## Task 2: Adjust fits

For all objects marked as a street light which needs refitting, draw a new line which correctly signifies the top, bottom and angle



## Task 3: Validate types

For all objects previously marked as street lights (with both correct or re-adjusted fit), validate the type and select a new one if needed



## Multiple attachments

In case there are multiple lamps on a single pole, we consider the highest one when annotating height and determining the type.



## Continue later

If you press [escape] at any point, you can always resume from where you left off by running the same command. Your intermediate results are saved after every example.

# Task 1: Validate poles and fits



## Run the program

by calling "`python3 pole_tagger.py --validate_pole`"  
in a terminal which has the same directory  
as the `pole_tagger.py` file



## Unsure?

Whenever in doubt, regard street lights as "unclear example" rather than (Not) a street light. This way, we can inspect the object further.



## Split up street lights

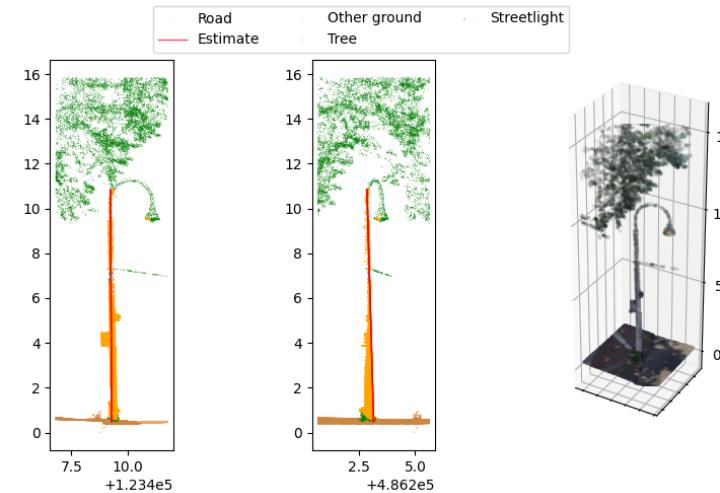
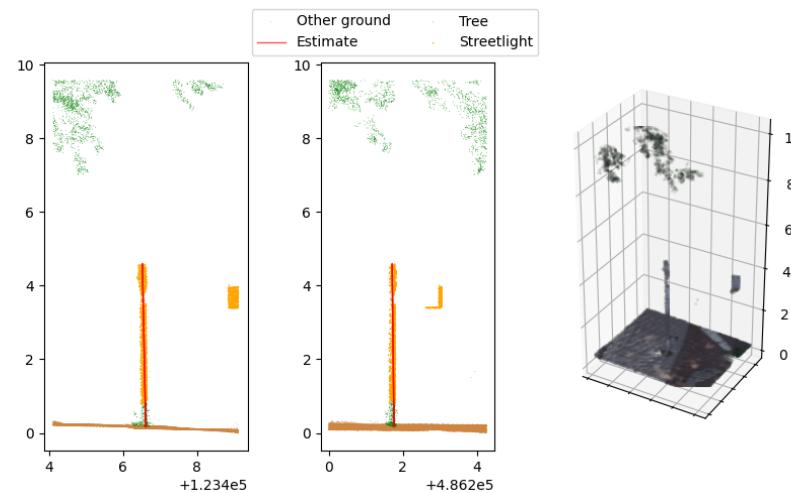
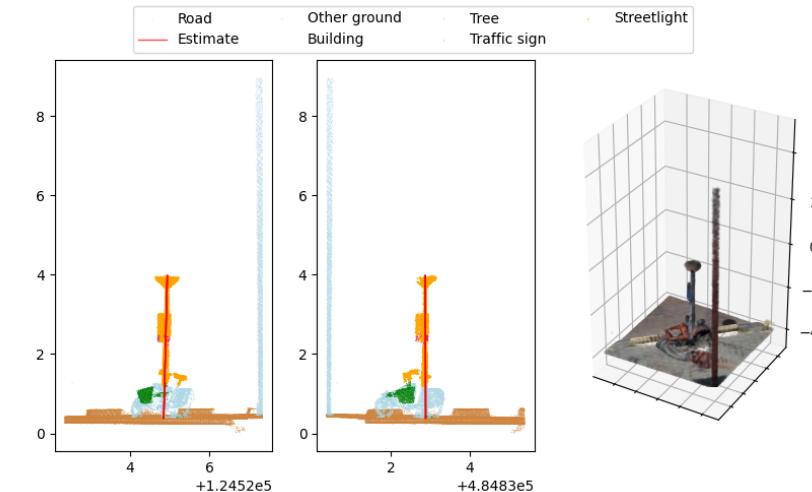
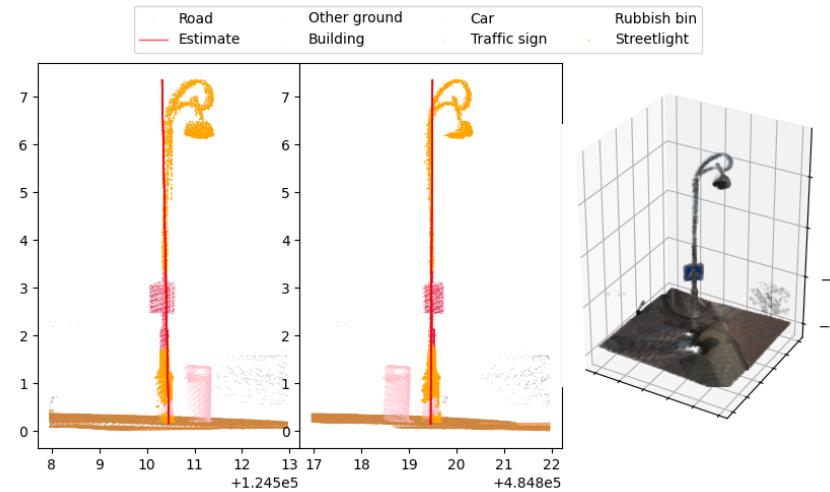
Sometimes, one street light is separated into two or more clusters, resulting in two or more images of the same pole that is fitted multiple times. If you notice this, you can regard one image as a true positive and the other(s) as false positives.



## Select one of the following options

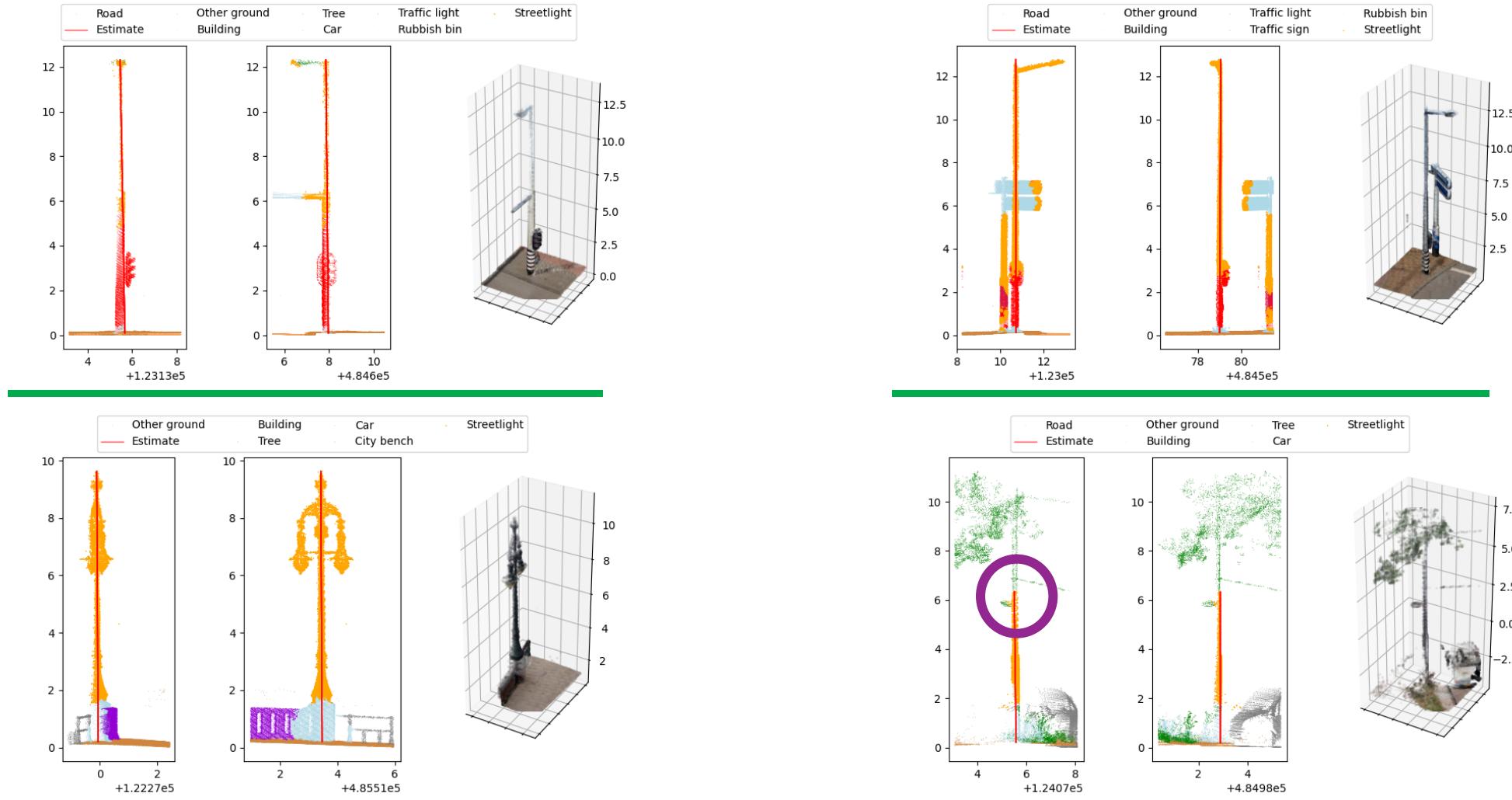
- Street light with good fit ([→], [enter], [space] or [1])
- Street light that needs a correction ([c] or [3])
- Not a street light ([f] or [2])
- Unclear example ([u] or [4])
- Previous example ([←] or [backspace])
- Exit the program ([escape])

# Examples: True Positives ([→], [enter], [space] or [1])



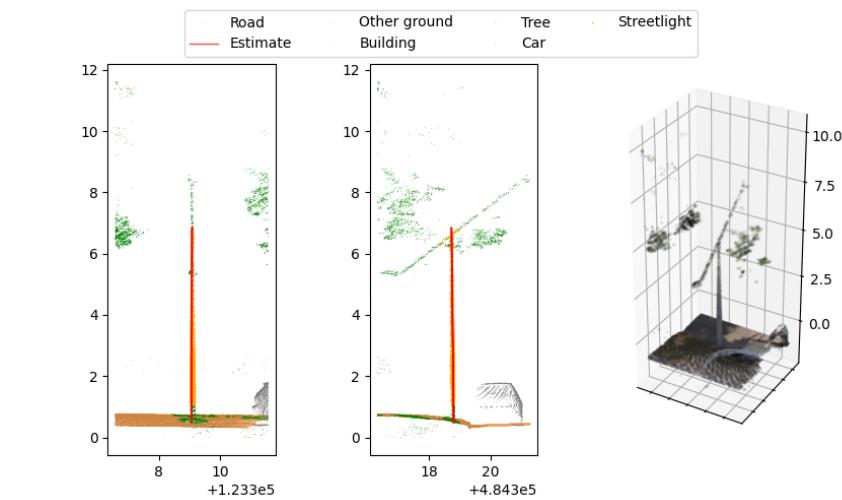
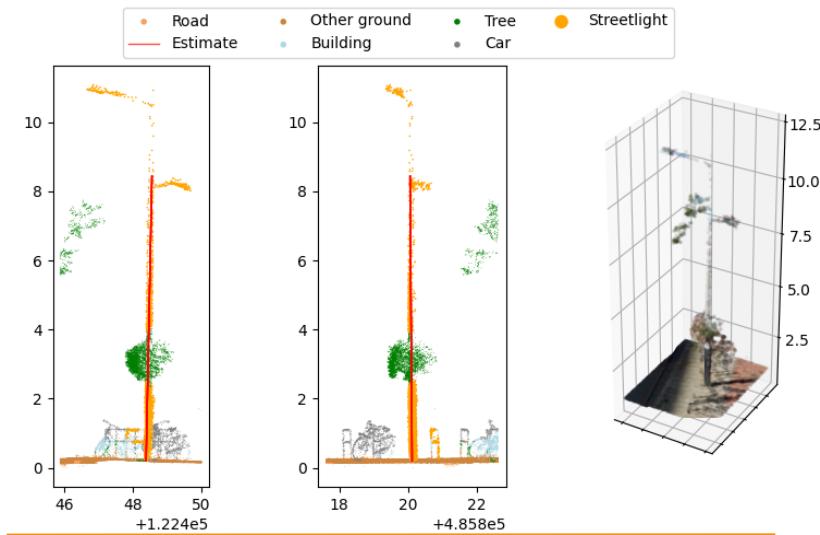
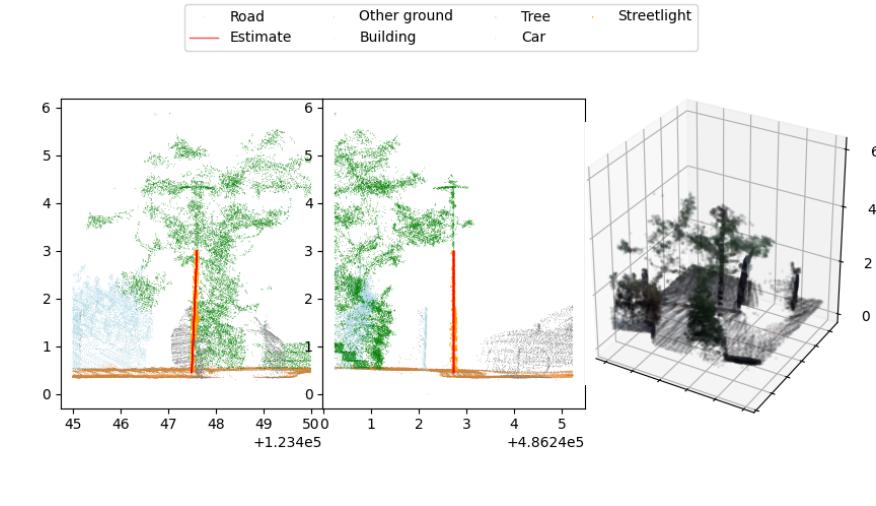
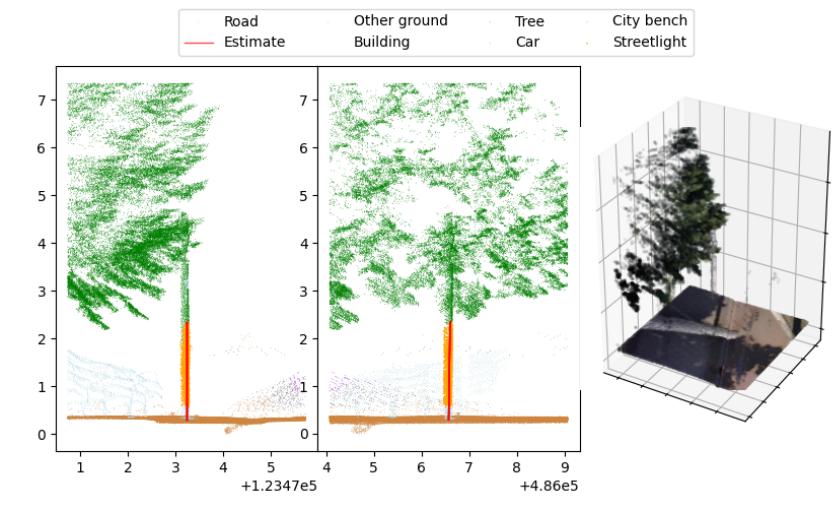
**Note:** The line does not go to the top, but the difference is small enough to determine the correct height category.

# Examples: True Positives ([→], [enter], [space] or [1])



**Note:** Small lamp as attachment, red line should roughly stop at the small lamp if there is no other lamp on the top

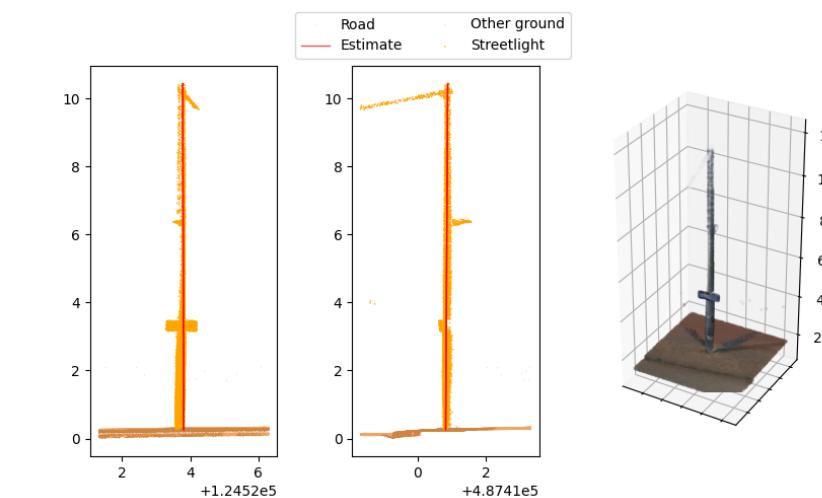
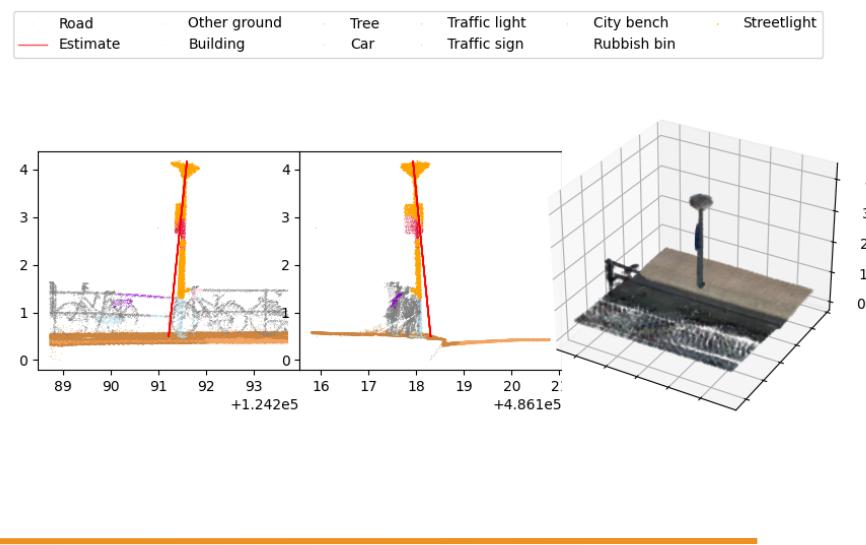
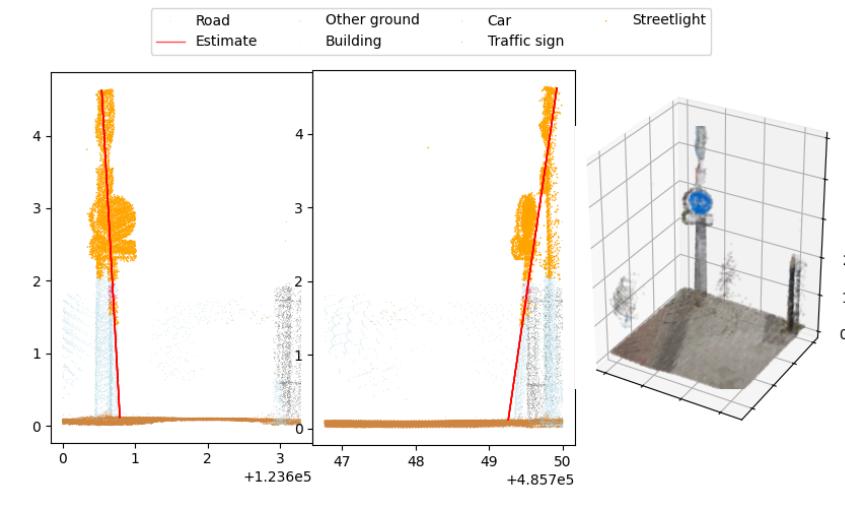
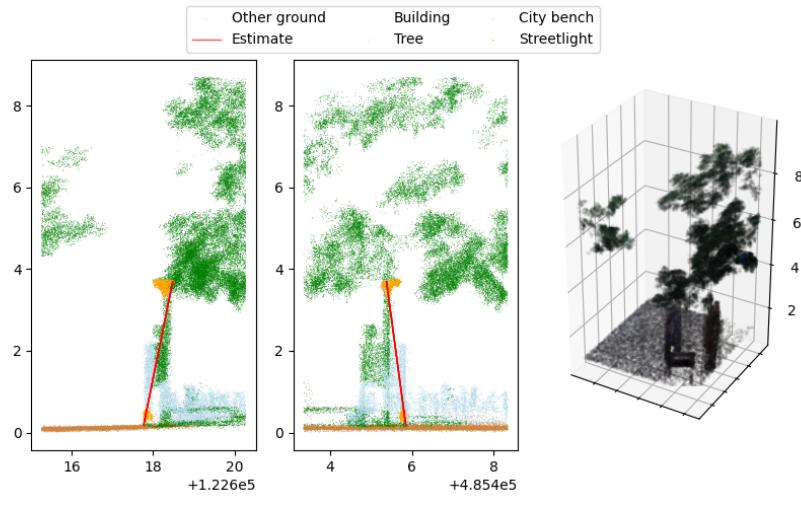
# Examples: True Positives that need a correction ([c] or [3])



**Note:** Pole with two attachments where the fitted line only reaches the first one

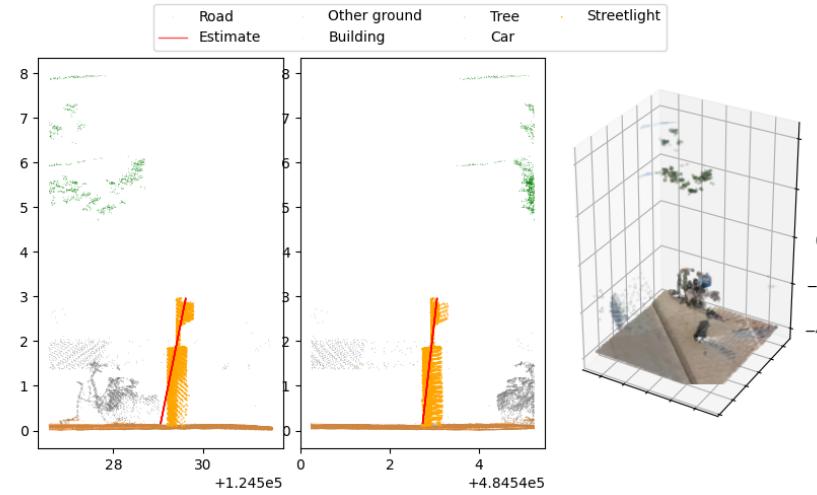
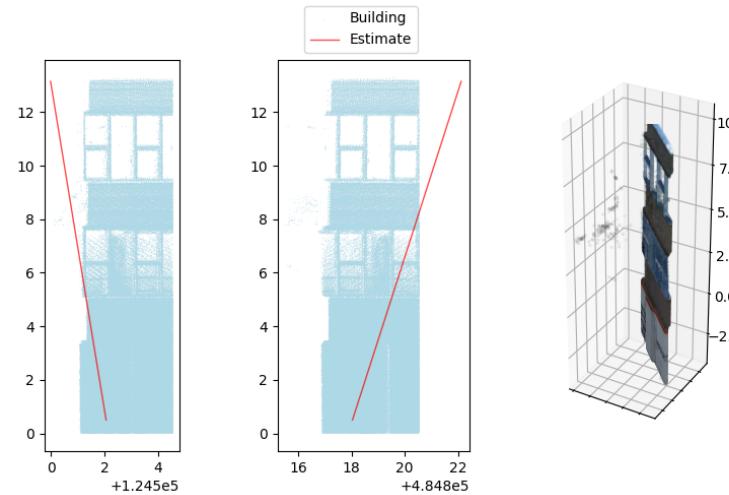
**Note:** The red line should go to the highest point of the object (including the diagonal attachment) - around 8.5m

## **Examples:** True Positives that need a correction ([c] or [3])

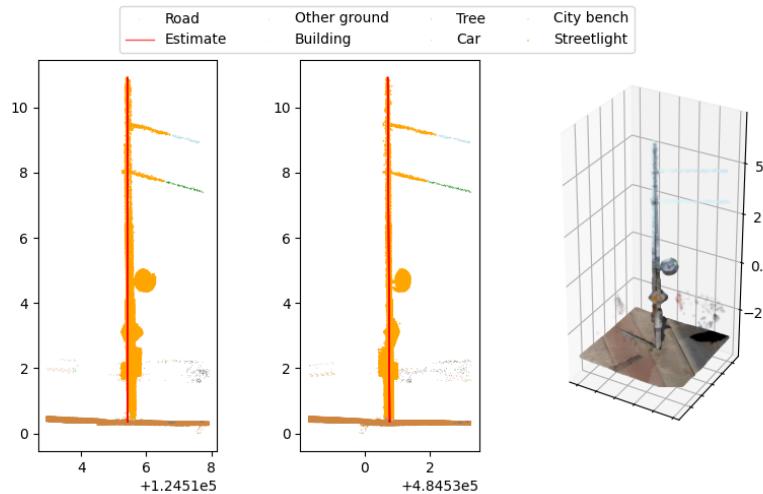


**Note:** Small lamp as attachment, red line should roughly stop at the small lamp if there is no lamp on the top

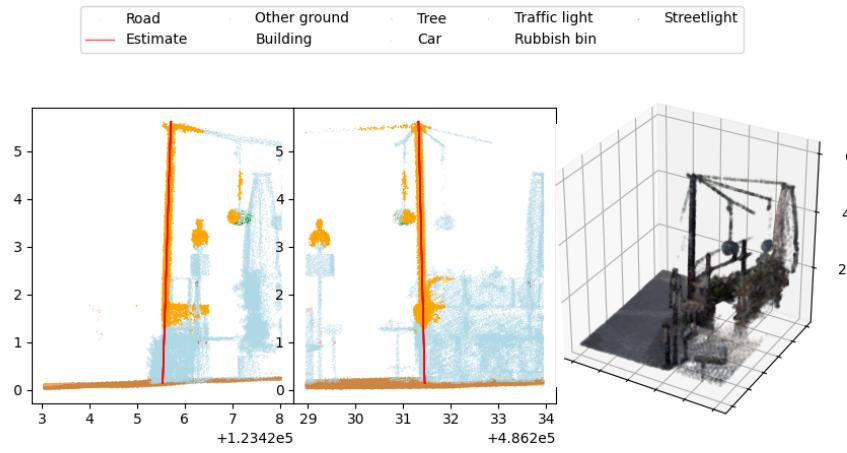
# Examples: False Positives ([f] or [2])



**Note:** Noise



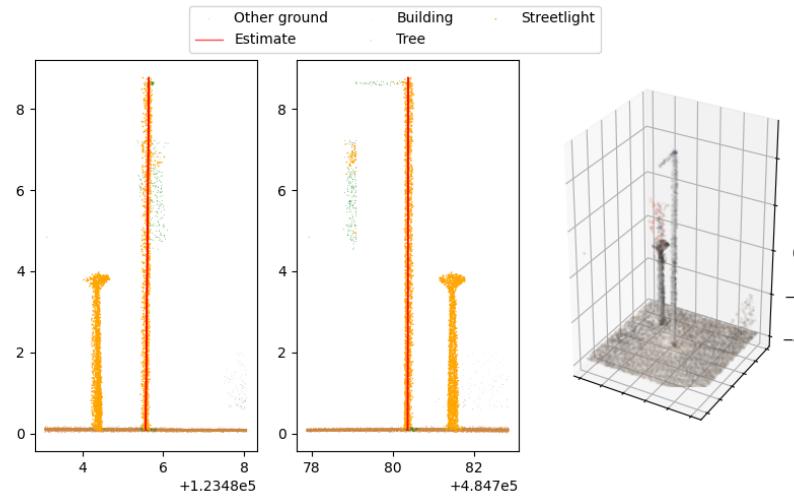
**Note:** A short pole-like object without a lamp attachment.



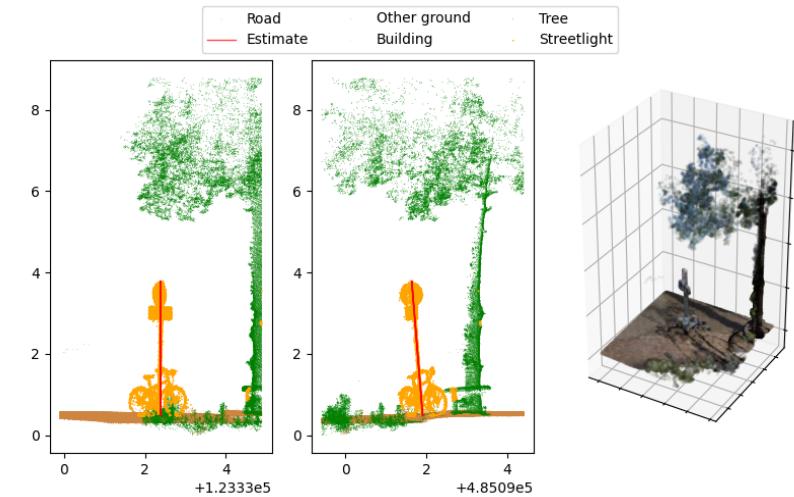
**Note:** No lamp as attachment

**Note:** No lamp as attachment

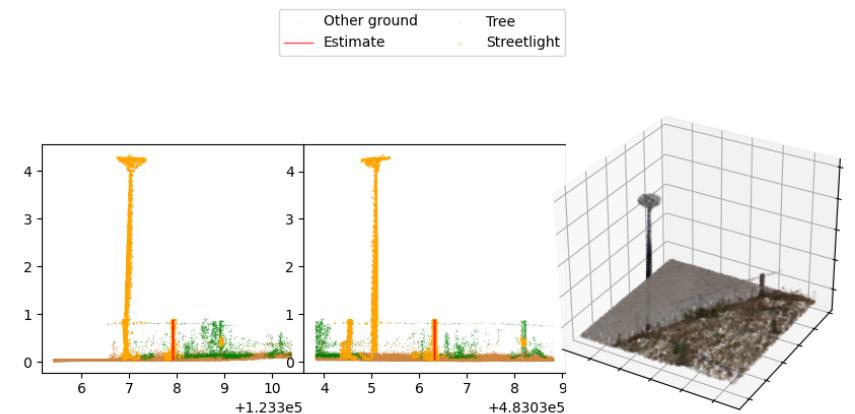
# Examples: False Positives ([f] or [2])



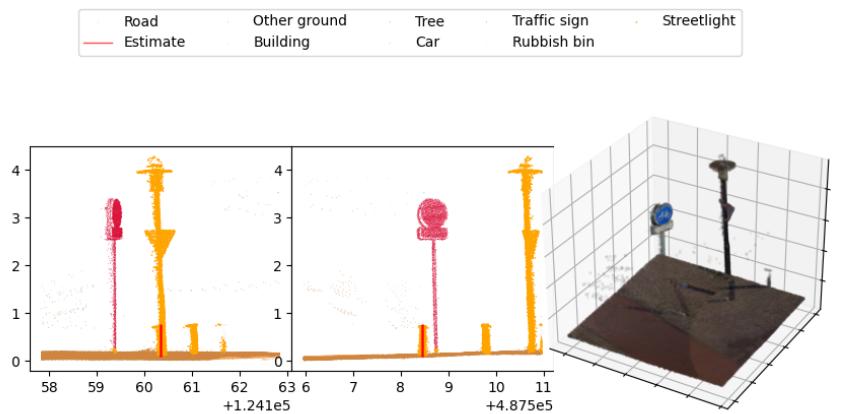
**Note:** No (clear) lamp as attachment. Also, there is another lamp to the side - they would never be so close



**Note:** Traffic sign

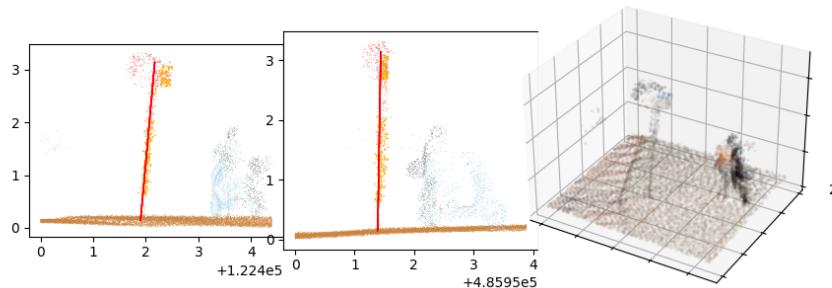
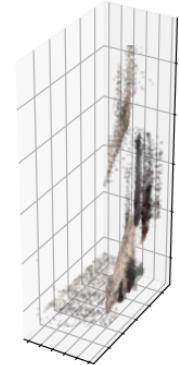
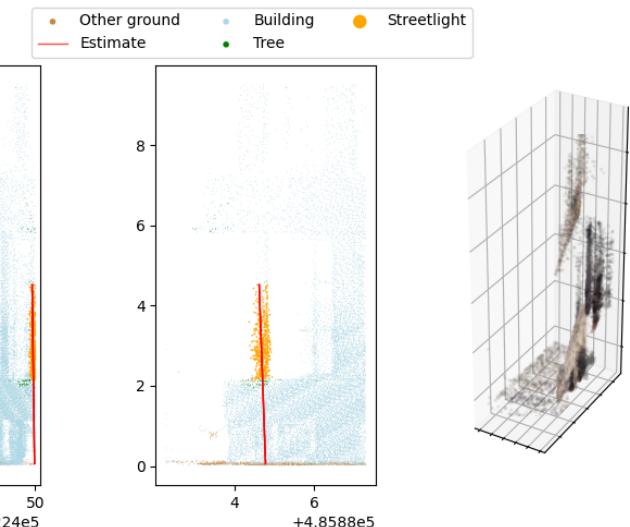
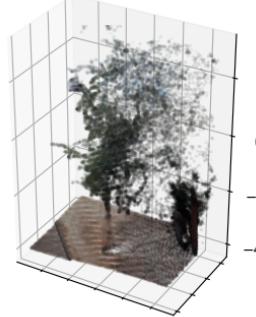
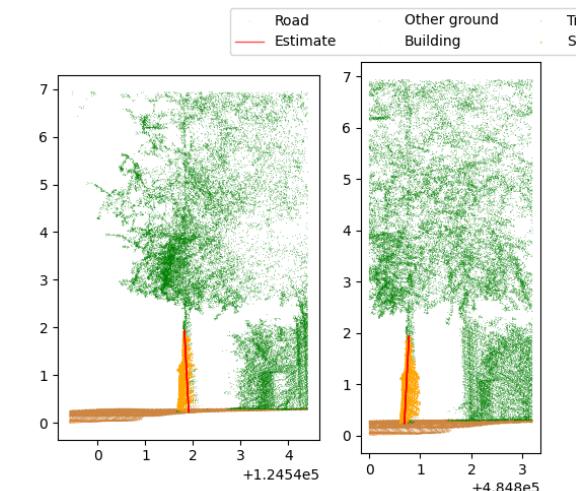


**Note:** Short pole-like object without a lamp attachment. Another lamp is located to the side



**Note:** Short pole-like object without a lamp attachment (clearly seen on the second image). Many other objects, including a lamp around it.

# Examples: Unclear Example ([u] or [4])



## Task 2: Adjust fits



### Run the program

by calling "`python3 pole_tagger.py --adjust_fit`"  
in a terminal which has the same directory  
as the `pole_tagger.py` file



### Occlusion?

If the view is obstructed or not all points from the light have been detected, click the point where the street light (either visibly or by approximation) hits the ground.



### Adjust the fit

By clicking first on the lowest point and then on the highest point (along the pole line in case of higher attachments). A fitted line will appear in red.



### Select one of the following options

- New fitting is correct ([enter] or [space])
- New fitting is still incorrect (any other key to redo the fitting)
- Exit the program ([escape])

# Task 3: Validate types



## Run the program

by calling "`python3 pole_tagger.py --validate_type`"  
in a terminal which has the same directory  
as the `pole_tagger.py` file



## Original vs Example

Two windows will pop up. On the left: the predicted type; on the right: the street light which needs a type validation.



## Select one of the following options

- **Correct predicted type** ([enter])
- **Go to next type** ([→])
- **Go to previous type** ([←])
- **Type is unknown or unclear** ([1] or [u])
  
- Go to previous street light example ([backspace])
- Exit the program ([escape])



## More likely types first

The types to iterate through are ranked by a trained regression model that utilizes the height, radius and color. Thereby, the tool aims to display the correct type as quickly as possible.



## Unsure?

Whenever in doubt, regard street lights as "unclear type" rather than assigning a possibly incorrect type. This way, we can inspect the object further.

# **Street Light Types To Distinguish**



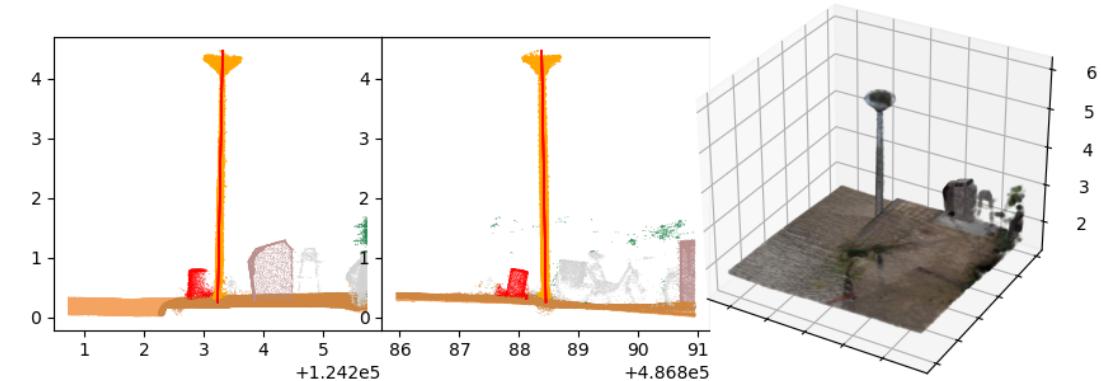
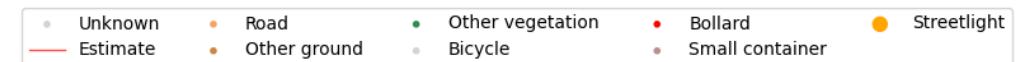
# Type 1

Around 4m tall  
Cone as lamp  
Two variants:

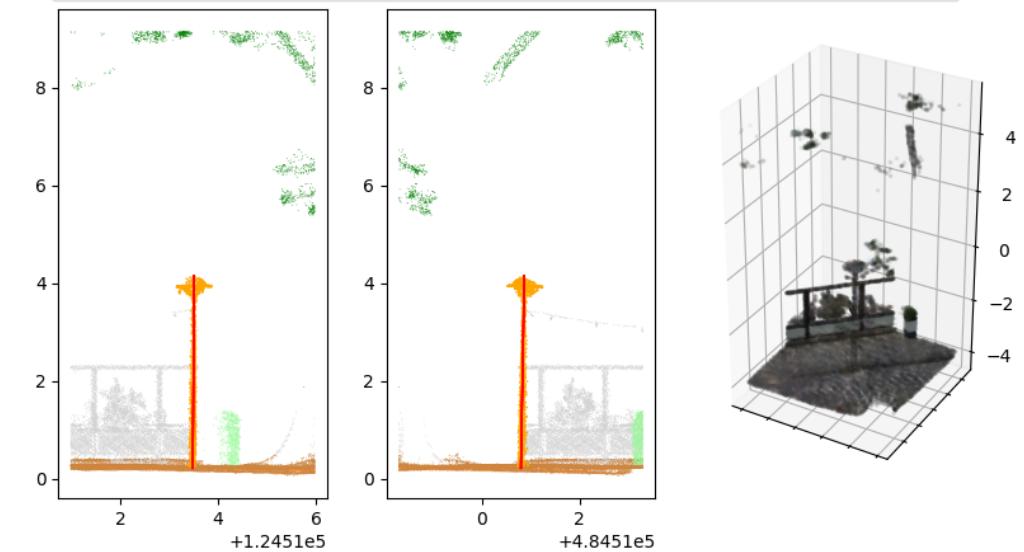
- Flat top
- Rounded top

Variant with rounded top similar to type 5

Flat top

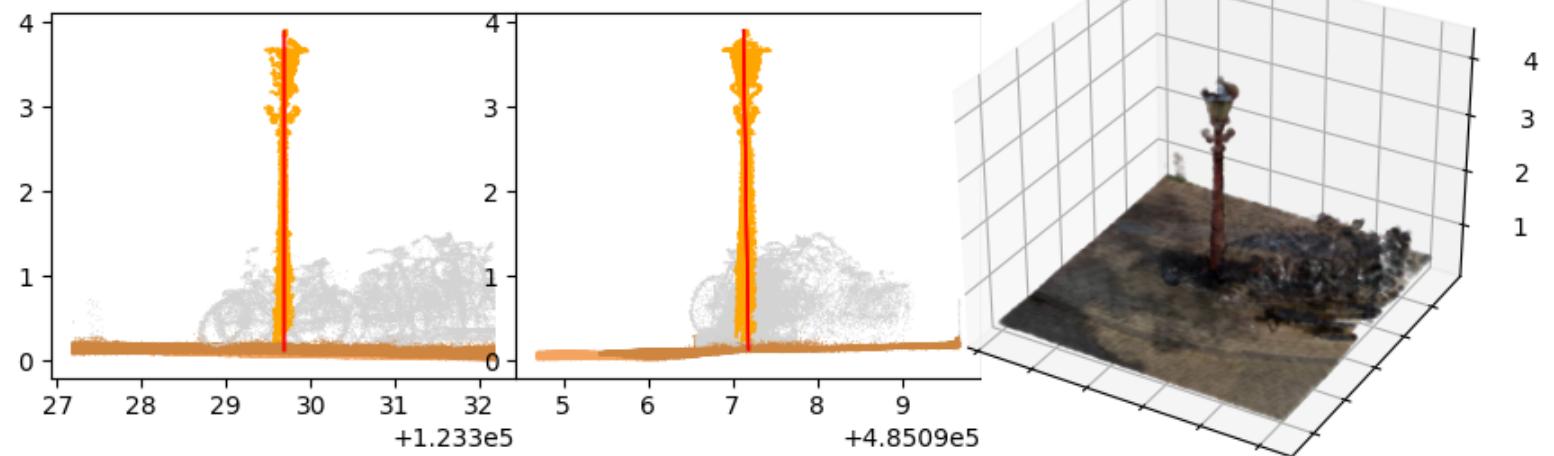
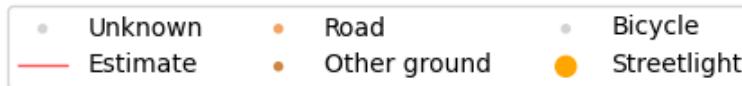


Rounded top



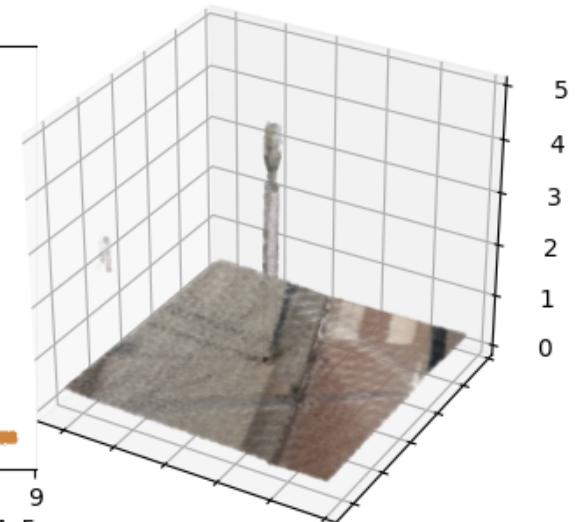
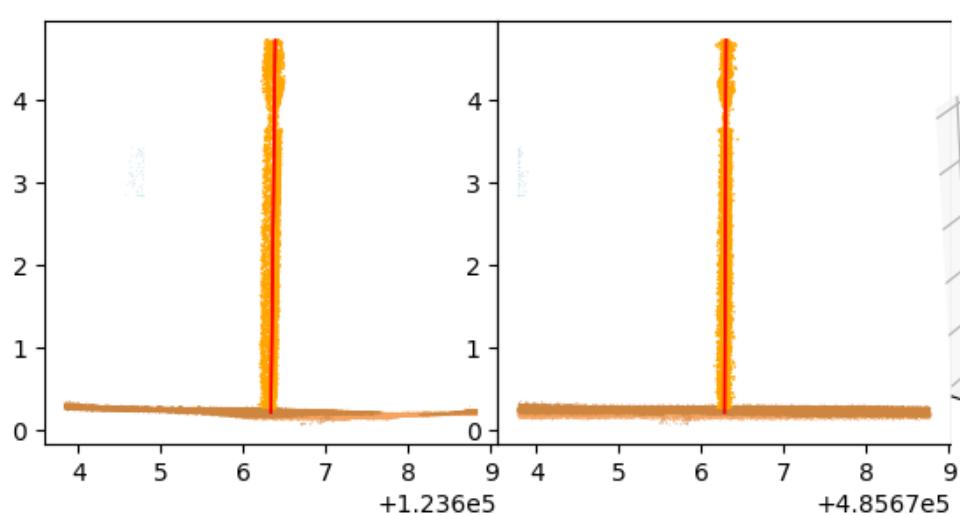
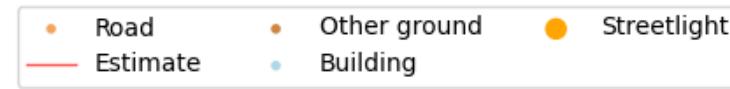
# Type 2

Around 4m tall  
Classical look  
Red coating



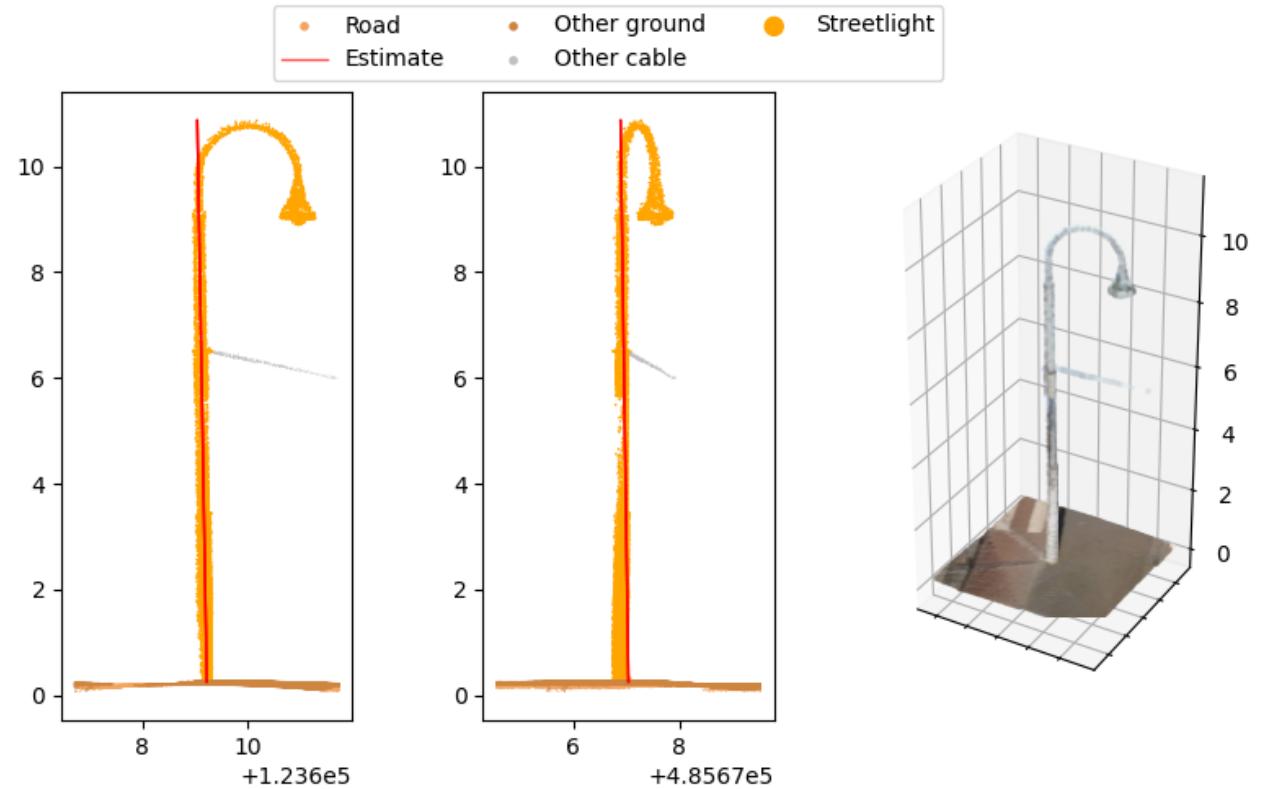
# Type 3

Around 4m tall  
Narrow pole  
Grey coating



# Type 4

Around 11m tall  
'Hanging' lamp  
Similar to type 13 but bigger

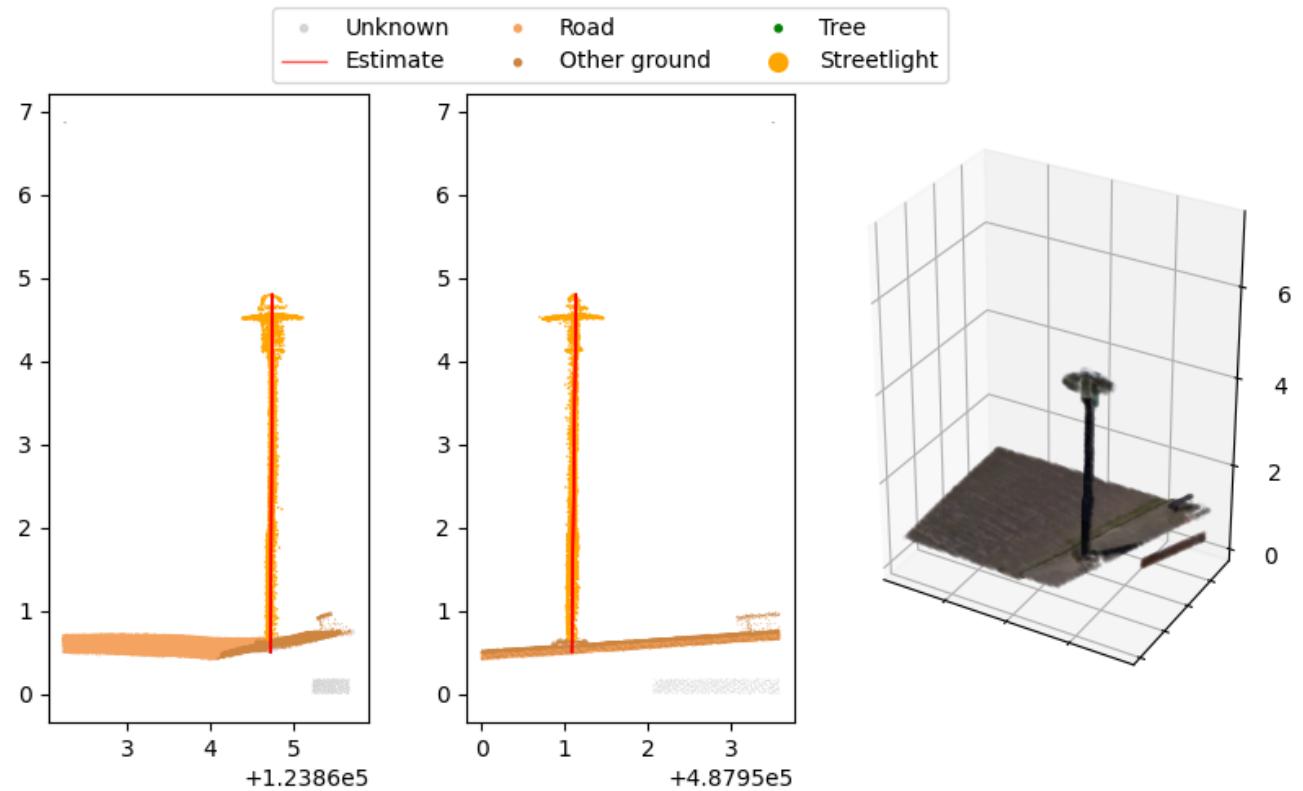


# Type 5

Around 4m tall

Hat-shaped lamp

Similar to type 1 (with rounded top)

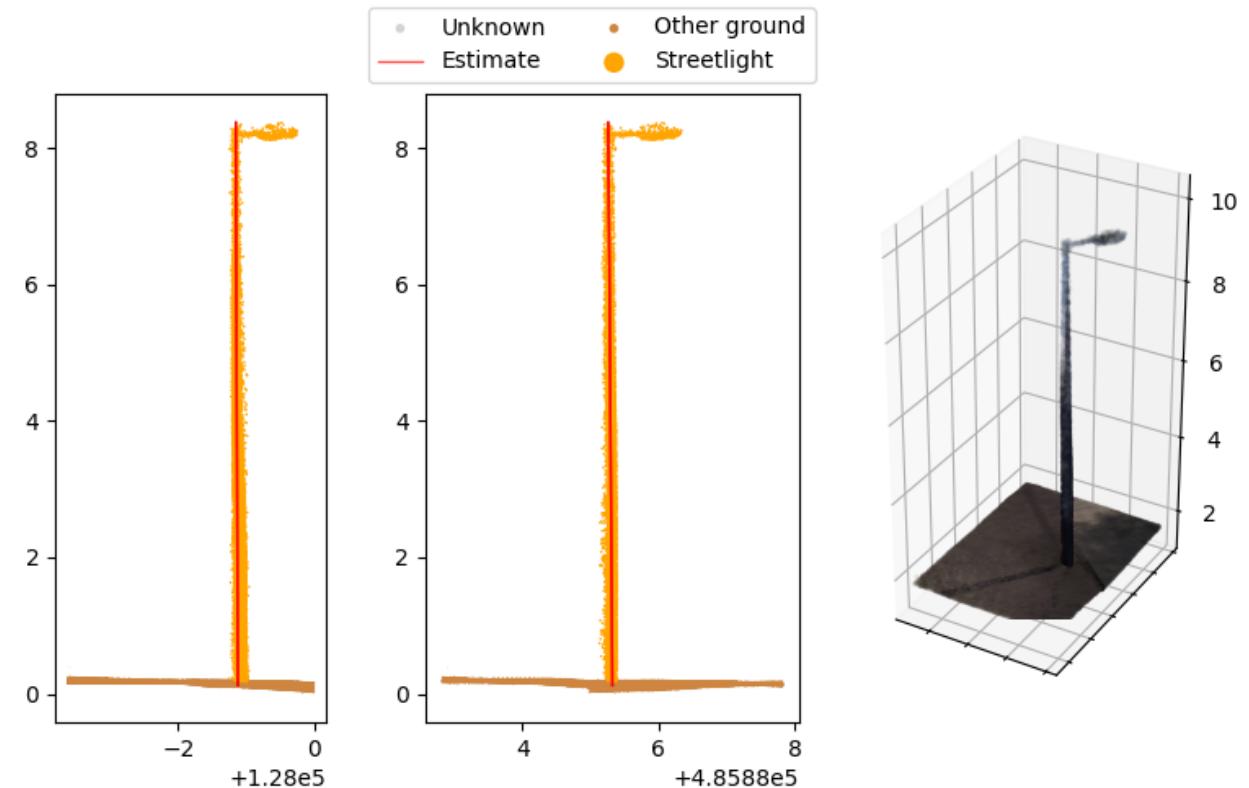


# Type 6

Around 8m tall

Similar to types 7 and 8 but different length

Usually black or grey

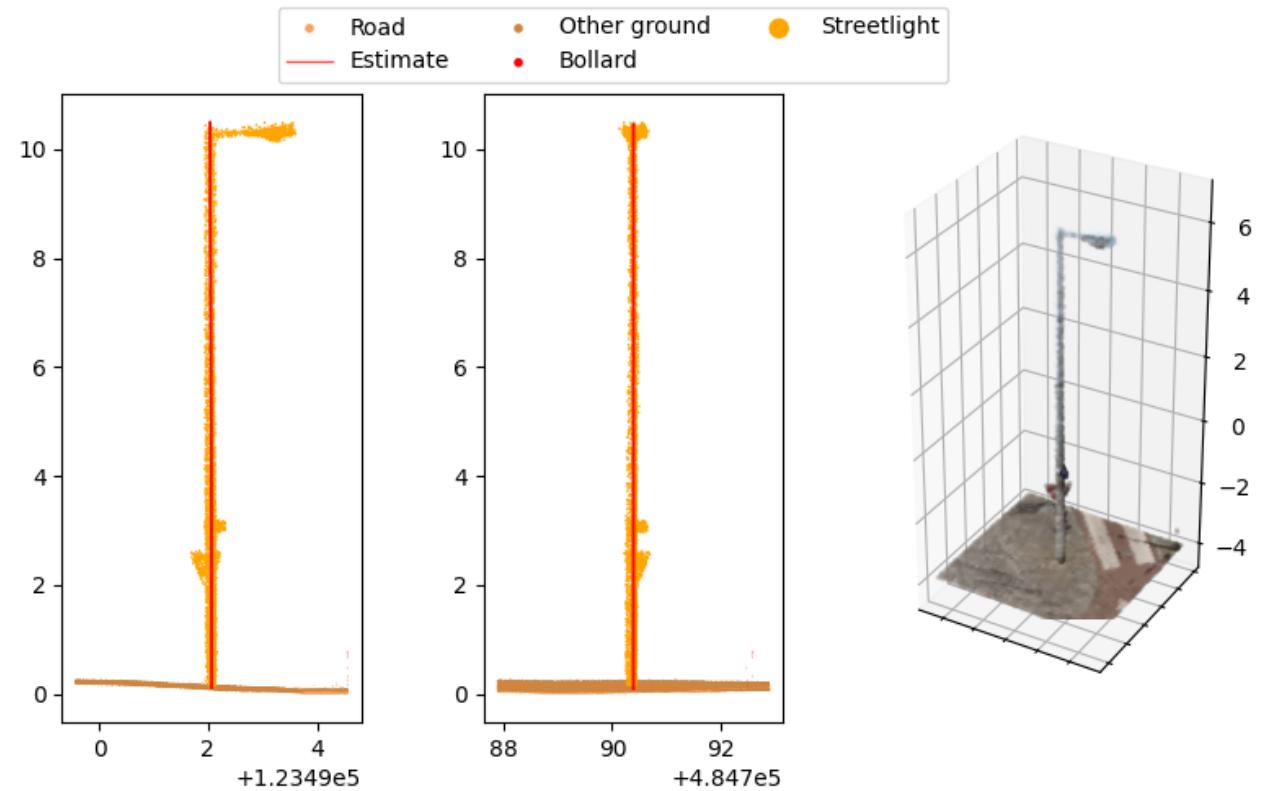


# Type 7

Around 10m tall

Similar to types 6 and 8 but different length

Usually grey

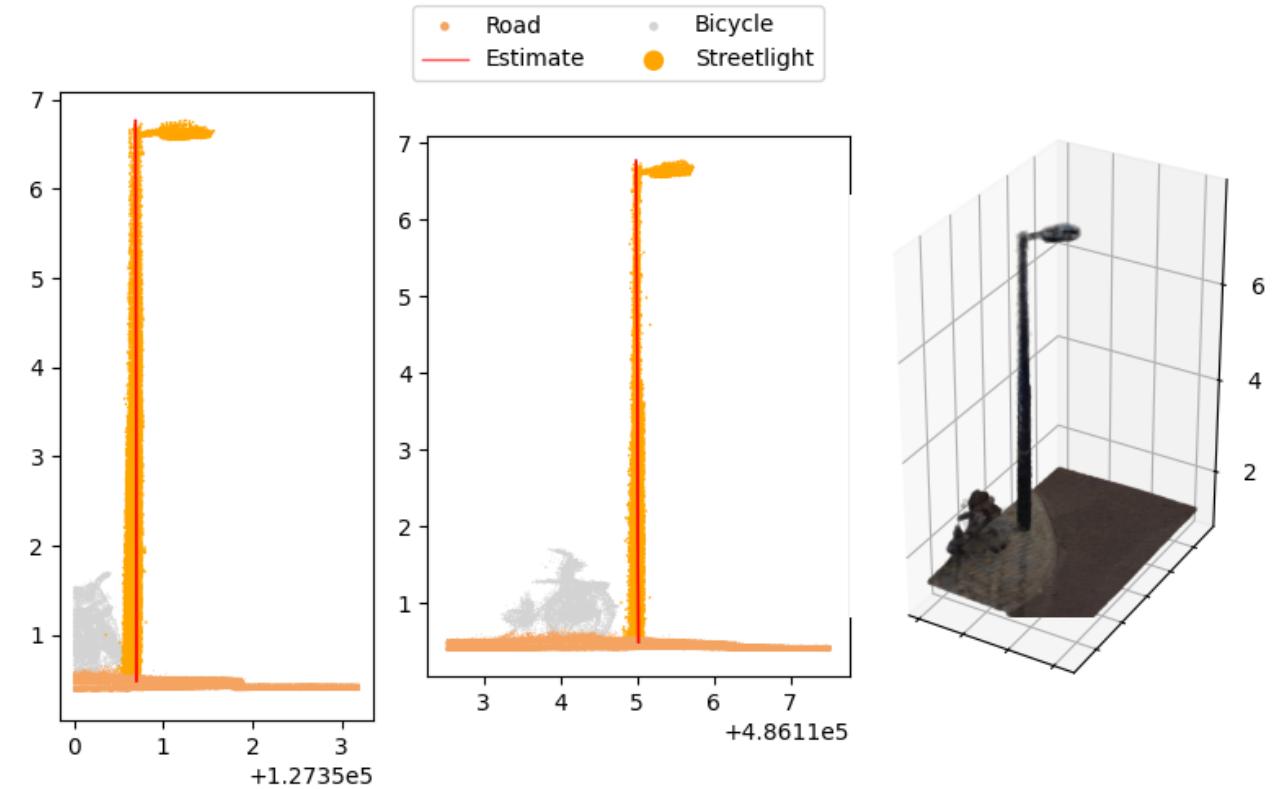


# Type 8

Around 6m tall

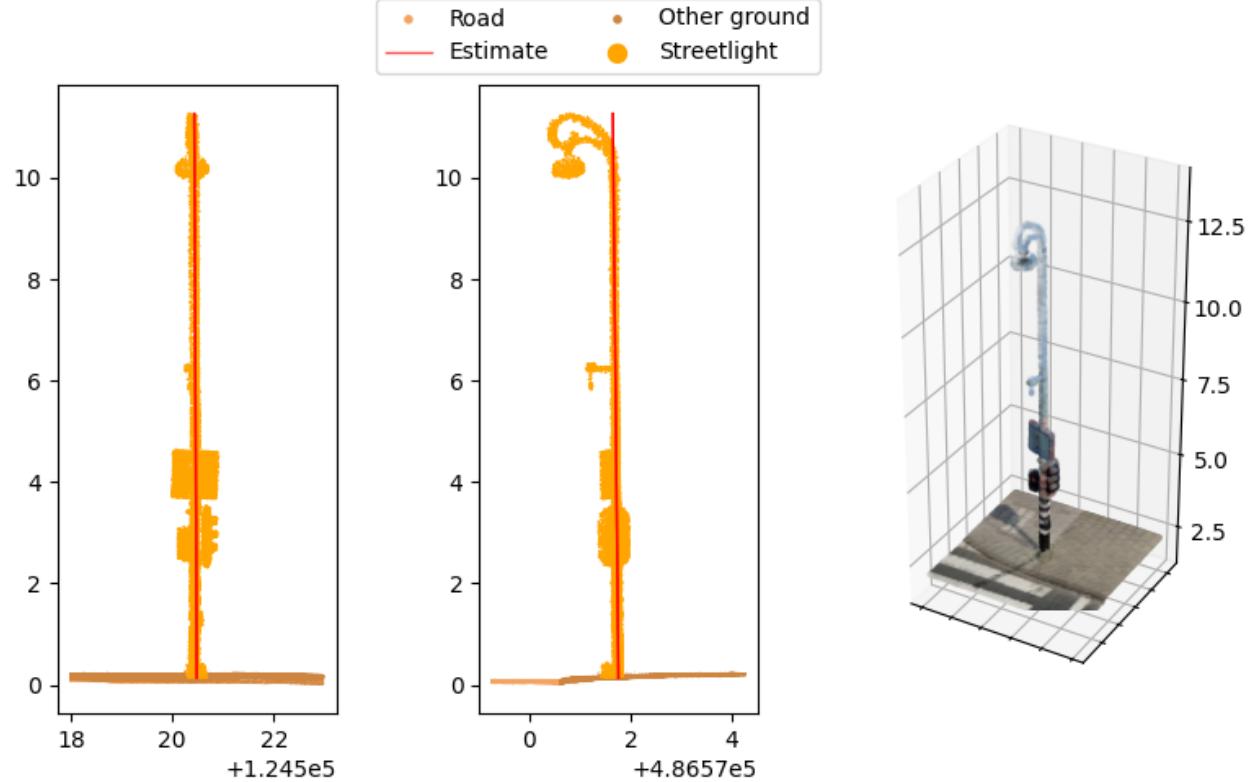
Similar to types 6 and 7 but different length

Usually black



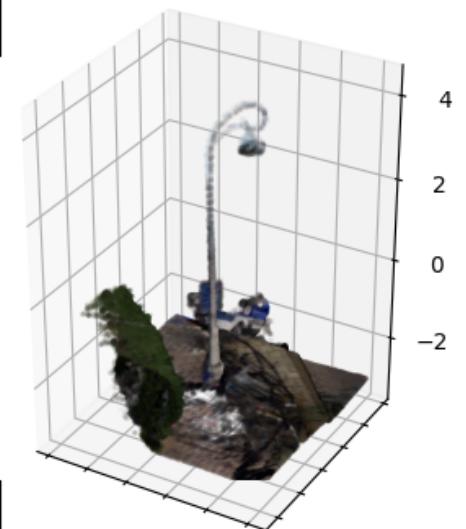
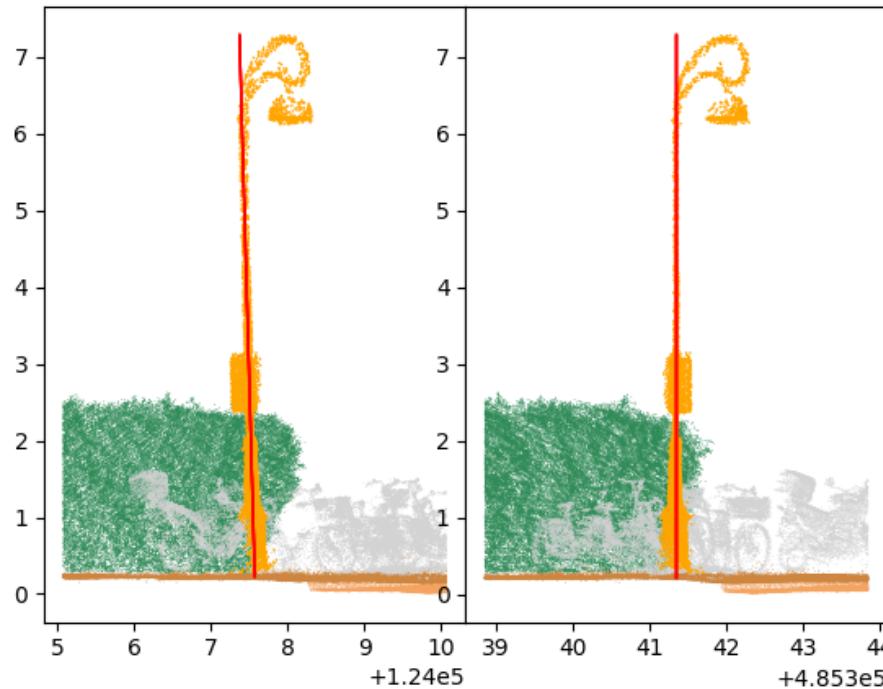
# Type 9

Around 11m tall  
Same as type 10 but taller  
Grey  
Classical look



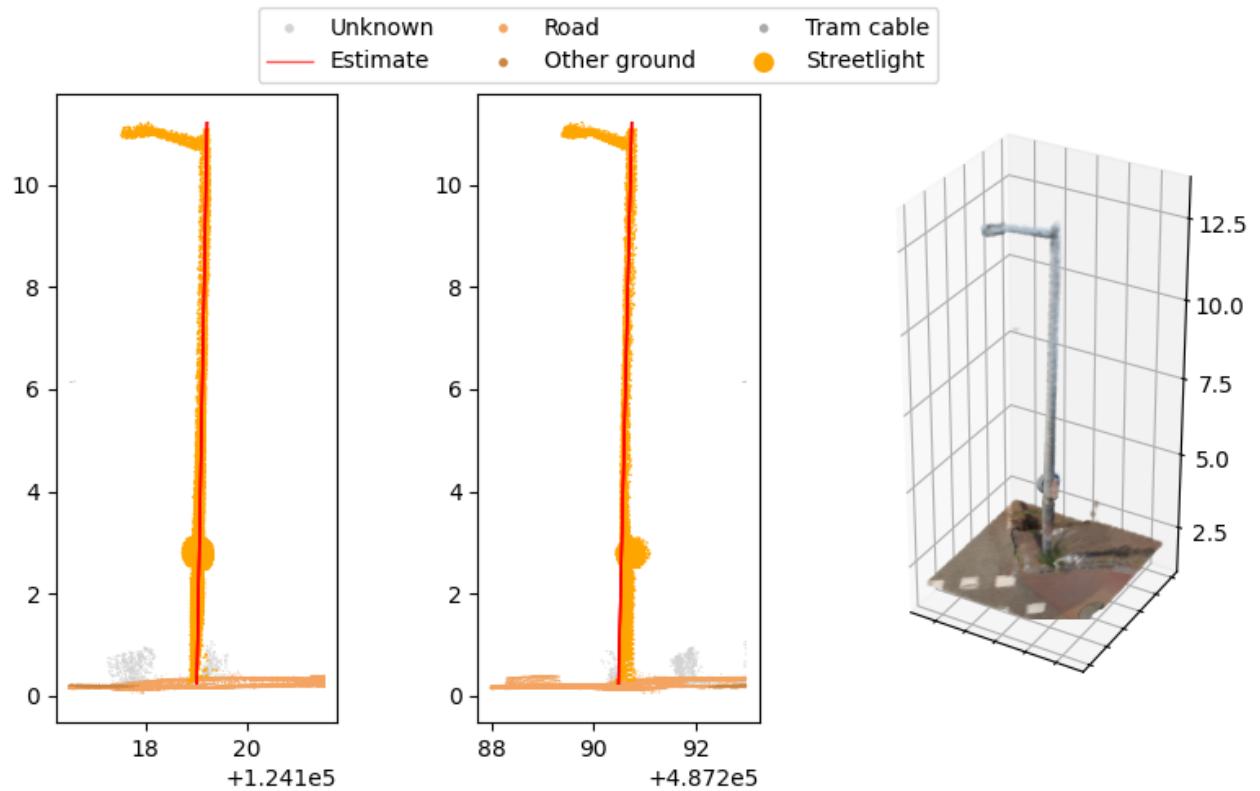
# Type 10

Around 7m tall  
Same as type 9 but shorter  
Grey  
Classical look



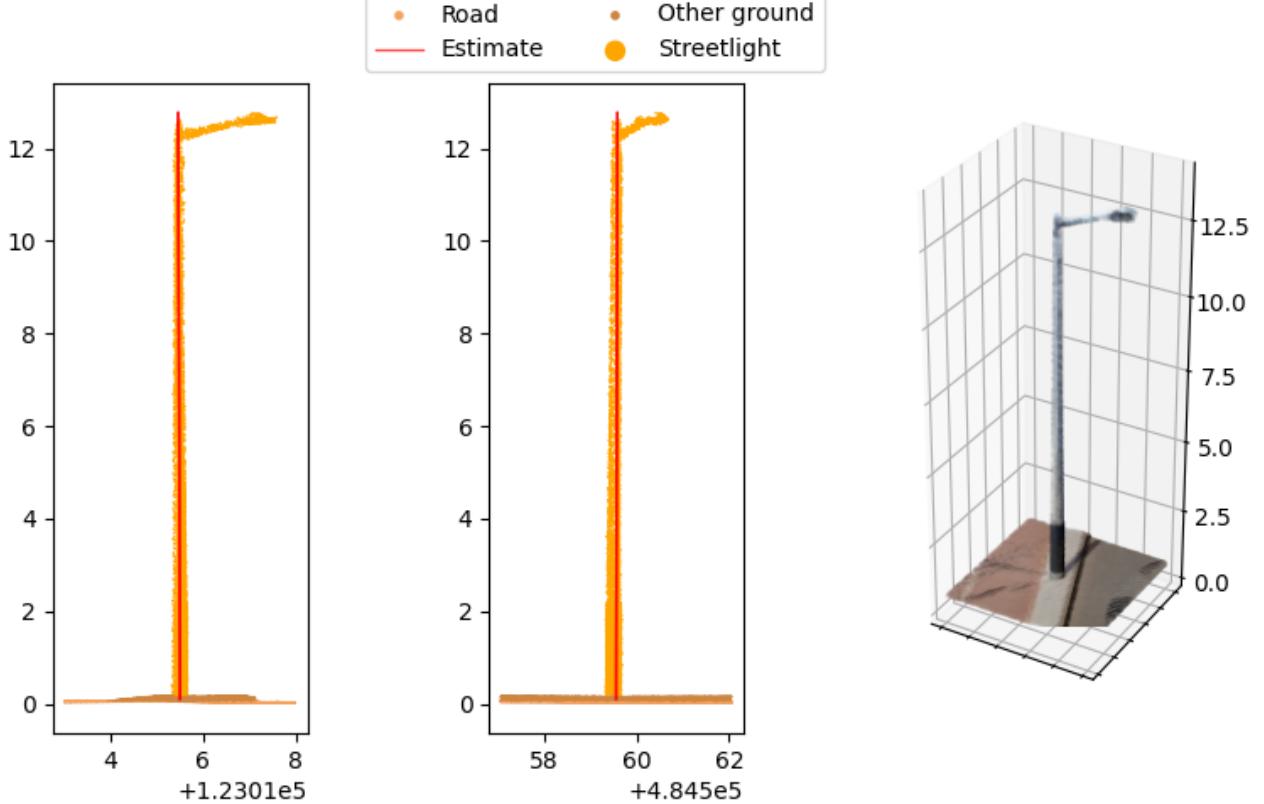
# Type 11

Around 11m tall  
Same as type 12 but shorter



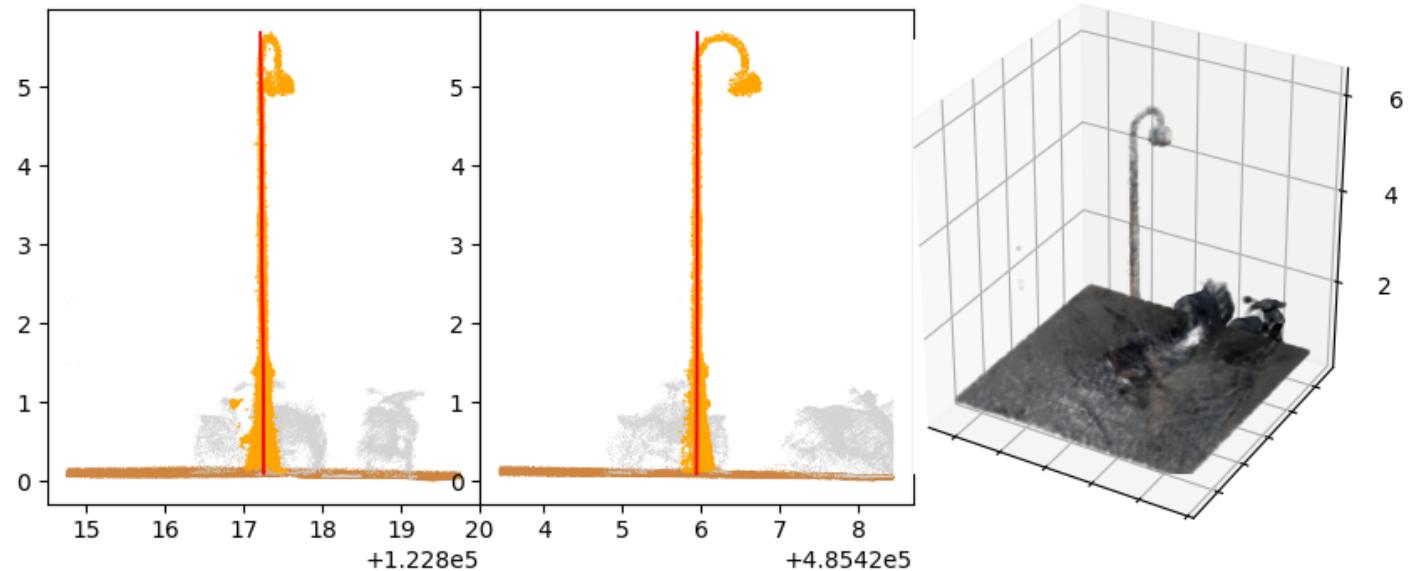
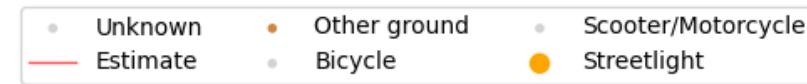
# Type 12

Around 13m tall  
Same as type 11 but taller



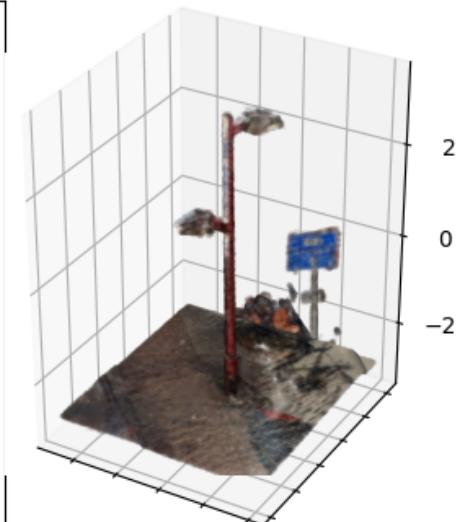
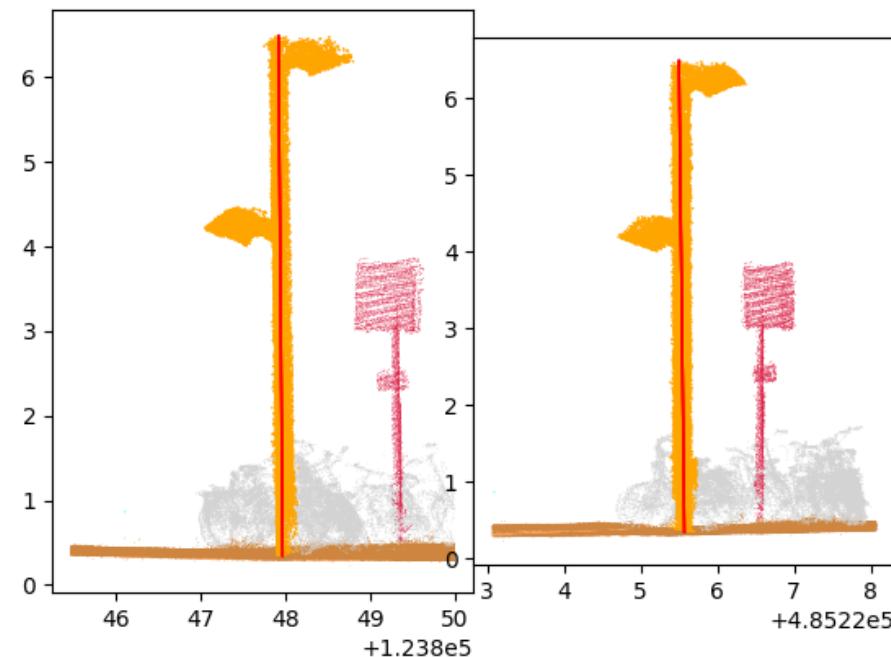
# Type 13

Around 6m tall  
'Hanging' lamp  
Similar to type 4 but bigger



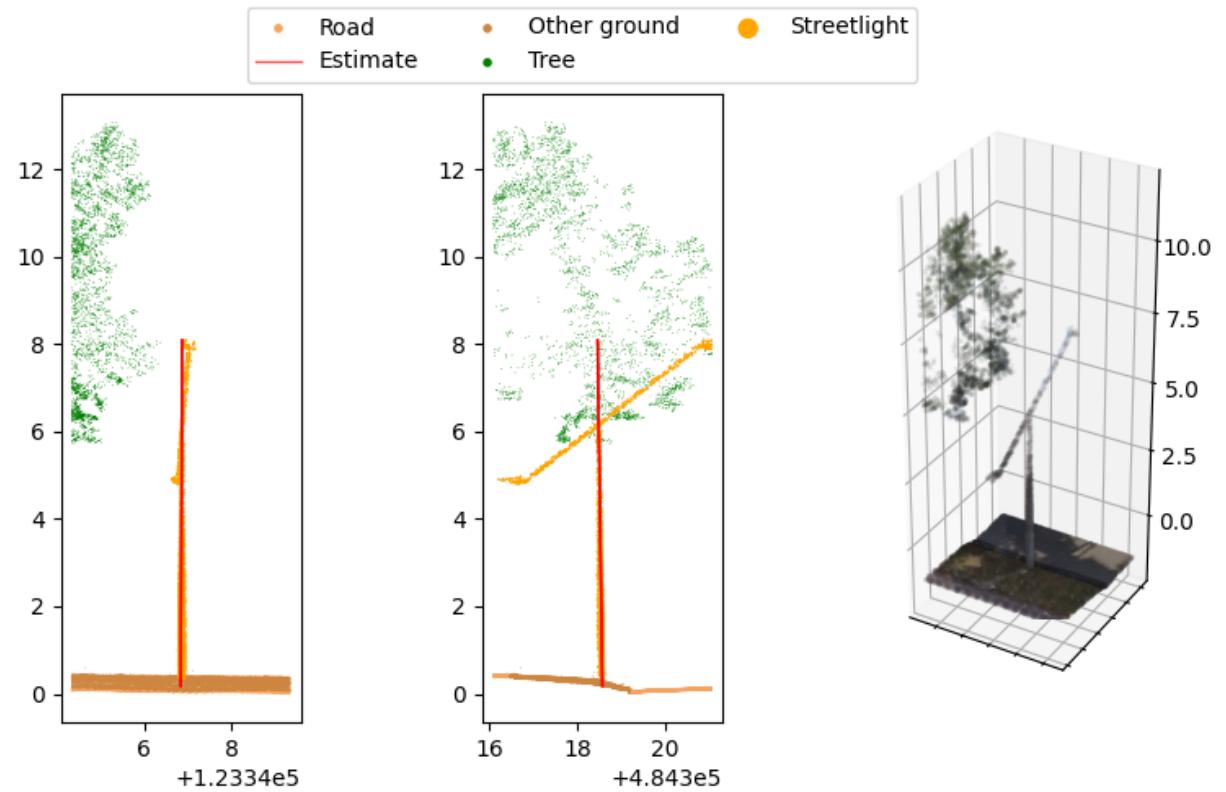
# Type 14

Around 6m tall  
Two lamps  
Unique shape



# Type 15

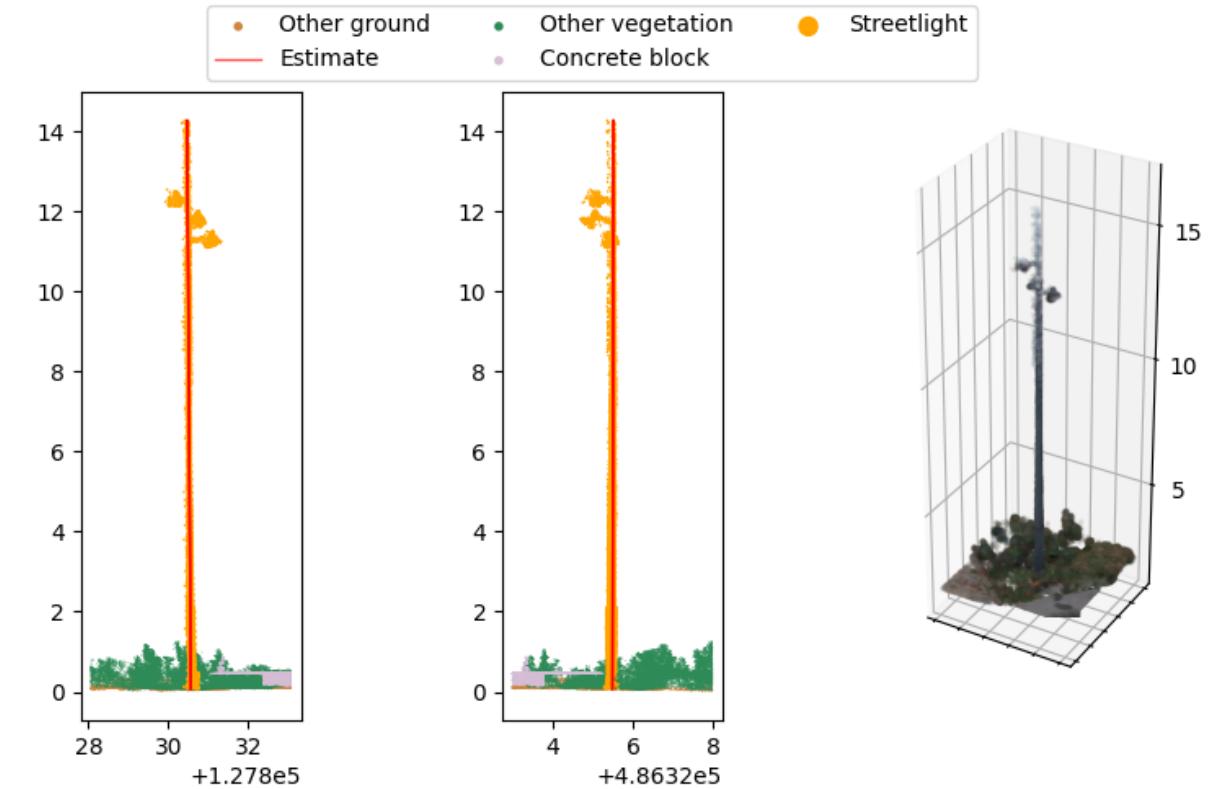
Around 8m tall  
Two lamps  
Unique shape



# Type 16

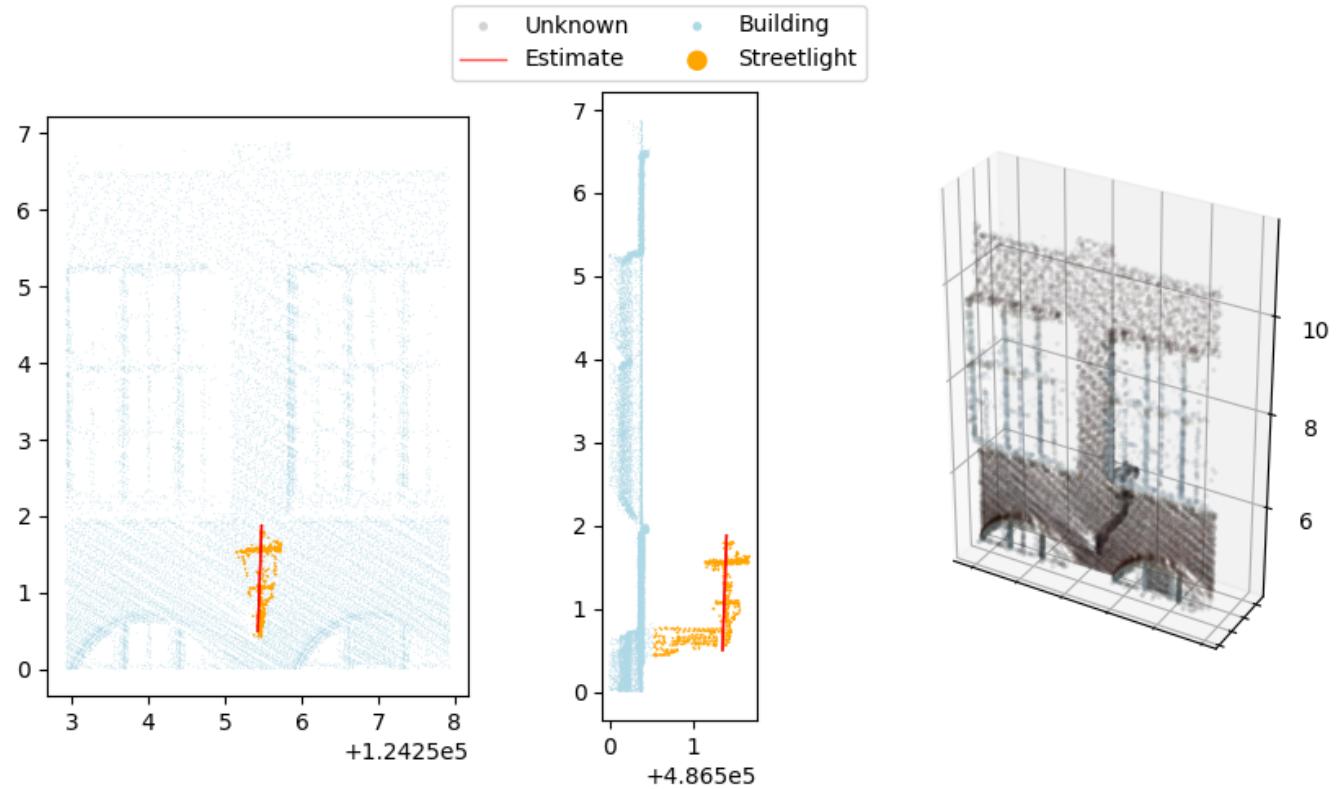
Between 10m/15m

Multiple lamps attached at almost the top



# Type 17

Wall attachment  
Can have different shapes

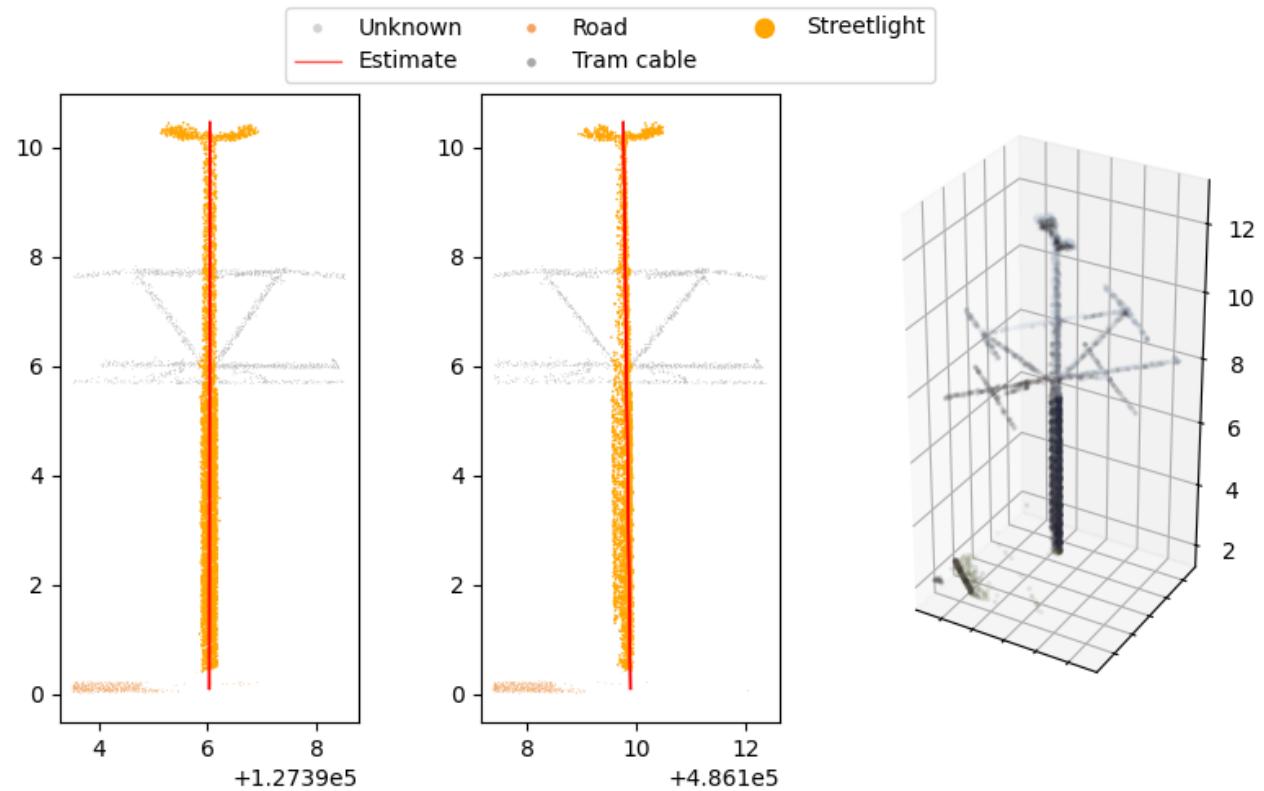


# Type 18

Around 10m tall

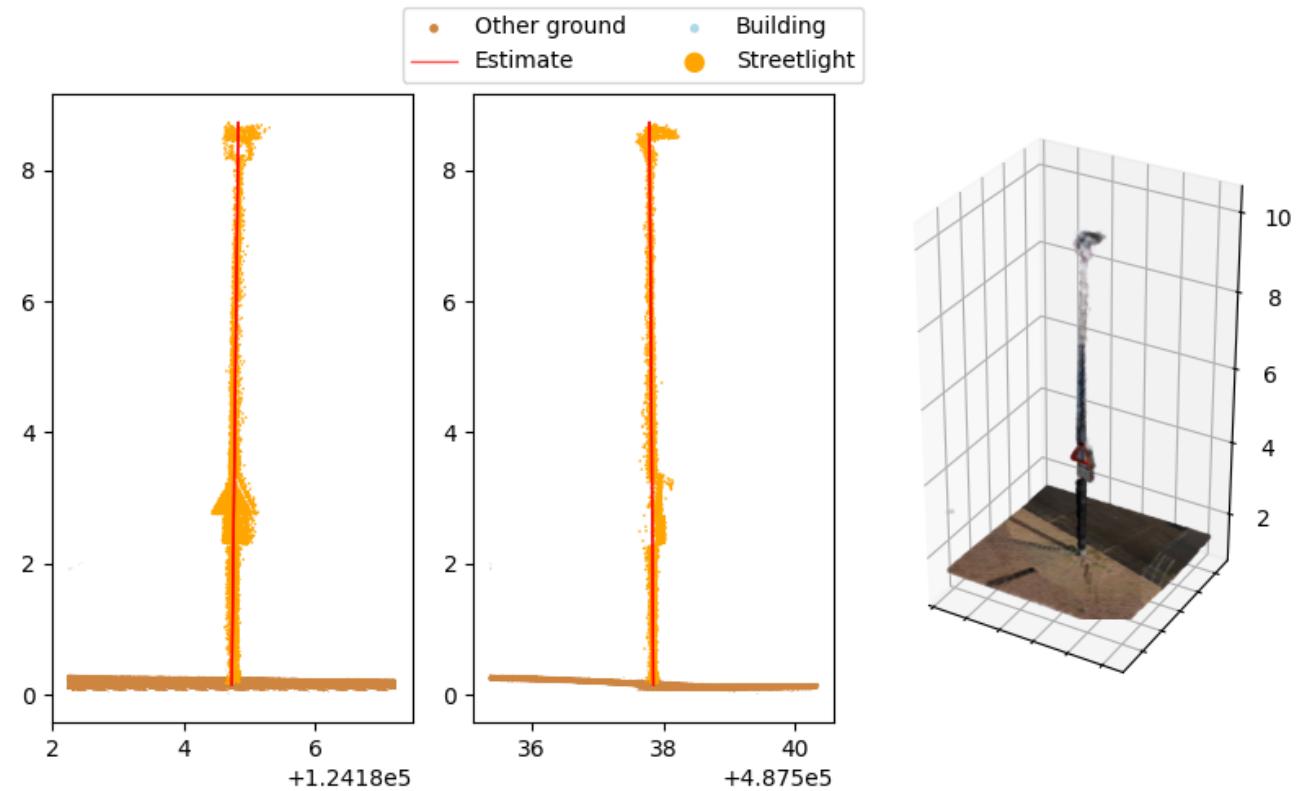
Usually 2 or 3 lamps

Usually grey and black



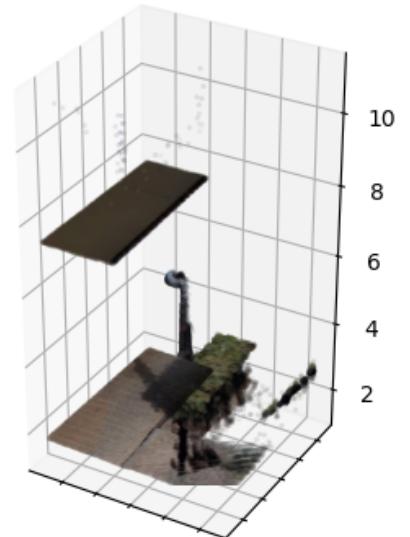
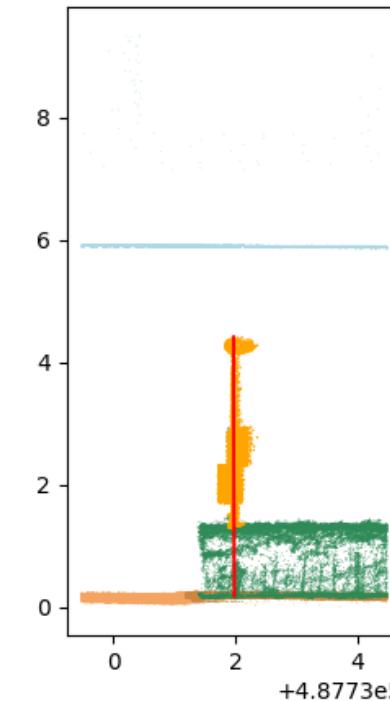
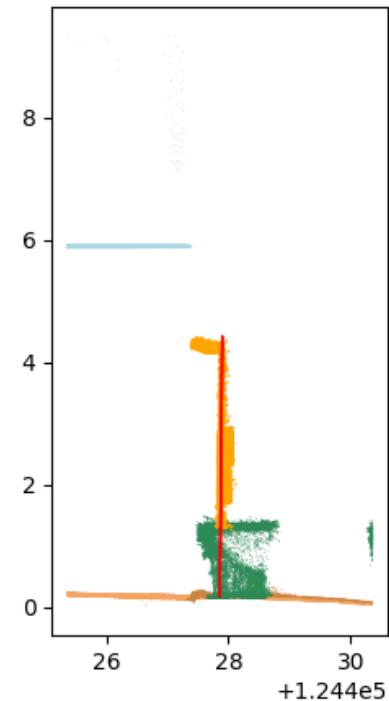
# Type 19

Around 8m tall  
Short lamp



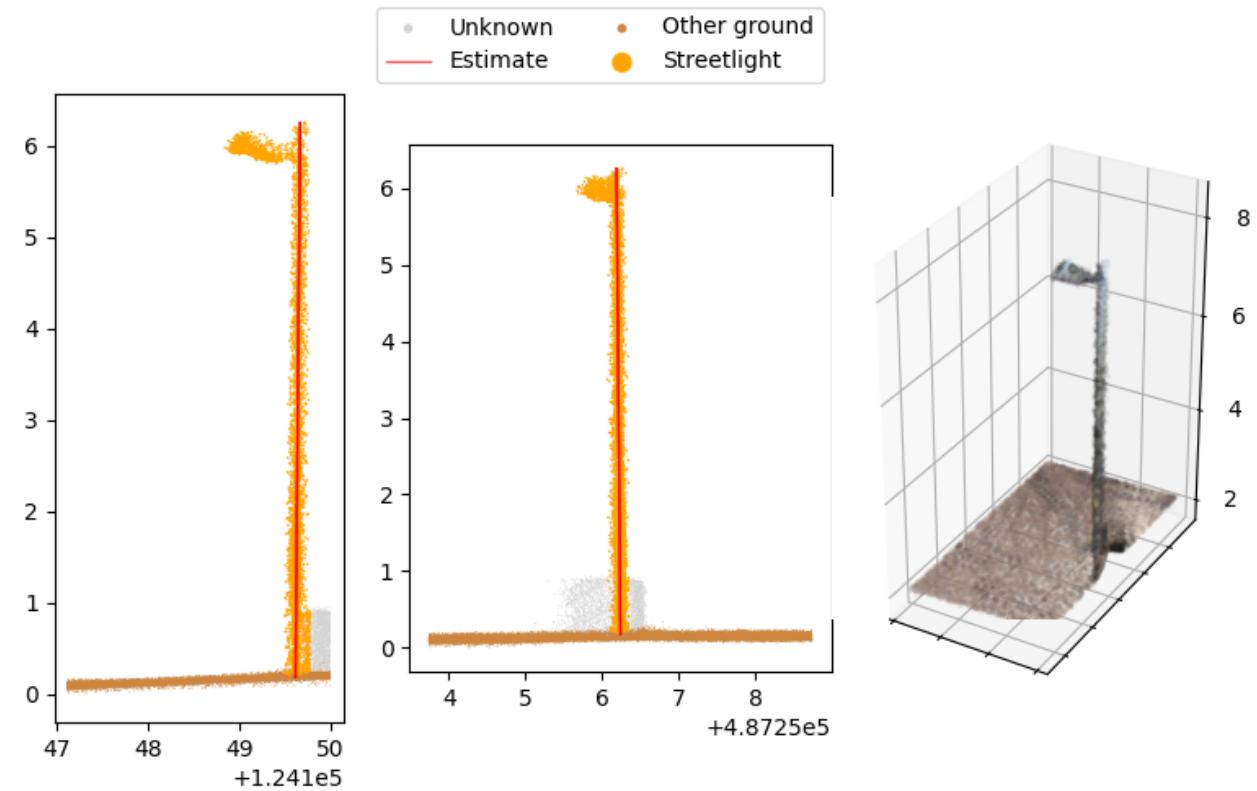
# Type 20

Around 4m tall  
Short lamp  
Usually black



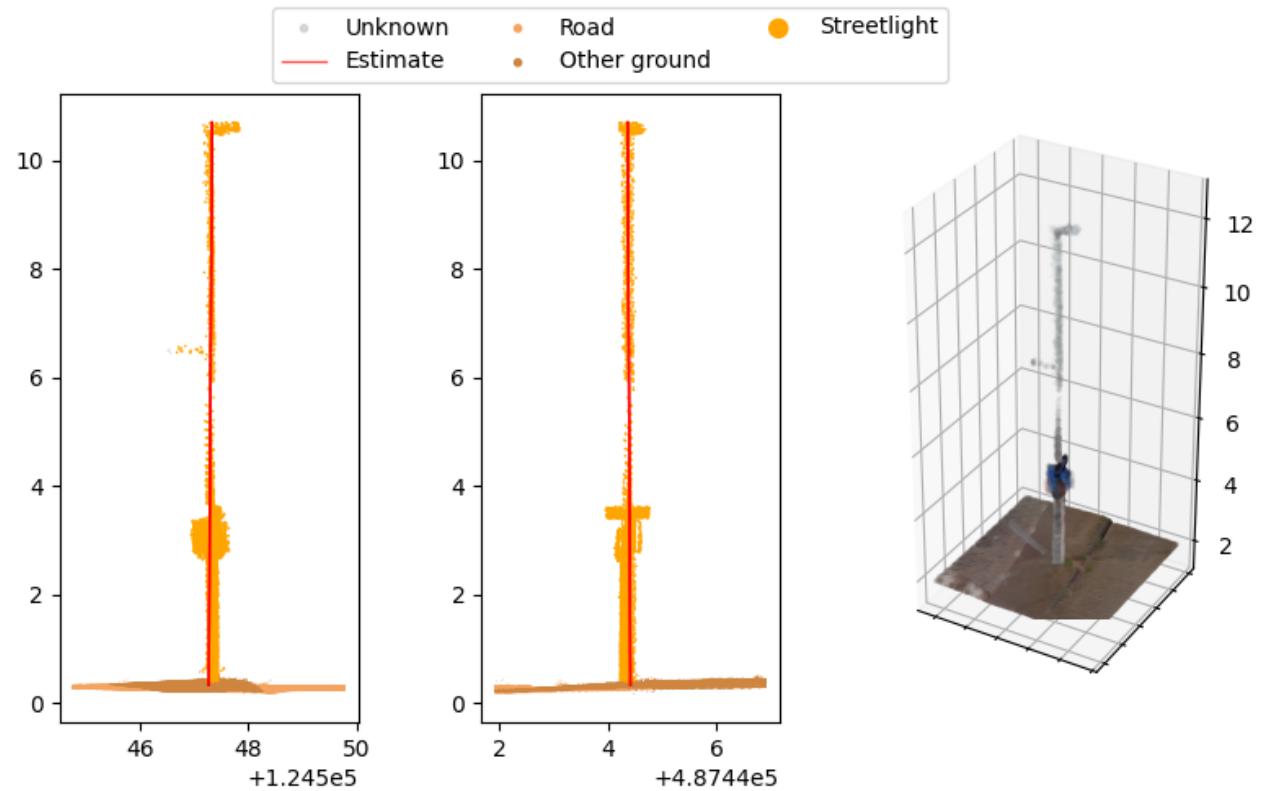
# Type 21

Around 6m tall  
Short lamp  
Usually grey



# Type 22

Around 10m tall  
Short lamp  
Usually grey



# Let's label poles together!

Email:

[aibalie@amsterdam.nl](mailto:aibalie@amsterdam.nl)

[amsterdamintelligence.com](http://amsterdamintelligence.com)

