Assignment-2, GEO-1002

Group Number 13: Ming-Chieh Hu, Dimitrios Lioumis, Neelabh Singh

October 2024

Assignment Group Members

In alphabetical order by surname:

SURNAME 1: Hu
NAME 1: Ming-Chieh
STUDENT ID 1: 6186416
STUDY PROGRAMME 1: Geomatics

SURNAME 2: Lioumis
NAME 2: Dimitrios
STUDENT ID 2: 6213944
STUDY PROGRAMME 2: Geomatics

SURNAME 3: Singh
NAME 3: Neelabh
STUDENT ID 3: 6052045
STUDY PROGRAMME 3: Geomatics

Fraud And Plagiarism Awareness

We confirm herewith that we have read and we are aware of the TU Delft rules (and consequences) regarding fraud and plagiarism, as written in https://www.tudelft.nl/en/student/my-study-me/rules-guidelines-and-participation/fraud-plagiarism

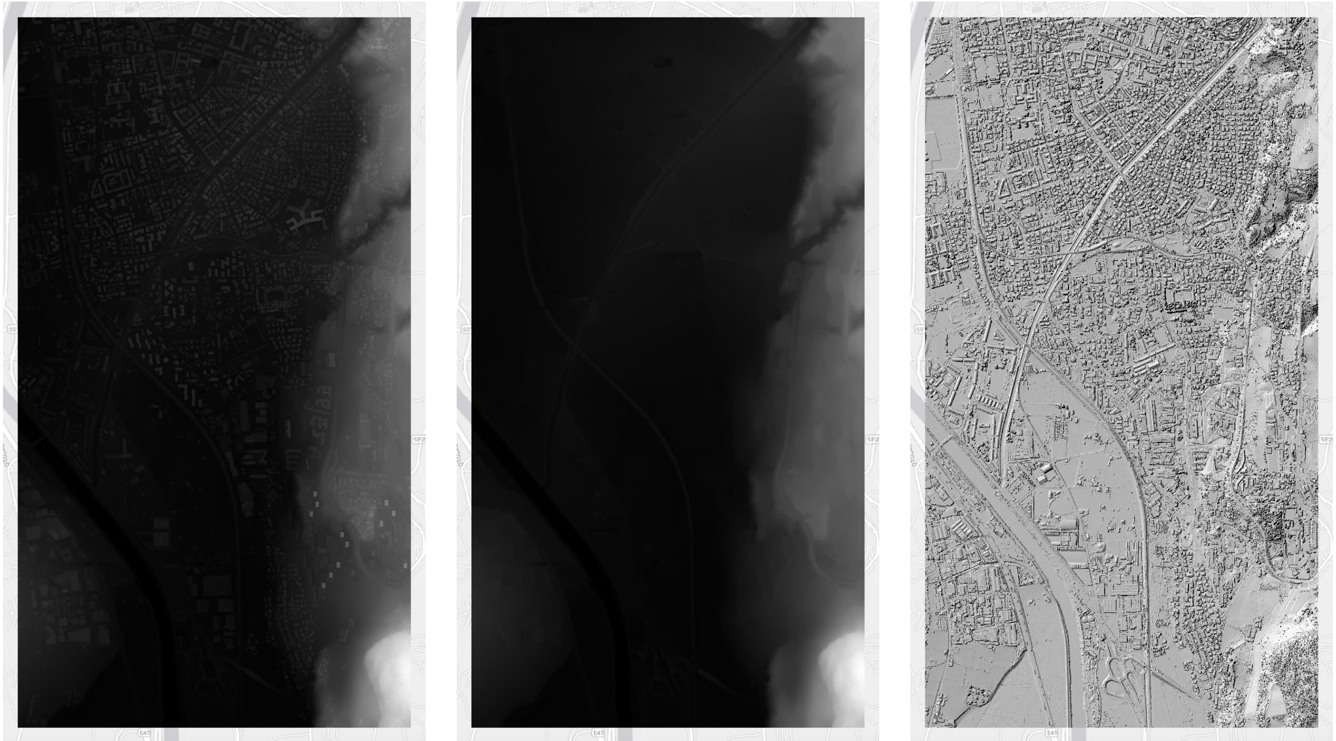
Work Allocation Summary

Part 1: Ming-Chieh Hu, Dimitrios Lioumis, Neelabh Singh, approx. 6 hours
Part 2: Ming-Chieh Hu, Dimitrios Lioumis, Neelabh Singh, approx. 12 hours
Part 3: Dimitrios Lioumis, approx. 8 hours
Part 4: Ming-Chieh Hu, approx. 8 hours

Contents

1 Part 1: Data integration, preparation and assessment

1.1 Exercise 1.1



dtm dsm dsm_shaded

Figure 1: Screenshots of the dtm, dsm and dsm_shaded layers

While loading and merging rasters in this exercise, we found that some data on the border of tiles are missing. (figure.2) It could be problematic to proceed with this data because the default setting of the "merge" function will simply set pixels with no data to 0, making further manipulation inaccurate.

We change the detail settings in the "merge" function to reveal the missing data pixels (figure.3). After that, we use "Fill NoData" (figure.4) to interpolate the missing values, with a kernel size 3 to only use the local values. By default in this function, pixels are interpolated using an inverse distance weighting (inv_dist).

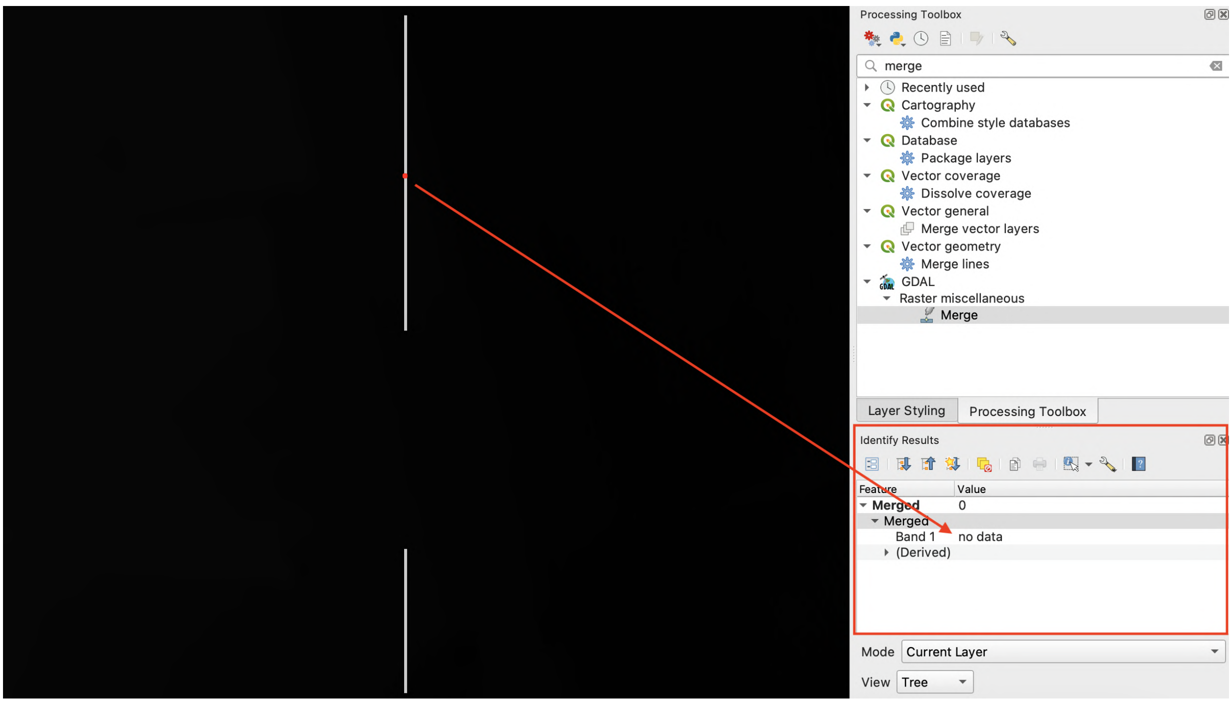


Figure 2: Inspecting data using "Identify Features" function

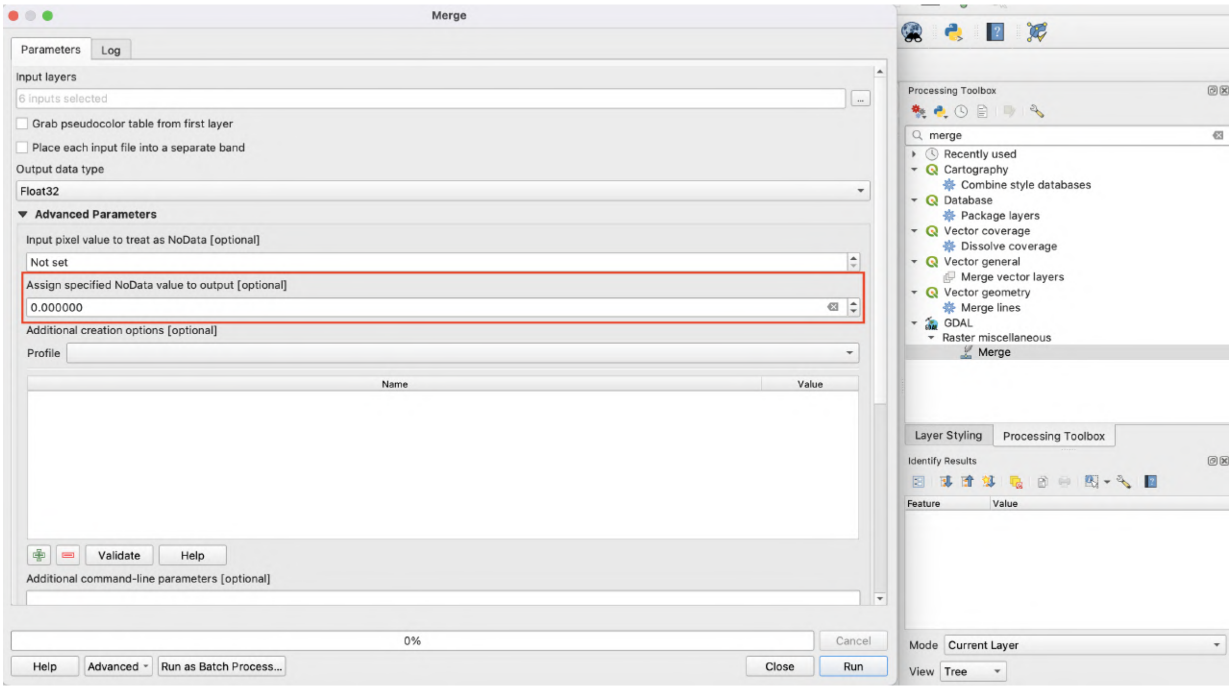


Figure 3: Extra settings to prevent NULL values being set to 0

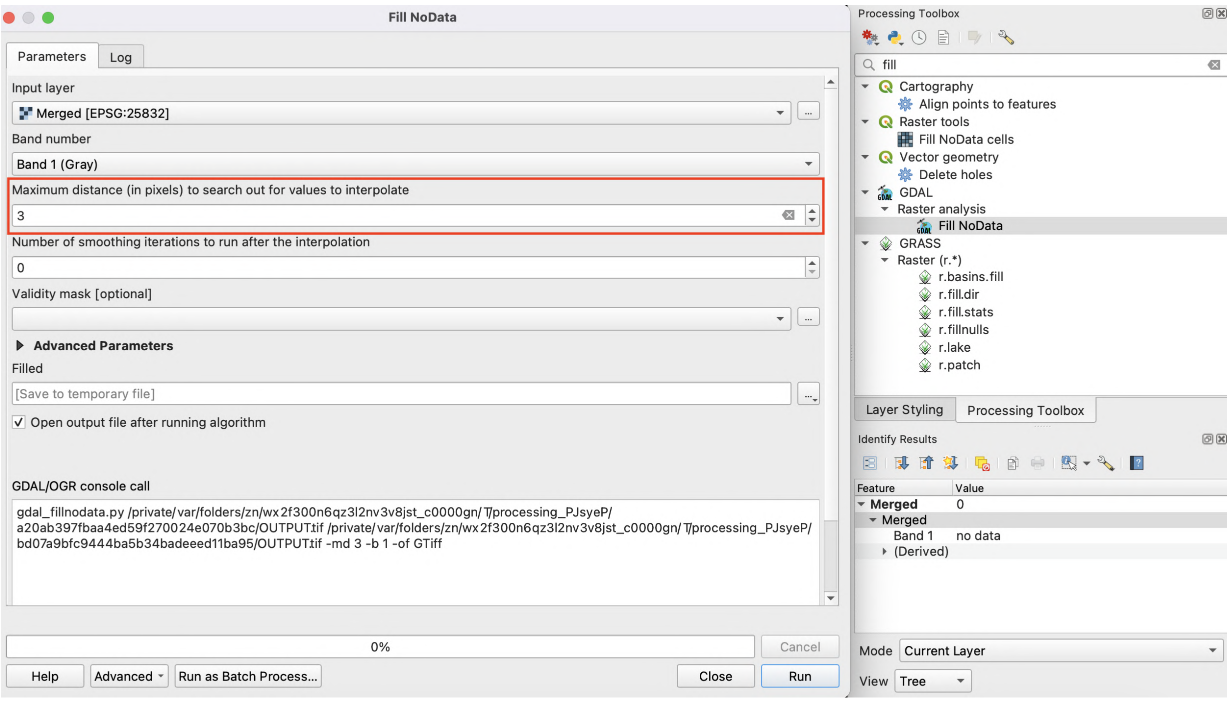


Figure 4: "Fill NoData" function parameters

1.2 Exercise 1.2

You can assume that the height is the result of a point cloud processing carried out before by other Geomatics colleagues of another office. It is however your duty to assess the validity of these results. For this reason, look at those buildings parts that might have been wrongly reconstructed. Search for those footprints with a $h_{roof} \leq 1.8m$.

How many buildings parts have you selected?
In total, **65 building parts have been selected**.

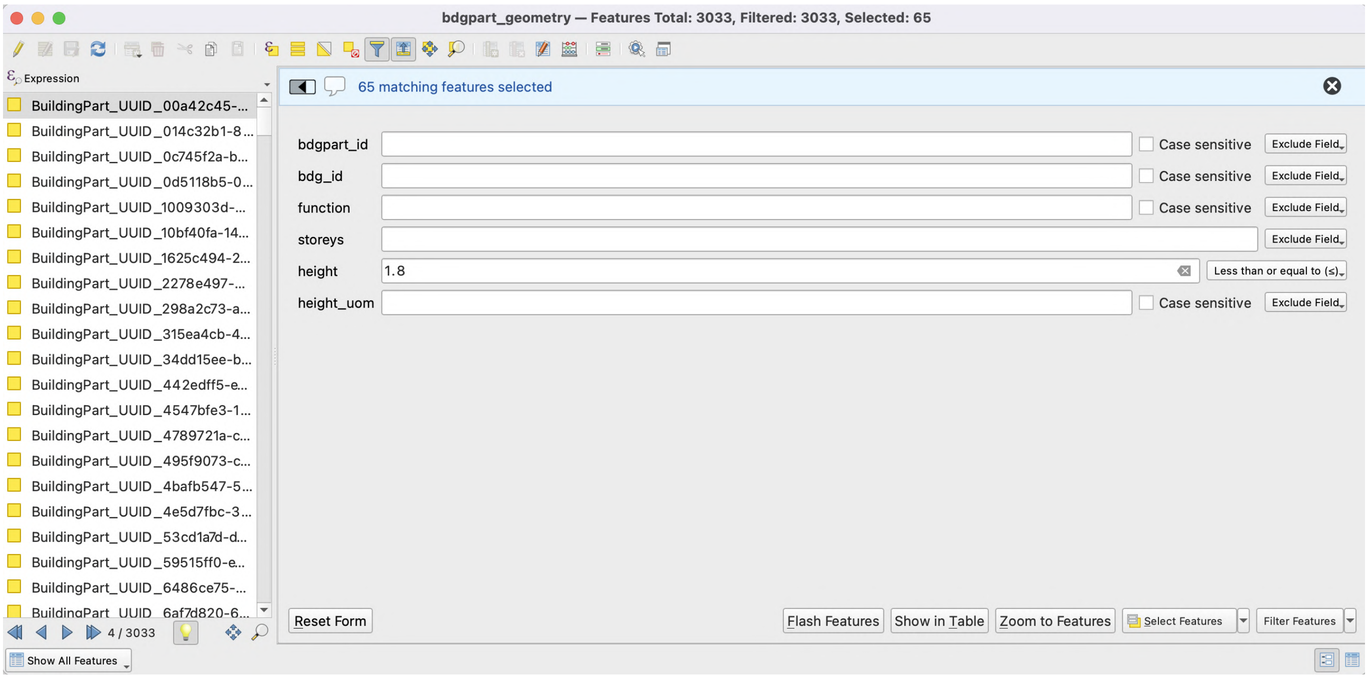


Figure 5: 65 bad building parts were selected

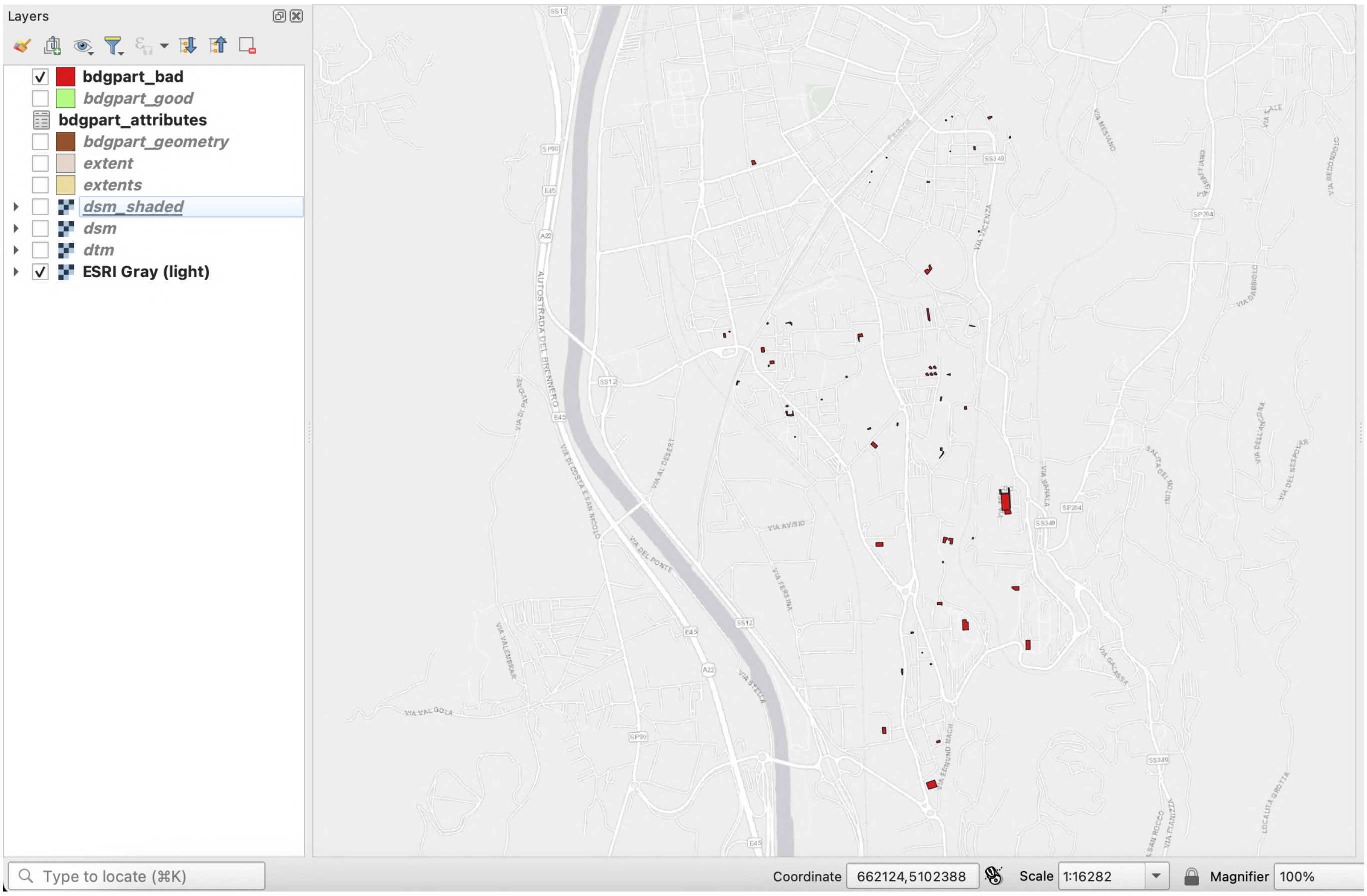


Figure 6: Visualization of bdgpart_bad.shp

1.3 Exercise 1.3

Regarding the previous exercise, write a short reflection (max 300 words) containing some reasoning and visual support of your findings.

- What can you say about the "badly reconstructed" building parts?
- What may have gone wrong in the building reconstruction process carried out by the "Geomatics colleagues" of the Trento office in Italy?

The bdg_part geometry layer has the building as a polygon feature including attributes like the roof height of the building, which is reconstructed using point cloud data. Total number of badly constructed buildings in the layer is 65, which have $h_{roof} \leq 1.8m$ meters. In which around 40 of them have a height value of 0, and the rest are less than 1.8m. These buildings in many cases are small features which are digitized incorrectly and not actually a building on the ground, or else, they have been digitized correctly but their height values are wrong. Following are some of the reasons because of which reconstruction process could have gone wrong:

1 - Temporal Difference We can see in the layer there are digitized buildings which are not present on the ground currently and have roof height of 0m. However, after checking the older imagery of the area we can see that these buildings were present but got demolished later on, so one thing could be possible that there is a temporal difference between the map or the data used for digitization and the DSM data or the current data used for adding height to the buildings.

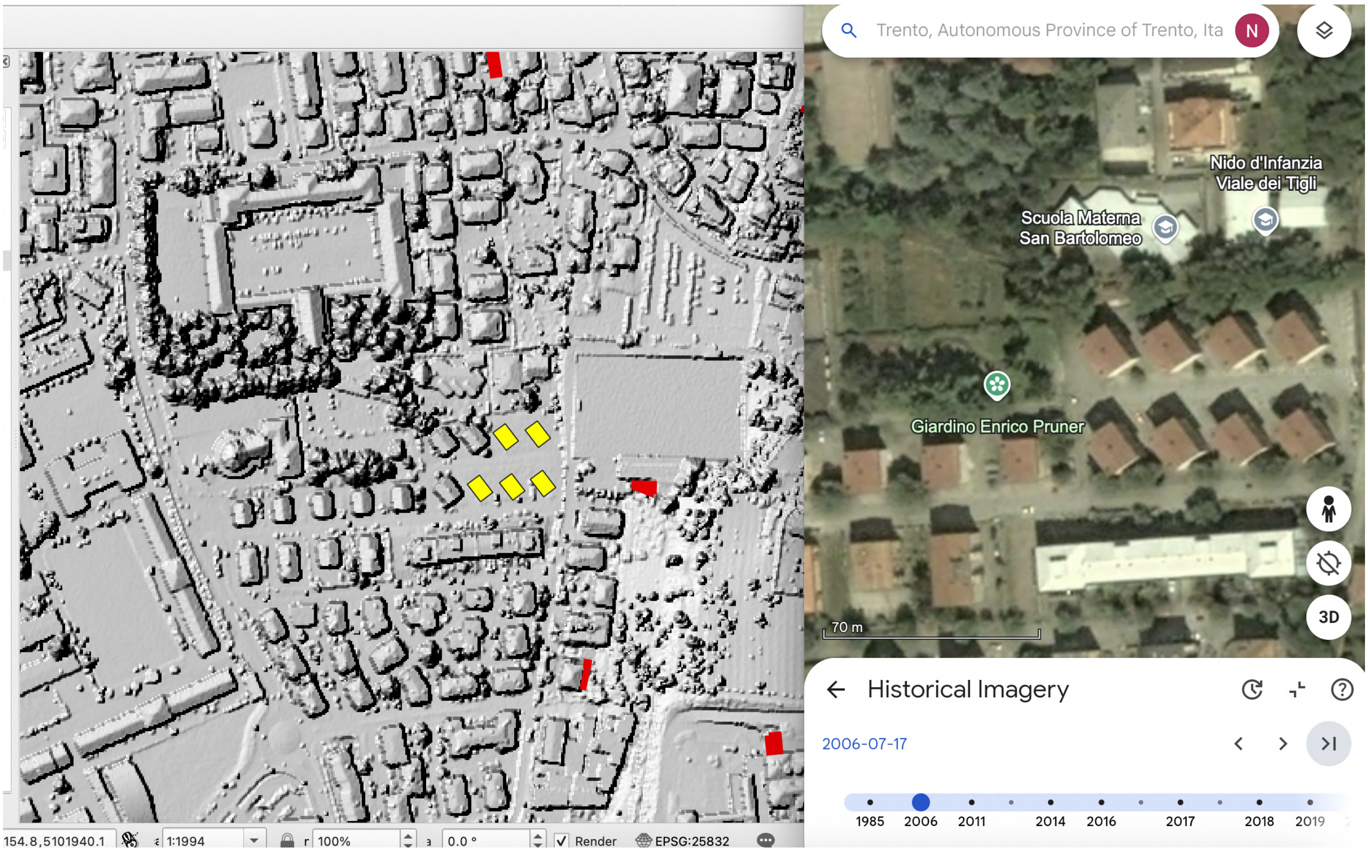


Figure 7: Examples of digitized buildings which were present during 2006 but not present currently

2 - Topographic features Across the site there are various elevations, and the floor of the building has sudden change from its surroundings. As a result, the estimated height of the building could be inaccurate.

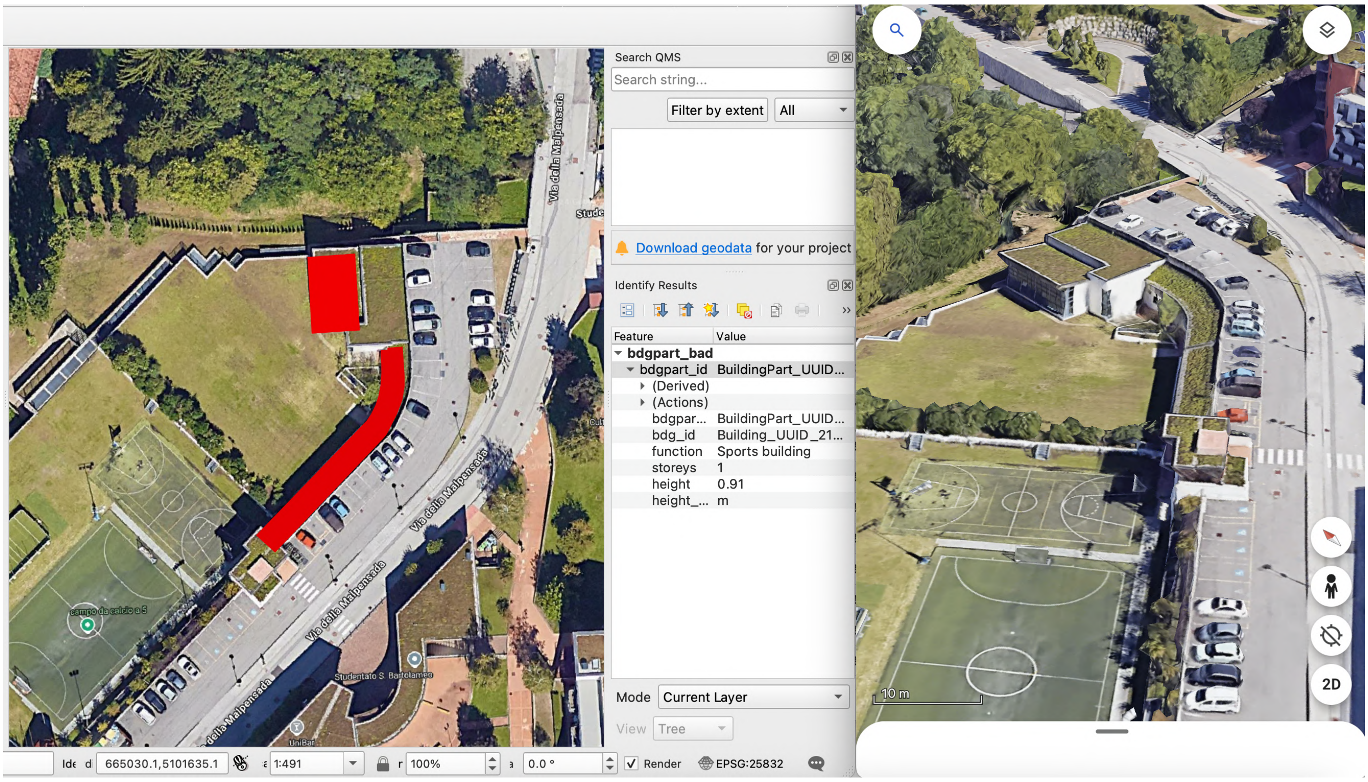


Figure 8: 3D view representing an elevated road at the level equal to roof of the building

3 - Digitization error There are some buildings which are originally something else on the ground. However, they look similar to the roof of the surrounding building features, so they could have been digitized inaccurately.

Figure 9: Figure representing a car park got digitized as building, due to similar looking concrete roof in the surrounding buildings

1.4 Exercise 1.4

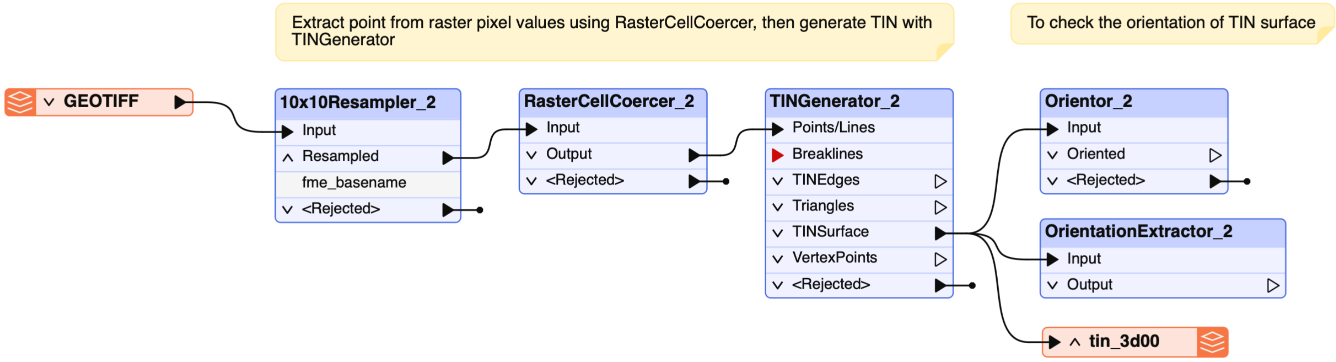


Figure 10: FME workflow

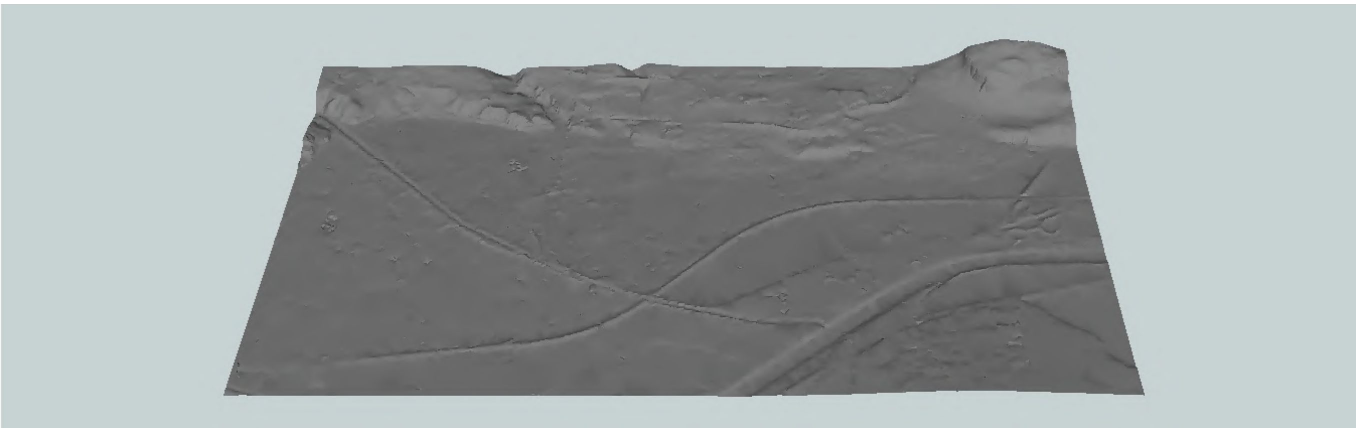


Figure 11: TIN surface

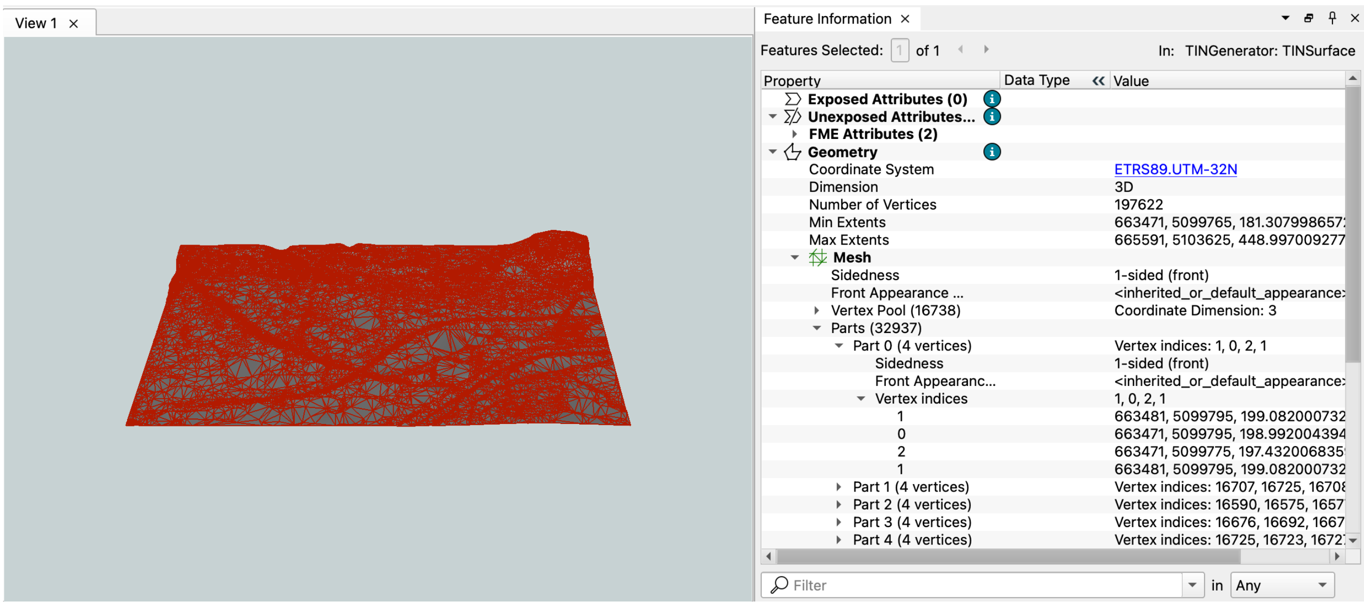


Figure 12: TIN surface (geometry data)

2 Part 2: Energy performance of residential buildings

2.1 Exercise 2.1

Calculate A_{gs} and V_{gross}

A_{gs} = `$area` ($area is the geodesic area)
V_{gross} = `"Ags"*"height"`
we use the A_{gs} to calculate the V_{gross} here.

Filter expression

```
"bdgpart_id" = 'BuildingPart_UUID_0a3f05f4-1331-417e-b2cd-303724cf'
OR "bdgpart_id" = 'BuildingPart_UUID_0c44e4d6-c8c5-4c94-bc9e-255950bb'
OR "bdgpart_id" = 'BuildingPart_UUID_0cbb2b41-852b-4761-8b3d-412e5ba0'
OR "bdgpart_id" = 'BuildingPart_UUID_0f0b0bc6-aca9-4dd9-9bf3-79485552'
```

Result table

Table1

bdgpart_id	Value type	Value	Value_uom
BuildingPart_UUID_0a3f05f4-1331-417e-b2cd-303724cf	A_{gs}	260.563	m^2
BuildingPart_UUID_0c44e4d6-c8c5-4c94-bc9e-255950bb	A_{gs}	101.857	m^2
BuildingPart_UUID_0cbb2b41-852b-4761-8b3d-412e5ba0	V_{gross}	1625.661	m^3
BuildingPart_UUID_0f0b0bc6-aca9-4dd9-9bf3-79485552	V_{gross}	8649.509	m^3

2.2 Exercise 2.2

How many residential and mixed-residential building parts are there in total?
There are 598 Mixed-residential building parts, 1749 Residential building parts, 2347 in total.

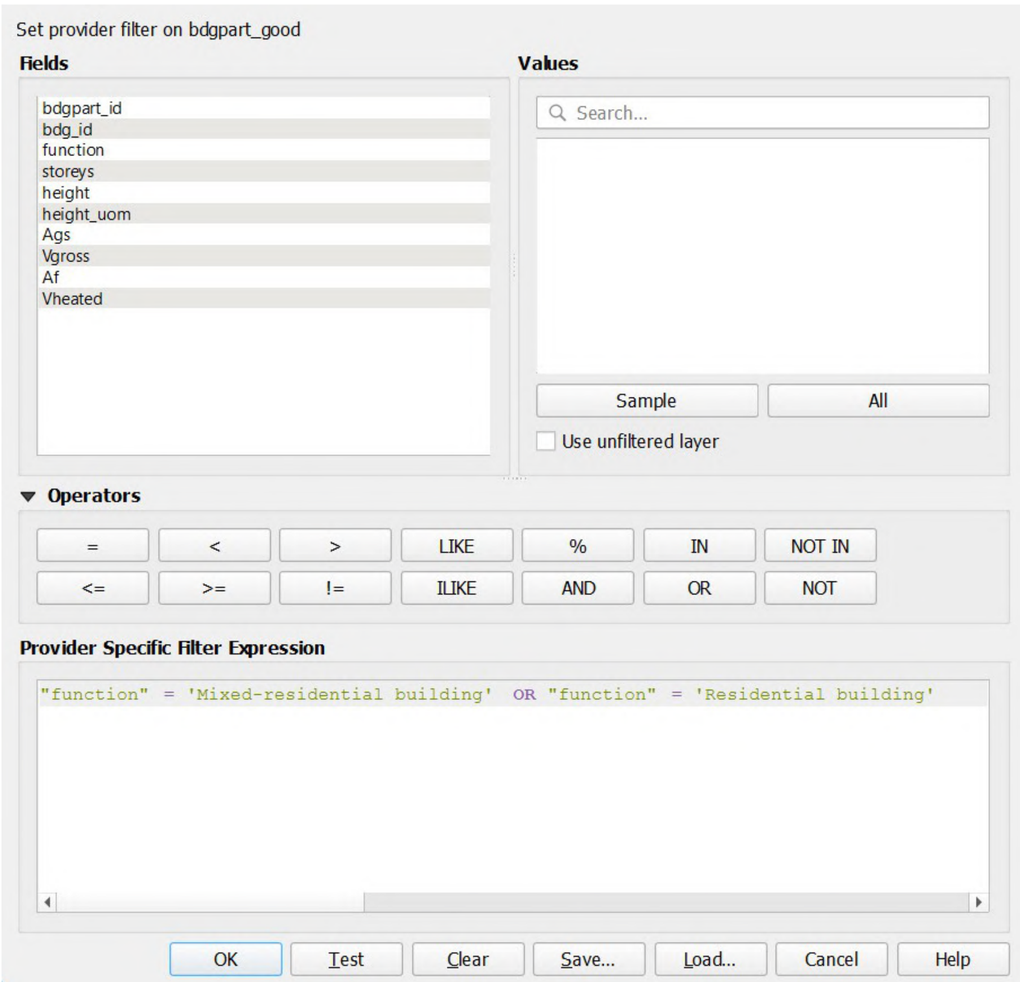


Figure 13: Selecting residential and mixed-residential building parts

Calculate A_f and V_{heated}

A_f = `0.85*"Ags"*"storeys"`
V_{heated} = `0.65*"Vgross"`

Filter expression

```
"bdgpart_id" = 'BuildingPart_UUID_0a2de3f3-faea-45cf-9259-0909ccc8'
OR "bdgpart_id" = 'BuildingPart_UUID_0a4dc959-9567-4ad7-87f5-3b3759d4'
OR "bdgpart_id" = 'BuildingPart_UUID_0a7386ca-3079-4090-b5d9-e9260773'
OR "bdgpart_id" = 'BuildingPart_UUID_00a95184-61eb-4b32-89e3-53311328'
```

Result table

Table2

bdgpart_id	$A_f[m^2]$	$V_{heated}[m^3]$
BuildingPart_UUID_0a2de3f3-faea-45cf-9259-0909ccc8	267.590	1068.157
BuildingPart_UUID_0a4dc959-9567-4ad7-87f5-3b3759d4	0	0
BuildingPart_UUID_0a7386ca-3079-4090-b5d9-e9260773	97.296	301.332
BuildingPart_UUID_00a95184-61eb-4b32-89e3-53311328	759.900	2133.799

2.3 Exercise 2.3

For aggregate function, we dropped the *bdgpart_id*, *height*, *height_uom*, *storey* because these attributes is only used in building parts data. We also dropped the *function* because we will join another function field from bdg_attributes.csv later in section 2.4. The aggregate function dissolves geometries based on user selected feature (in this case bdg_id), so we don't have to call the dissolve function.

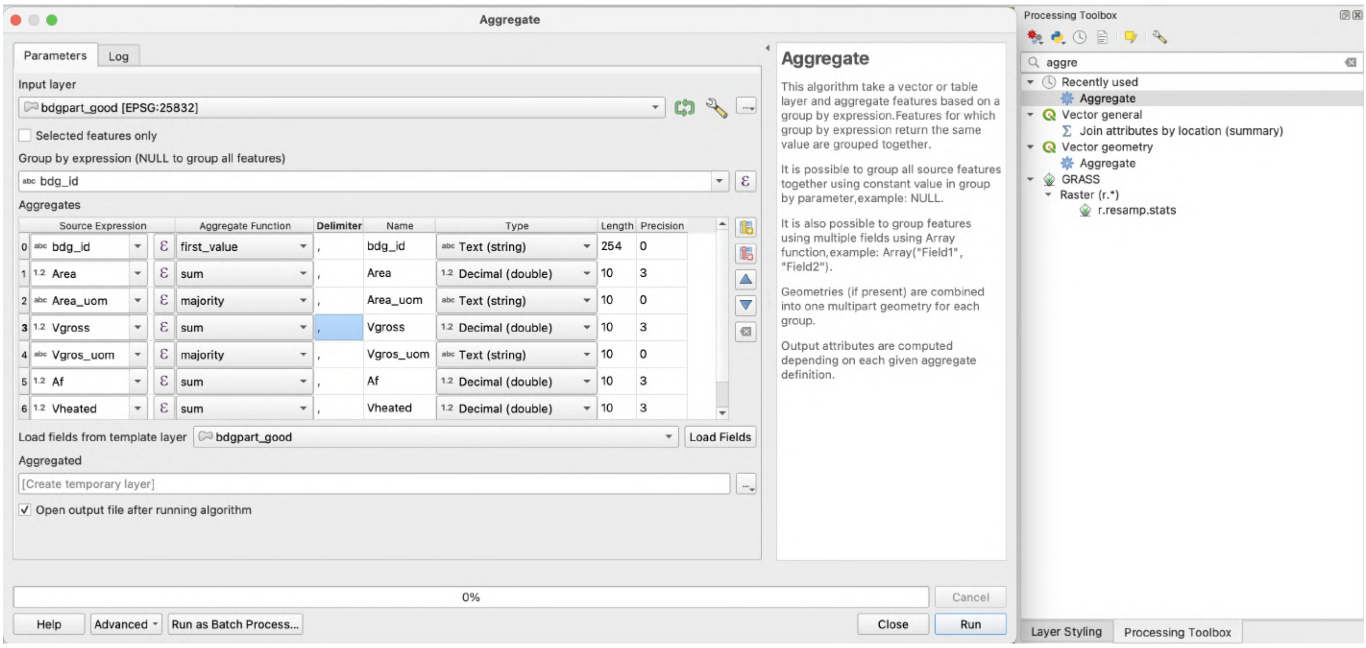
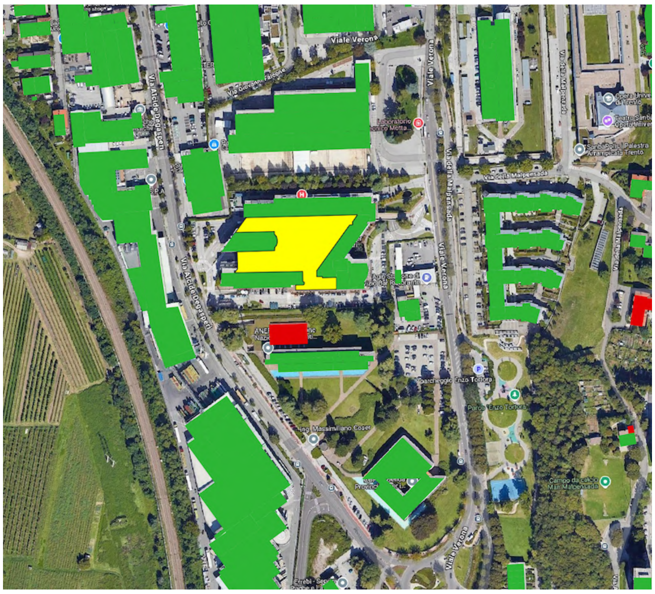
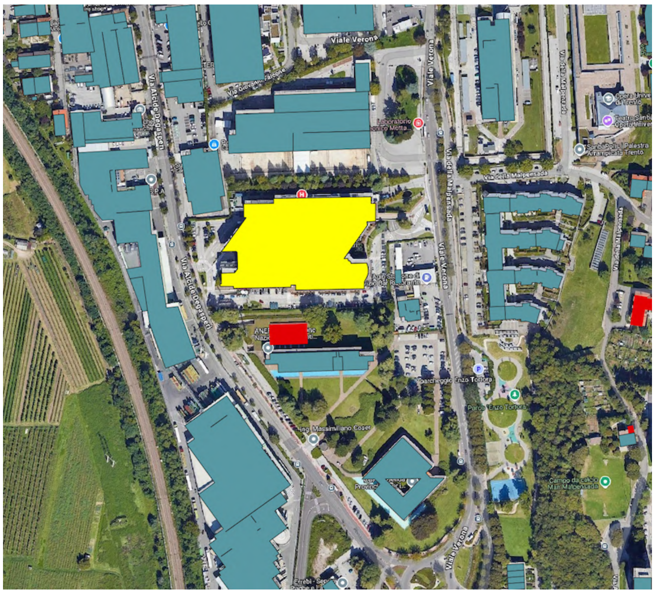


Figure 14: Aggregate function parameters



Original layer (Building parts)



Aggregated layer (Building parts dissolved)

Figure 15: Geometries before and after aggregation

How many buildings are resulting from this operation?
There are 2256 buildings after aggregation.

Filter expression

```
"bdg_id" = 'Building_UUID_0a8d8c0a-a11e-4b6e-a21e-60b493ce6573'
OR "bdg_id" = 'Building_UUID_0b5b3583-8df5-4c60-b429-e63bdddfd44e'
OR "bdg_id" = 'Building_UUID_0c5a87cb-765d-467e-9279-7d6e93fc9fe8'
OR "bdg_id" = 'Building_UUID_0d7e1777-6e91-4988-92fd-ada22baea55a'
```

Result table

Table3

bdg_id	$A_{gs}[m^2]$	$A_f[m^2]$	$V_{gross}[m^3]$	$V_{heated}[m^3]$
Building_UUID_0a8d8c0a-a11e-4b6e-a21e-60b493ce6573	156.386	0	577.064	0
Building_UUID_0b5b3583-8df5-4c60-b429-e63bdddfd44e	271.283	862.165	3986.161	2591.005
Building_UUID_0c5a87cb-765d-467e-9279-7d6e93fc9fe8	77.606	191.686	830.503	539.827
Building_UUID_0d7e1777-6e91-4988-92fd-ada22baea55a	182.860	621.724	2841.644	1847.069

2.4 Exercise 2.4

We found some issue when reading the **bdg_attributes.csv** file (figure.16). There are quotation marks in names like: **Scuola Elementare "G.B. De Gaspari"**, there's also quotation mark in building function field like: **"Warehouse, Storage"**. The csv data looked just fine in professional CSV readers like Excel, Numbers and FME, but reading this kind of file in QGIS can cause some data loss.

Some data samples of bdg_attributes.csv

```
Building_UUID_...,Bdg_91305,"Warehouse, Storage"
...
Building_UUID_...,Scuola Elementare "G.B. De Gaspari",Education building
```

In QGIS, the default csv text delimiter is comma (,) and quote by quotation mark ("). But for this specific case, we have to modify the csv file to preserve all the names and building functions (otherwise it's displayed as NULL in QGIS).

```
Building_UUID_716518b2-5b56-47fc-967f-81474a6c31d5,Bdg_90315,Residential building
Building_UUID_66852124-508b-436f-9c68-a524a297784d,Bdg_88509,Residential building
Building_UUID_d3157de9-9d02-46ba-bbc8-d9c176c1f2e6,Complesso I Girasoli,Mixed-residential building
Building_UUID_0f395f1b-7a05-4379-a0bf-dc8e49410780,Bdg_91305,"Warehouse, Storage"
Building_UUID_178dc316-1196-4e9c-8ad9-c67ad3ce62a3,Bdg_92049,Sports building
Building_UUID_5df22e1e-53cf-47a7-988e-838fc95ec7b9,Bdg_90382,Residential building
Building_UUID_0bc5e1ce-031d-4652-a074-b2aaf572f5a1,Bdg_90142,Residential building
Building_UUID_fa839ef3-478c-4cc2-ad75-2ff65bd52a81,Bdg_89825,Residential building
Building_UUID_46f51f3e-a6ff-4d83-a15a-8c15a23d4d8c,Bdg_90697,Military building
Building_UUID_165608d0-646a-436d-9cbc-a25de9eb3a0a,Case al Fersina,Residential building
Building_UUID_df56165b-7a07-4c8a-b8ae-f6257de8e12f,Bdg_90628,Residential building
Building_UUID_d2fc05b1-e934-4def-85b1-fc4556026e31,Bdg_90020,Garage
Building_UUID_6484c913-b12f-460b-acfe-6a242cfe80ba,Condominio Sestante,Residential building
Building_UUID_0d5f2756-4aab-467e-b63f-94cf0dd3b568,Bdg_90808,Residential building
Building_UUID_added7d5-1444-4c52-b46d-2ea0dcb8f8df,Scuola Elementare "G.B. De Gaspari",Education building
Building_UUID_855c93d9-b8d0-472a-8d02-9443ad7b3b56,Bdg_92807,Residential building
Building_UUID_12e47c0f-b8cc-4899-8356-937d08f836dc,Condominio Buccella 9,Residential building
Building_UUID_f865b264-3545-4ff7-92fd-1f1f49221740,Bdg_93194,Mixed-residential building
```

Figure 16: Original bdg_attributes.csv

```
Building_UUID_716518b2-5b56-47fc-967f-81474a6c31d5,Bdg_90315,Residential building
Building_UUID_66852124-508b-436f-9c68-a524a297784d,Bdg_88509,Residential building
Building_UUID_d3157de9-9d02-46ba-bbc8-d9c176c1f2e6,Complesso I Girasoli,Mixed-residential building
Building_UUID_0f395f1b-7a05-4379-a0bf-dc8e49410780,Bdg_91305,`Warehouse, Storage`
Building_UUID_178dc316-1196-4e9c-8ad9-c67ad3ce62a3,Bdg_92049,Sports building
Building_UUID_5df22e1e-53cf-47a7-988e-838fc95ec7b9,Bdg_90382,Residential building
Building_UUID_0bc5e1ce-031d-4652-a074-b2aaf572f5a1,Bdg_90142,Residential building
Building_UUID_fa839ef3-478c-4cc2-ad75-2ff65bd52a81,Bdg_89825,Residential building
Building_UUID_46f51f3e-a6ff-4d83-a15a-8c15a23d4d8c,Bdg_90697,Military building
Building_UUID_165608d0-646a-436d-9cbc-a25de9eb3a0a,Case al Fersina,Residential building
Building_UUID_df56165b-7a07-4c8a-b8ae-f6257de8e12f,Bdg_90628,Residential building
Building_UUID_d2fc05b1-e934-4def-85b1-fc4556026e31,Bdg_90020,Garage
Building_UUID_6484c913-b12f-460b-acfe-6a242cfe80ba,Condominio Sestante,Residential building
Building_UUID_0d5f2756-4aab-467e-b63f-94cf0dd3b568,Bdg_90808,Residential building
Building_UUID_added7d5-1444-4c52-b46d-2ea0dcb8f8df,Scuola Elementare "G.B. De Gaspari",Education building
Building_UUID_855c93d9-b8d0-472a-8d02-9443ad7b3b56,Bdg_92807,Residential building
Building_UUID_12e47c0f-b8cc-4899-8356-937d08f836dc,Condominio Buccella 9,Residential building
Building_UUID_f865b264-3545-4ff7-92fd-1f1f49221740,Bdg_93194,Mixed-residential building
```

Figure 17: Updated bdg_attributes.csv

We updated some of the building functions in bdg_attributes.csv by changing the quotation mark (") into back-quote/inverted comma (‘).
In total, 11 data entities have been modified (figure.17).

- 9 data entities of "Warehouse, Storage"
- 1 data entity of "Commercial building, Library"
- 1 data entity of "Commercial building, Factory"

To read the modified file, use the new settings displayed in figure.18.

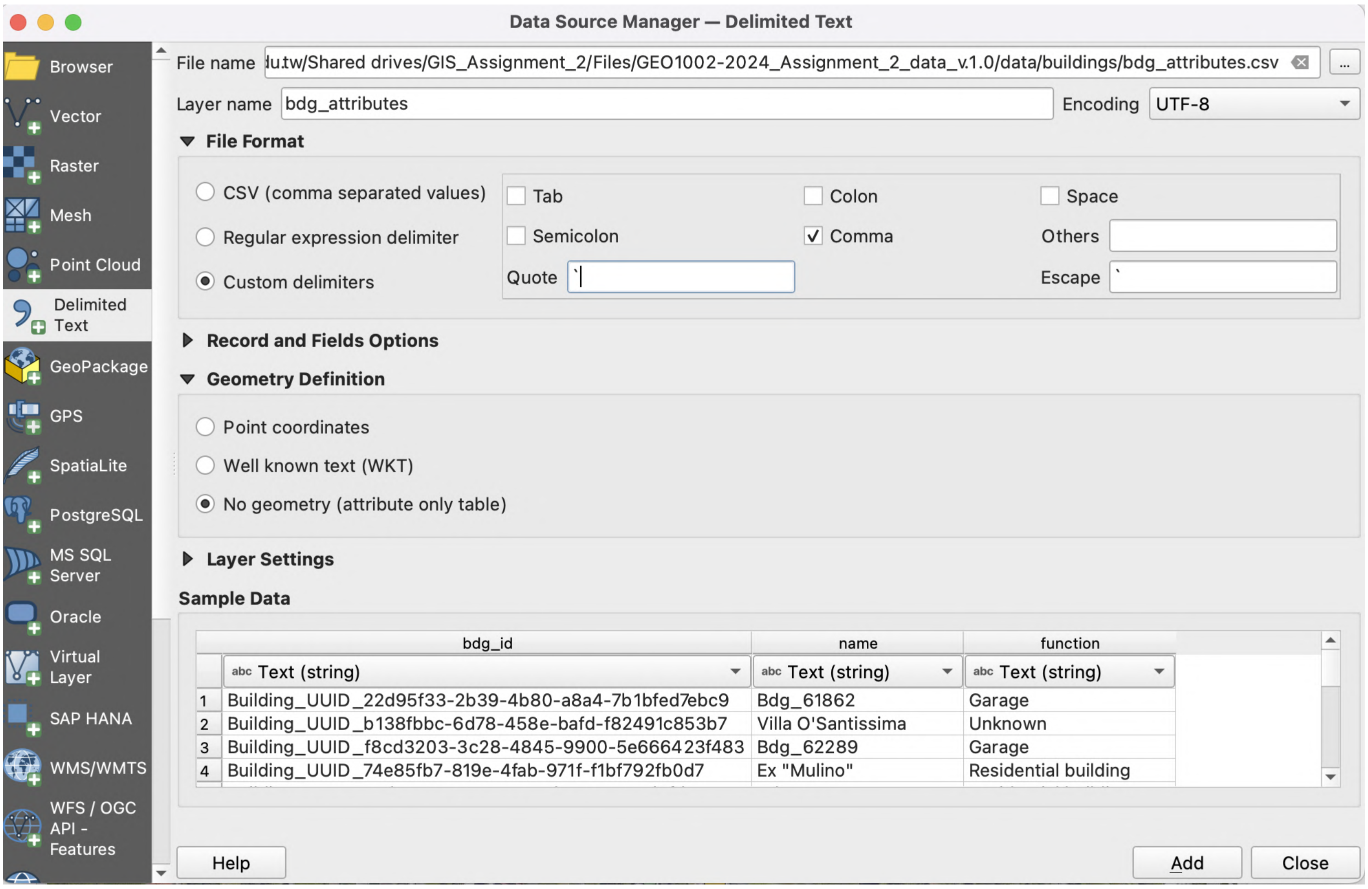


Figure 18: Reading the updated bdg_attributes.csv

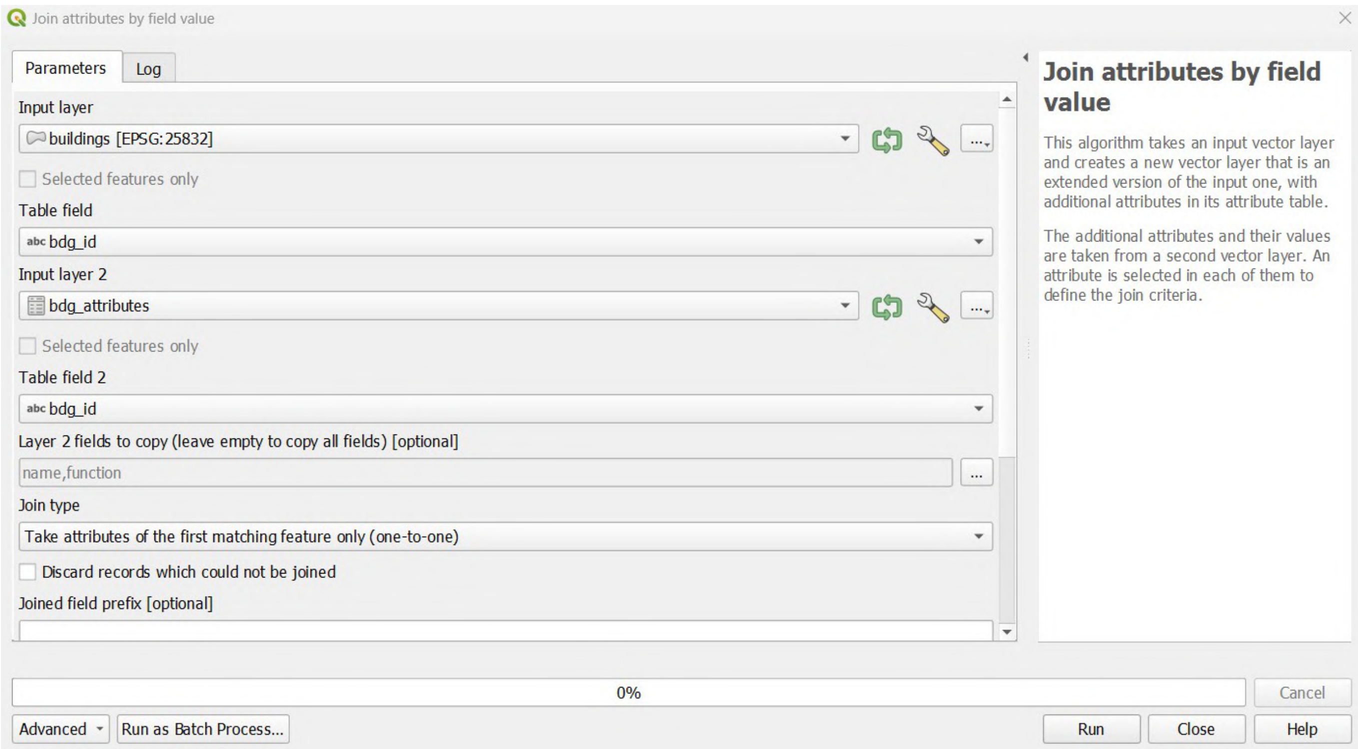
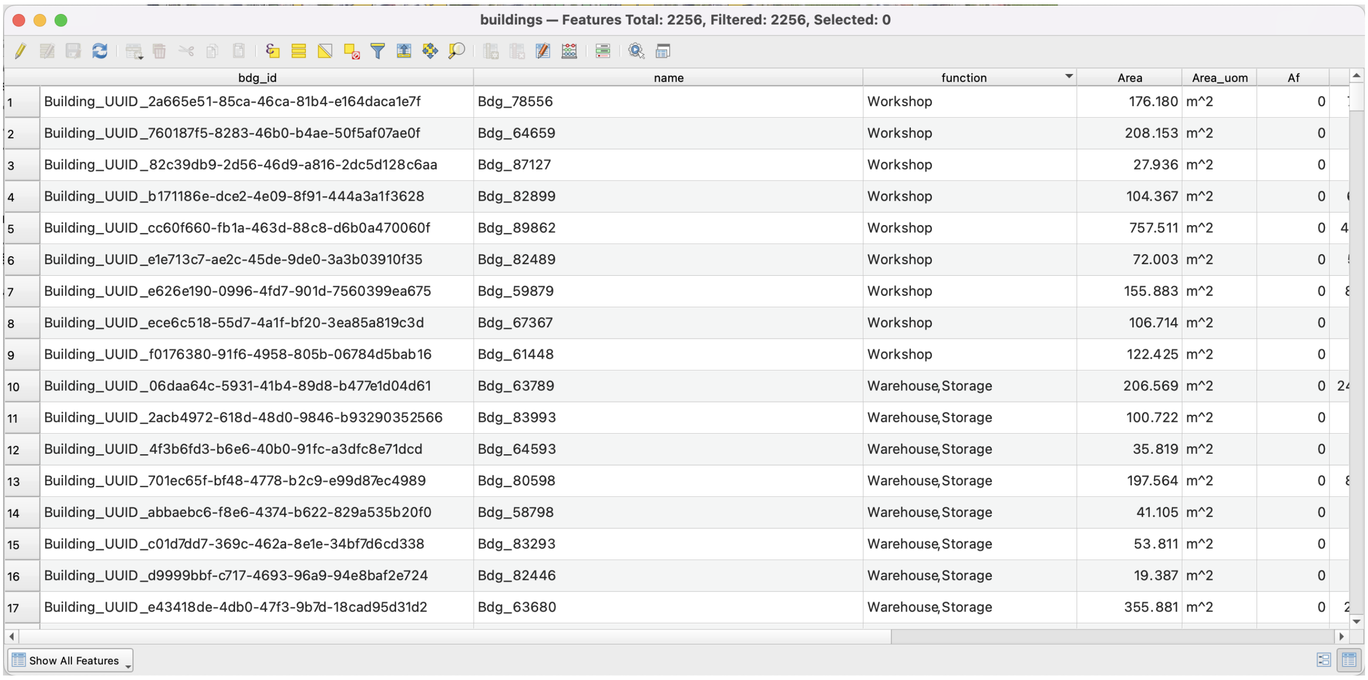


Figure 19: Joining the table



buildings — Features Total: 2256, Filtered: 2256, Selected: 0

	bdg_id	name	function	Area	Area_uom	Af	
1	Building_UUID_2a665e51-85ca-46ca-81b4-e164daca1e7f	Bdg_78556	Workshop	176.180	m^2	0	'
2	Building_UUID_760187f5-8283-46b0-b4ae-50f5af07ae0f	Bdg_64659	Workshop	208.153	m^2	0	
3	Building_UUID_82c39db9-2d56-46d9-a816-2dc5d128c6aa	Bdg_87127	Workshop	27.936	m^2	0	
4	Building_UUID_b171186e-dce2-4e09-8f91-444a3a1f3628	Bdg_82899	Workshop	104.367	m^2	0	(
5	Building_UUID_cc60f660-fb1a-463d-88c8-d6b0a470060f	Bdg_89862	Workshop	757.511	m^2	0	4
6	Building_UUID_e1e713c7-ae2c-45de-9de0-3a3b03910f35	Bdg_82489	Workshop	72.003	m^2	0	!
7	Building_UUID_e626e190-0996-4fd7-901d-7560399ea675	Bdg_59879	Workshop	155.883	m^2	0	8
8	Building_UUID_ece6c518-55d7-4a1f-bf20-3ea85a819c3d	Bdg_67367	Workshop	106.714	m^2	0	
9	Building_UUID_f0176380-91f6-4958-805b-06784d5bab16	Bdg_61448	Workshop	122.425	m^2	0	
10	Building_UUID_06daa64c-5931-41b4-89d8-b477e1d04d61	Bdg_63789	Warehouse,Storage	206.569	m^2	0	24
11	Building_UUID_2acb4972-618d-48d0-9846-b93290352566	Bdg_83993	Warehouse,Storage	100.722	m^2	0	
12	Building_UUID_4f3b6fd3-b6e6-40b0-91fc-a3dfc8e71dcd	Bdg_64593	Warehouse,Storage	35.819	m^2	0	
13	Building_UUID_701ec65f-bf48-4778-b2c9-e99d87ec4989	Bdg_80598	Warehouse,Storage	197.564	m^2	0	8
14	Building_UUID_abbaebc6-f8e6-4374-b622-829a535b20f0	Bdg_58798	Warehouse,Storage	41.105	m^2	0	
15	Building_UUID_c01d7dd7-369c-462a-8e1e-34bf7d6cd338	Bdg_83293	Warehouse,Storage	53.811	m^2	0	
16	Building_UUID_d9999bbf-c717-4693-96a9-94e8baf2e724	Bdg_82446	Warehouse,Storage	19.387	m^2	0	
17	Building_UUID_e43418de-4db0-47f3-9b7d-18cad95d31d2	Bdg_63680	Warehouse,Storage	355.881	m^2	0	2

Show All Features

Figure 20: The joined buildings layer with no NULL values

2.5 Exercise 2.5

Calculate $Q_{H,p}$ and EP_H value

$Q_{H,p}$ = `"Vheated"*"v_heat_ped"*0.001`
EP_H = `1000*"Qhp"/"Af"`

Filter expression

```
"name"='Bdg_87826'
OR "name"='Condominio Mope'
OR "name"='Condominio Sole 3'
```

Result table

Table4

Building name	Scenario	$Q_{H,p}[MWh/a]$	$EP_H[kWh/(m2 \cdot a)]$
Bdg_87826	Original state	86.581	377.097
	Refurbished state	13.954	60.776
Condominio Mope	Original state	538.504	148.134
	Refurbished state	96.915	26.660
Condominio Sole 3	Original state	39.917	24.809
	Refurbished state	39.917	24.809

2.6 Exercise 2.6

Calculate a value

```
CASE
    WHEN "Af" <= 50 THEN 1.8
    WHEN "Af" <= 200 THEN 4.514*"Af"^(-0.2356)
    WHEN "Af" > 200 THEN 1.3
    ELSE NULL
END
```

Calculate $Q_{h,W}$ value

```
0.001*1.162*"a"*"Af"*(40-15)*365
```

Calculate $Q_{W,p}$ value

```
"QhW"/0.8
```

Calculate EP_W value

```
"QWp"/"Af"
```

Filter expression

```
"bdg_id"='Building_UUID_0c5a87cb-765d-467e-9279-7d6e93fc9fe8'
OR "bdg_id"='Building_UUID_3ad65628-a606-4865-97de-1facec04414d'
OR "bdg_id"='Building_UUID_44e8b84f-5c77-4cf2-87f3-5c9b8c23341d'
OR "bdg_id"='Building_UUID_b99f0521-d03b-4348-a871-676e08acd3b4'
OR "bdg_id"='Building_UUID_f900720c-2970-4cb0-a6f4-f4003251e77e'
```

Result table

Table5

bdgpart_id	$A_f[m^2]$	$a[m3/(m2 \cdot d)]$
Building_UUID_0c5a87cb-765d-467e-9279-7d6e93fc9fe8	77.606	1.3085
Building_UUID_3ad65628-a606-4865-97de-1facec04414d	26.608	1.8000
Building_UUID_44e8b84f-5c77-4cf2-87f3-5c9b8c23341d	70.902	1.3000
Building_UUID_b99f0521-d03b-4348-a871-676e08acd3b4	220.298	1.3158
Building_UUID_f900720c-2970-4cb0-a6f4-f4003251e77e	59.460	1.7914

2.7 Exercise 2.7

Calculate EP_{gl} value

```
"EPh"+"EPw"
```

Calculate energy class label

```
CASE
    WHEN "EPgl" < 0 THEN 'N/A'
    WHEN "EPgl" <= 30 THEN 'A+'
    WHEN "EPgl" <= 40 THEN 'A'
    WHEN "EPgl" <= 50 THEN 'B+'
    WHEN "EPgl" <= 60 THEN 'B'
    WHEN "EPgl" <= 80 THEN 'C+'
    WHEN "EPgl" <= 120 THEN 'C'
    WHEN "EPgl" <= 180 THEN 'D'
    WHEN "EPgl" <= 225 THEN 'E'
    WHEN "EPgl" <= 270 THEN 'F'
    WHEN "EPgl" > 270 THEN 'G'
    ELSE 'N/A'
END
```

Filter expression

```
"scenario_i"=1
AND (
    "bdg_id"='Building_UUID_0a0d4438-b36b-4413-a7ed-87642ffda7a5' OR
    "bdg_id"='Building_UUID_0ab13f9f-523d-41bd-83c5-03fa0b158187' OR
    "bdg_id"='Building_UUID_0b5b3583-8df5-4c60-b429-e63bdddfd44e' OR
    "bdg_id"='Building_UUID_0b7dc746-8706-4508-8610-2811d9ed9d48' OR
    "bdg_id"='Building_UUID_00c94702-d753-4e24-b9b7-7a67bb18c4f5' OR
    "bdg_id"='Building_UUID_f4a77093-7db3-4fac-9163-5701198c7c50'
)
```

Result table

Table7				
Scenario 1 ("Original state")				
bdg_id	EP_H	EP_W	EP_{gl}	EP_{gl} label
Building_UUID_0a0d4438-b36b-4413-a7ed-87642ffda7a5	140.489	17.230	157.230	D
Building_UUID_0ab13f9f-523d-41bd-83c5-03fa0b158187	NULL	NULL	NULL	N/A
Building_UUID_0b5b3583-8df5-4c60-b429-e63bdddfd44e	192.274	17.230	209.230	E
Building_UUID_0b7dc746-8706-4508-8610-2811d9ed9d48	180.202	19.140	199.140	E
Building_UUID_00c94702-d753-4e24-b9b7-7a67bb18c4f5	186.678	23.857	210.857	E
Building_UUID_f4a77093-7db3-4fac-9163-5701198c7c50	9.703	18.149	28.149	A+

Filter expression

```
"scenario_i"=2
AND (
    "bdg_id"='Building_UUID_0a0d4438-b36b-4413-a7ed-87642ffda7a5' OR
    "bdg_id"='Building_UUID_0ab13f9f-523d-41bd-83c5-03fa0b158187' OR
    "bdg_id"='Building_UUID_0b5b3583-8df5-4c60-b429-e63bdddfd44e' OR
    "bdg_id"='Building_UUID_0b7dc746-8706-4508-8610-2811d9ed9d48' OR
    "bdg_id"='Building_UUID_00c94702-d753-4e24-b9b7-7a67bb18c4f5' OR
    "bdg_id"='Building_UUID_f4a77093-7db3-4fac-9163-5701198c7c50'
)
```

Result table

Table7				
Scenario 2 ("Refurbished state")				
bdg_id	EP_H	EP_W	EP_{gl}	EP_{gl} label
Building_UUID_0a0d4438-b36b-4413-a7ed-87642ffda7a5	26.498	17.230	43.230	B+
Building_UUID_0ab13f9f-523d-41bd-83c5-03fa0b158187	NULL	NULL	NULL	N/A
Building_UUID_0b5b3583-8df5-4c60-b429-e63bdddfd44e	43.395	17.230	60.230	C+
Building_UUID_0b7dc746-8706-4508-8610-2811d9ed9d48	55.006	19.140	74.140	C+
Building_UUID_00c94702-d753-4e24-b9b7-7a67bb18c4f5	34.170	23.857	57.857	B
Building_UUID_f4a77093-7db3-4fac-9163-5701198c7c50	9.703	18.149	28.149	A+

2.8 Exercise 2.8

First we filter based on the scenarios (Original State - Scenario 1 & Refurbished State - Scenario 2).

1. Loaded the census_areas.shp.

2. Made an intersection between buildings and the census polygons.

3. Ran statistics by category to count the polygons that intersect with the census polygons (figure.21).

4. Found that **Building_UUID_c77eacff-1249-4f98-91de-ad33862bdebd** has 4 records instead of 2. That means that this polygon overlaps 2 census polygons. If we had not taken that to consideration, for the overlapping census polygons, $Q_{H,p}$ would have been calculated twice. (figure.22, 23).

5. To fix this we calculated the centroid of the buildings polygons, so no point can intersect two census polygons(figure.24, 25).

6. We ran the join attributes by location (summary) and calculated the sum of $Q_{H,p}$ inside the census polygons for Scenario 1.

7. Then we ran again the join attributes by location (summary) and calculated the sum of $Q_{H,p}$ inside the output polygons of the aforementioned step, but for Scenario 2 this time. That way we joined both summarized fields ($Q_{H,p}$ for Scenario 1 and $Q_{H,p}$ for Scenario 2) without taking one extra step to join them.

8. Then we calculated the difference (%) based on the following formula rounding it up to 1 decimal place:

   ($Q_{H,p}$ (1) - $Q_{H,p}$ (2)) * 100 / $Q_{H,p}$ (1)

Result table

Table8

section_id	$Q_{H,p}[MWh/a]$		Diff [%]
	Scenario 1	Scenario 2	
108	835.060	126.657	84.8
134	2125.218	415.817	80.4
266	7731.977	2281.099	70.5
418	4359.546	909.195	79.1

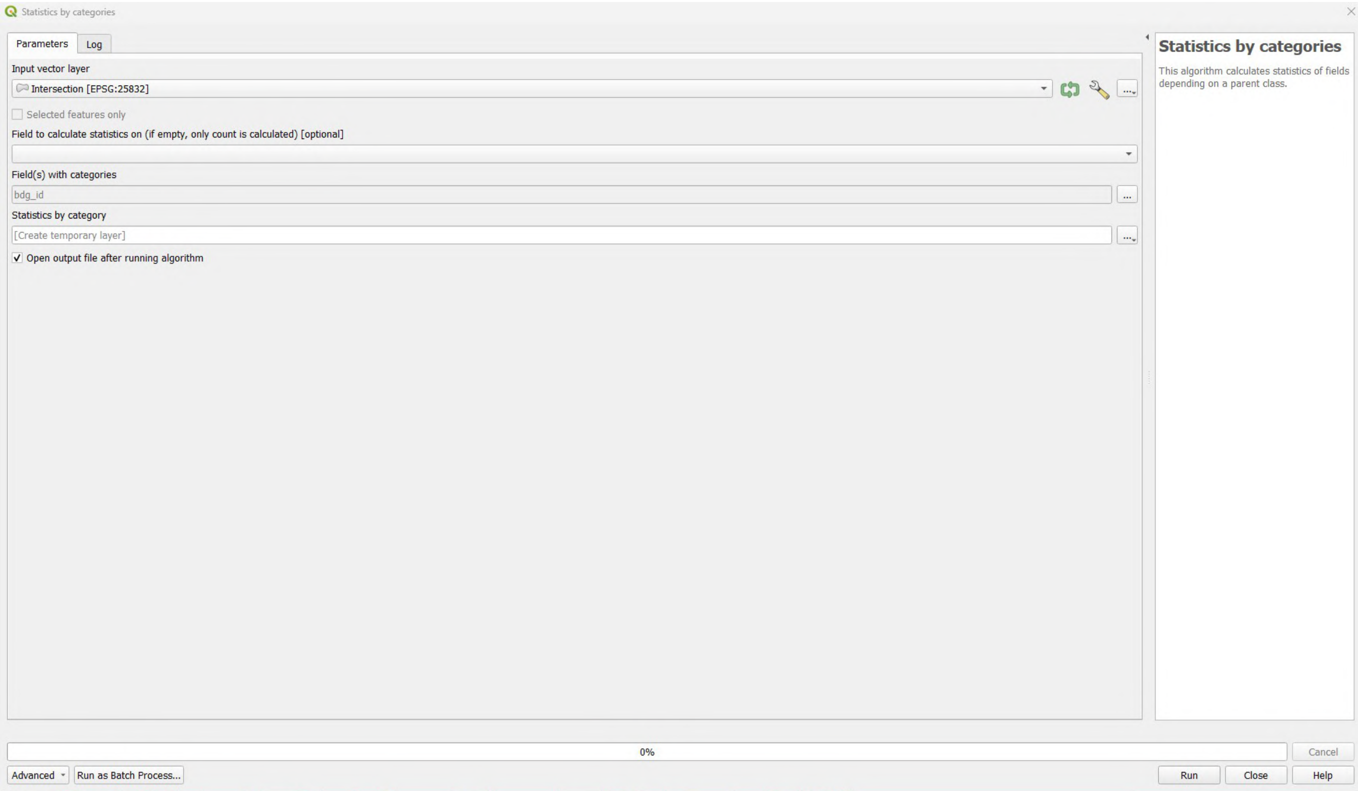
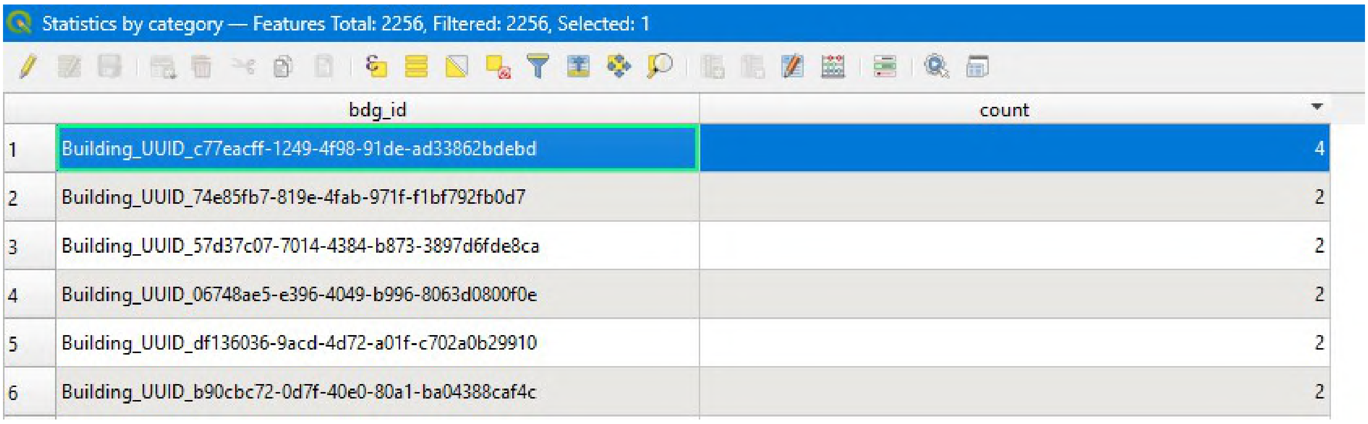


Figure 21: Checking the buildings only intersect with one census



	bdg_id	count
1	Building_UUID_c77eacff-1249-4f98-91de-ad33862bdebd	4
2	Building_UUID_74e85fb7-819e-4fab-971f-f1bf792fb0d7	2
3	Building_UUID_57d37c07-7014-4384-b873-3897d6fde8ca	2
4	Building_UUID_06748ae5-e396-4049-b996-8063d0800f0e	2
5	Building_UUID_df136036-9acd-4d72-a01f-c702a0b29910	2
6	Building_UUID_b90cbc72-0d7f-40e0-80a1-ba04388caf4c	2

Figure 22: 4 data entries found (2 scenarios and 2 overlapping cencus)

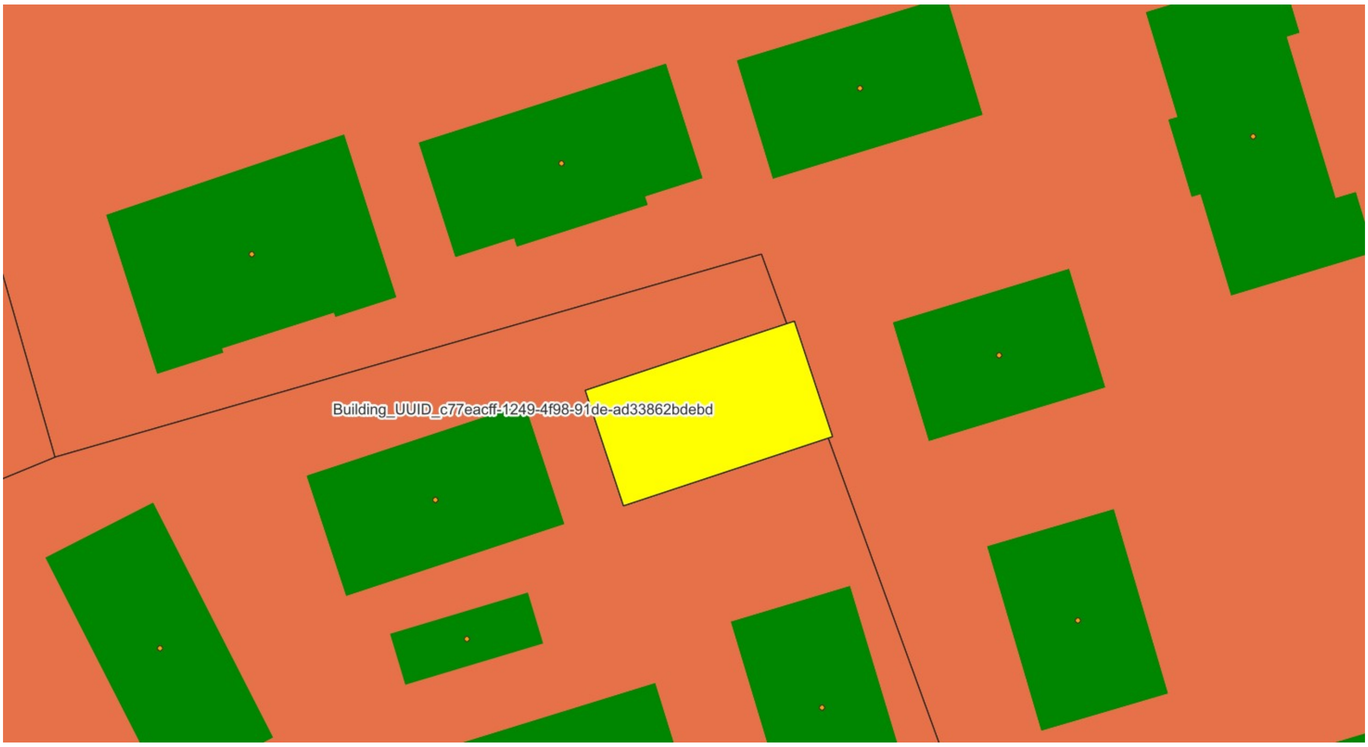


Figure 23: The overlapping building

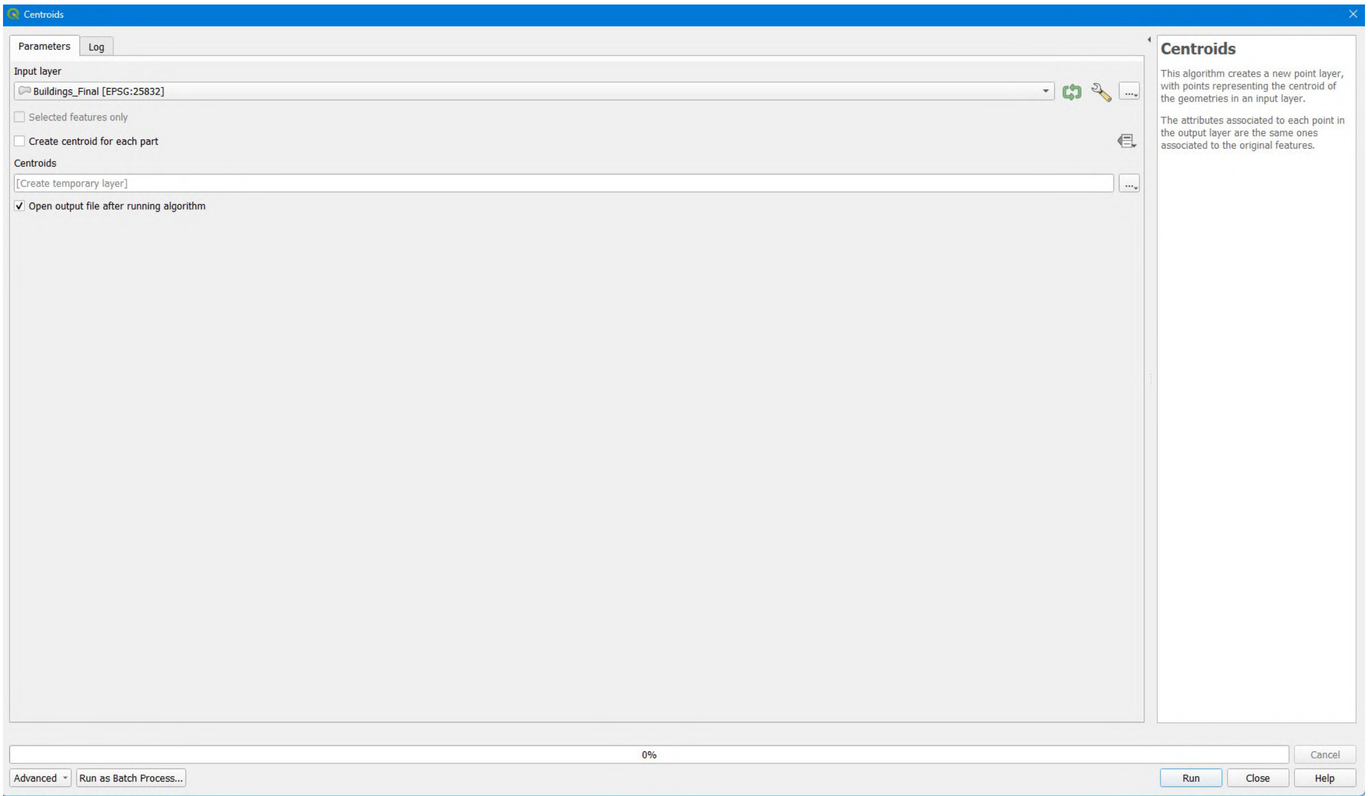


Figure 24: Using centroid function to assign the building to its belonged region

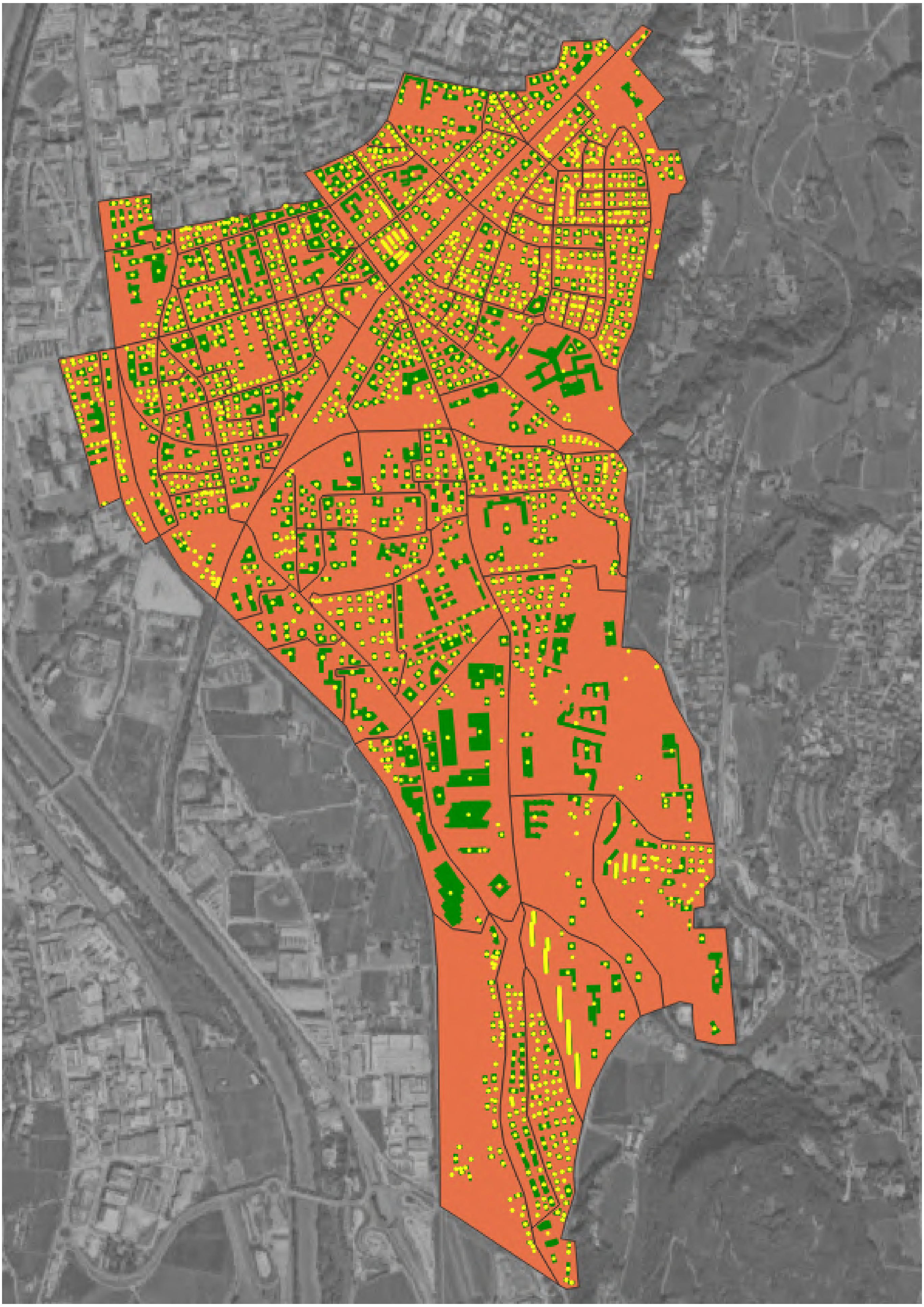


Figure 25: Centroids visualization

3 Part 3: 2D data presentation

3.1 Exercise 3.1

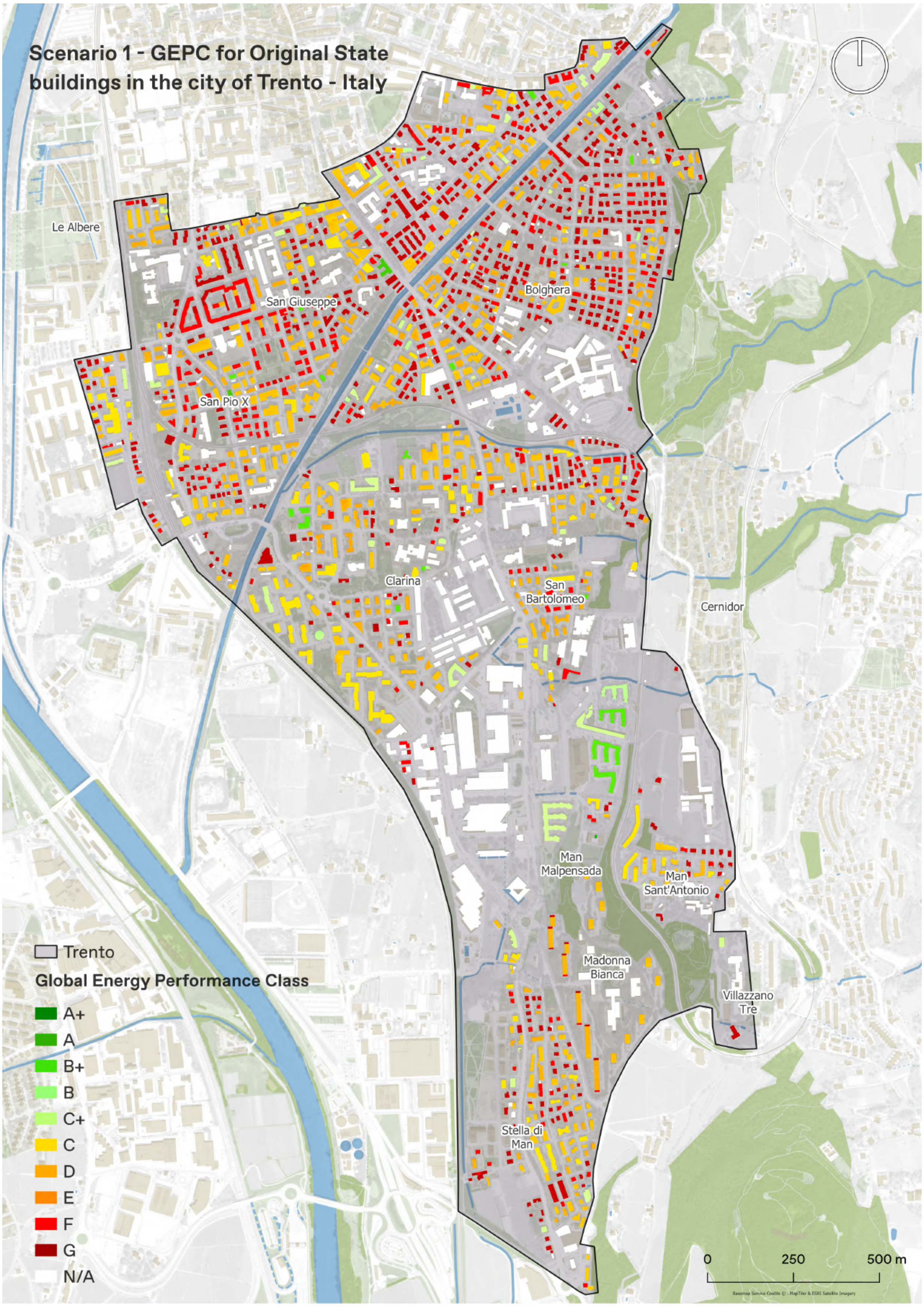


Figure 26: Color coded 2D map representing each building footprint according to its global-energy performance class in original state

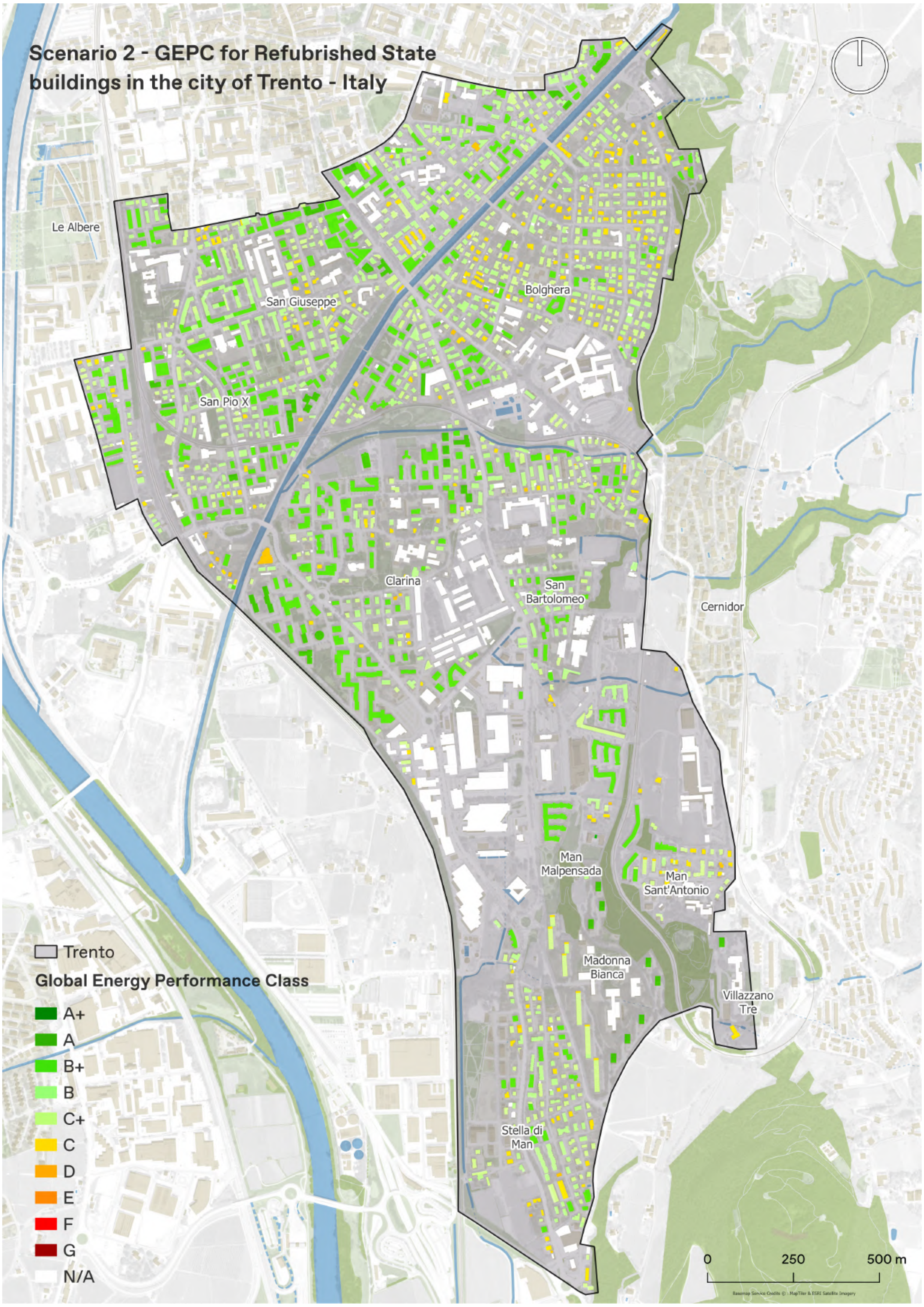


Figure 27: Color coded 2D map representing building footprint according to its global-energy performance class after it refurbished

3.2 Exercise 3.2

We chose to use a **bivariate choropleth map** to simultaneously represent both the "before" and "after" values of $Q_{H,p}$ for each census area. This allows for an easy comparison of energy demand changes in a single visual, without needing multiple maps. The bivariate color scheme clearly distinguishes areas where energy demand improved (high-to-low), worsened (low-to-high), or stayed constant (both high or both low). The approach efficiently highlights spatial patterns of refurbishment impact, making it easy for viewers to identify areas of significant change, while maintaining a clear and uncluttered map layout.

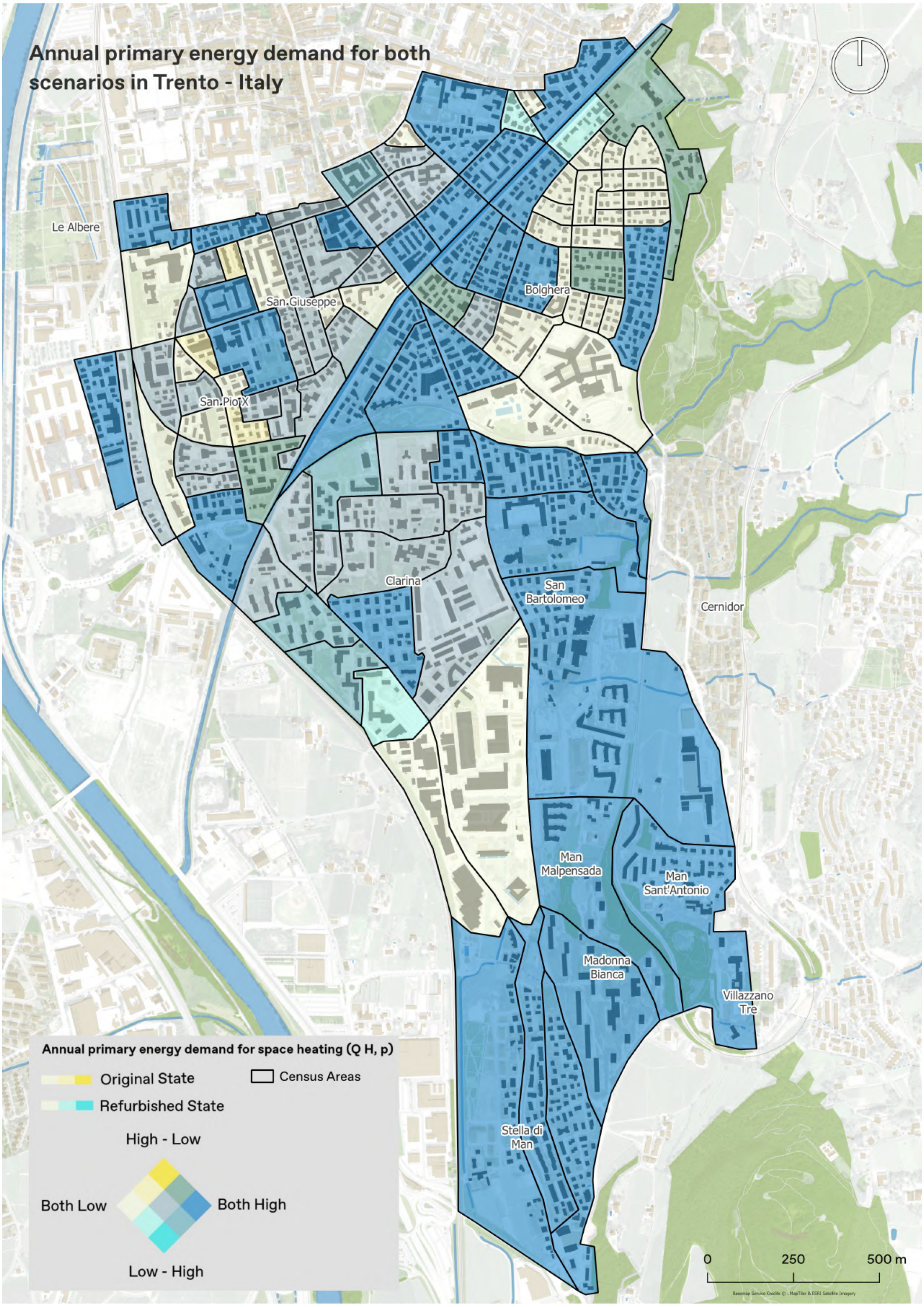


Figure 28: bivariate choropleth map showing $Q_{H,p}$ before and after for each census area

3.3 Exercise 3.3

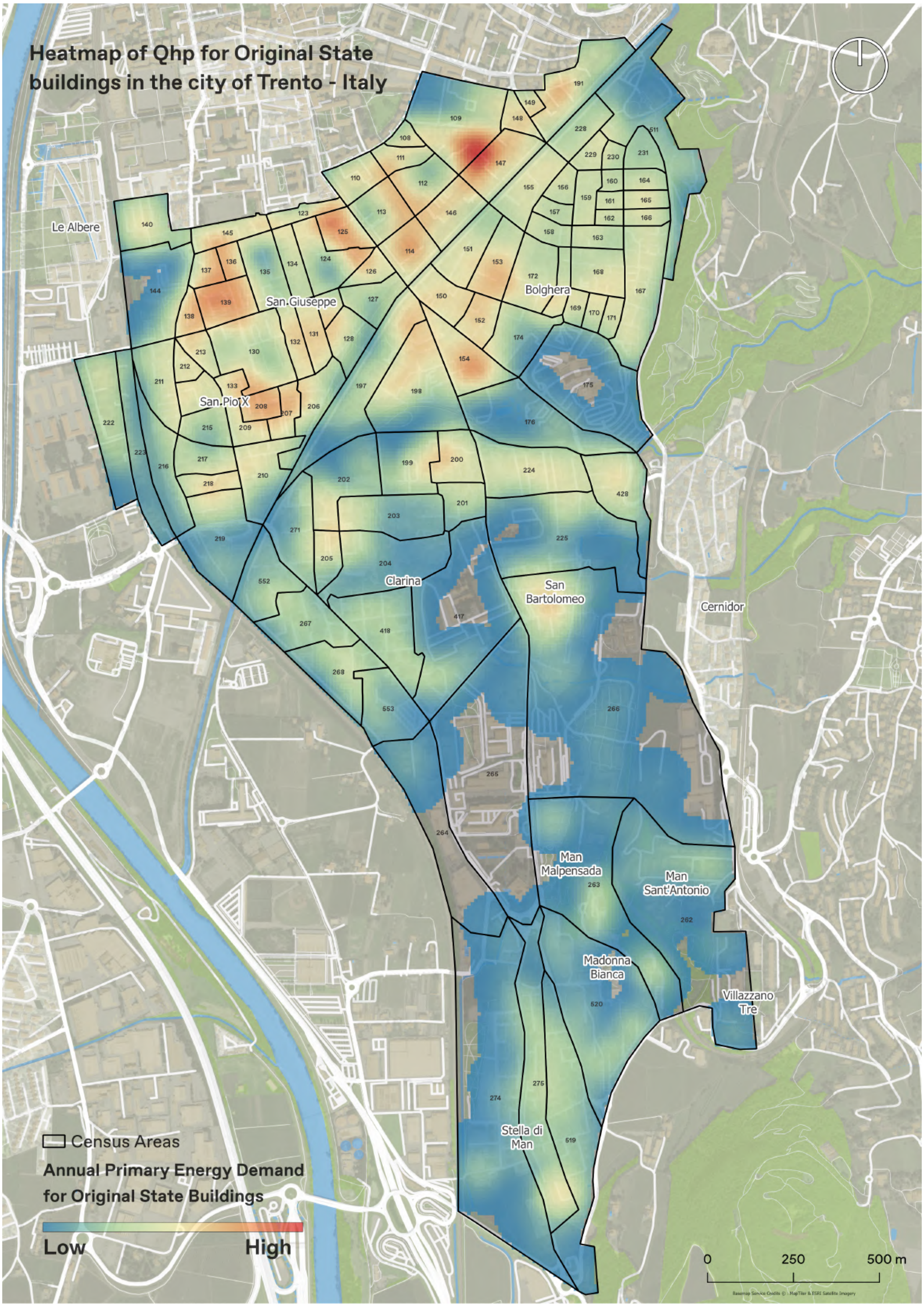


Figure 29: Heatmap showing aggregated values of $Q_{H,p}$ for each census area

Explanation of the process:

1. **Radius**: The algorithm calculates the density of points (centroids) within a 100-meter radius around each point. The larger the radius, the smoother the heatmap, as more points contribute to each pixel.

2. **Weight Field**: By using the $Q_{H,p}$ field as a weight, the heatmap reflects the varying energy demands across different buildings. Points with higher $Q_{H,p}$ values will have a stronger influence on the heatmap, highlighting areas with greater energy demand.

3. **Quartic Kernel**: This kernel shape gives more weight to points closer to the center of the radius and less weight to those farther away, producing a smoothly distributed heatmap.

4. **Pixel Size**: The heatmap will have a grid resolution of 10 meters, which provides enough detail to visualize energy demand variations while keeping the map readable.

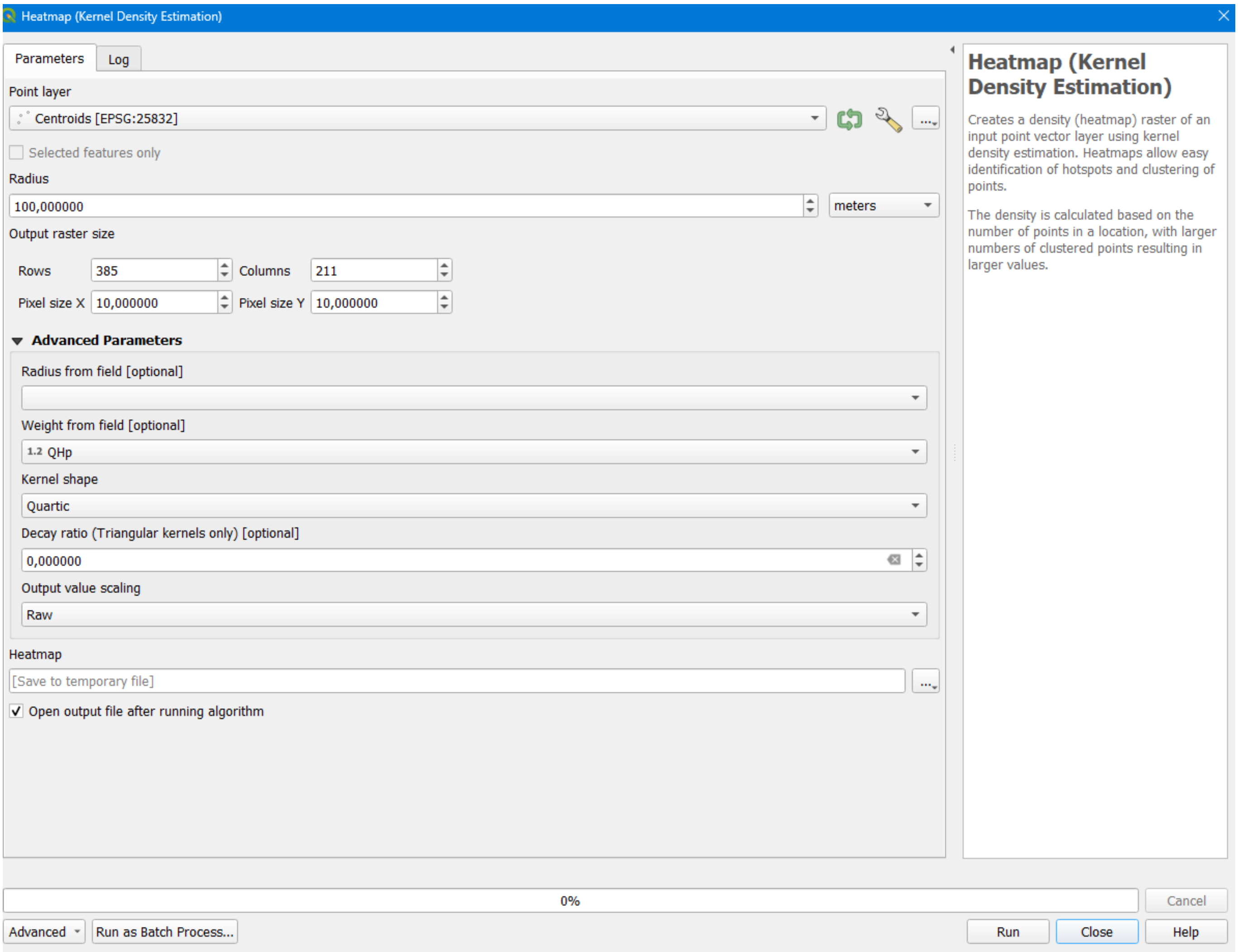


Figure 30: Heatmap settings

3.4 Exercise 3.4

Using a heat map to visualize the variable $Q_{H,p}$ (annual primary energy demand) can be effective in certain contexts, though it also has limitations. Heat maps are ideal for highlighting spatial intensity or clustering of values, making them useful for identifying areas with particularly high or low energy demand across a city. One key advantage is the ability to easily identify energy demand hotspots, which can inform urban planners about areas that may need targeted energy efficiency interventions or infrastructure upgrades. By using $Q_{H,p}$ as a weight field, the heat map directly reflects variations in energy consumption, making the visualization both meaningful and easy to interpret.

However, there are some disadvantages to this approach. Heat maps generalize spatial patterns, which can lead to a loss of precision, especially when dealing with building-level data like $Q_{H,p}$ The use of centroids and interpolation to create the heat map may blur boundaries between high and low demand areas, which could result in an oversimplified representation of energy use. Additionally, while the chosen grid resolution of 10 meters provides a reasonable balance between detail and readability, adjusting the radius too much could either obscure smaller patterns or oversimplify the map.

In conclusion, while a heat map is an effective tool for identifying general trends and hotspots in energy demand, it sacrifices some precision compared to other visualization methods like choropleth maps. It works well as an exploratory tool to highlight spatial trends but may not be the best choice when detailed, building-level analysis is required.

4 Part 4: 3D data presentation

4.1 Exercise 4.1

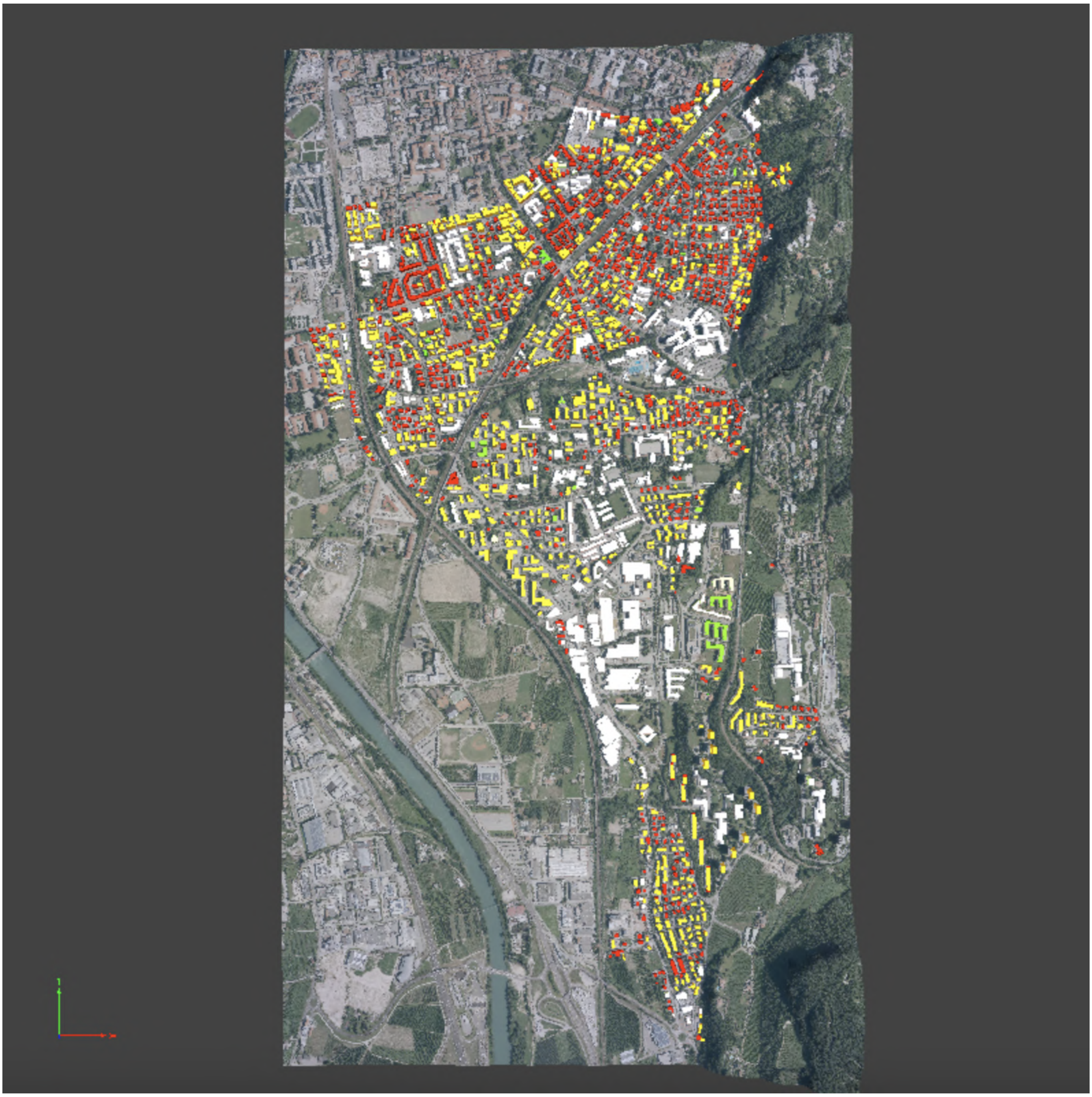


Figure 31: Output trento_3d.pdf

4.2 Exercise 4.2

The FME workflow is divided into 3 smaller task for better visualization and explanation:

1. Texurize 3D TIN (from part 1.4), section 4.2.1
2. Drape & create 3D geometries, section 4.2.2
3. Set extruded building appearance style, section 4.2.3

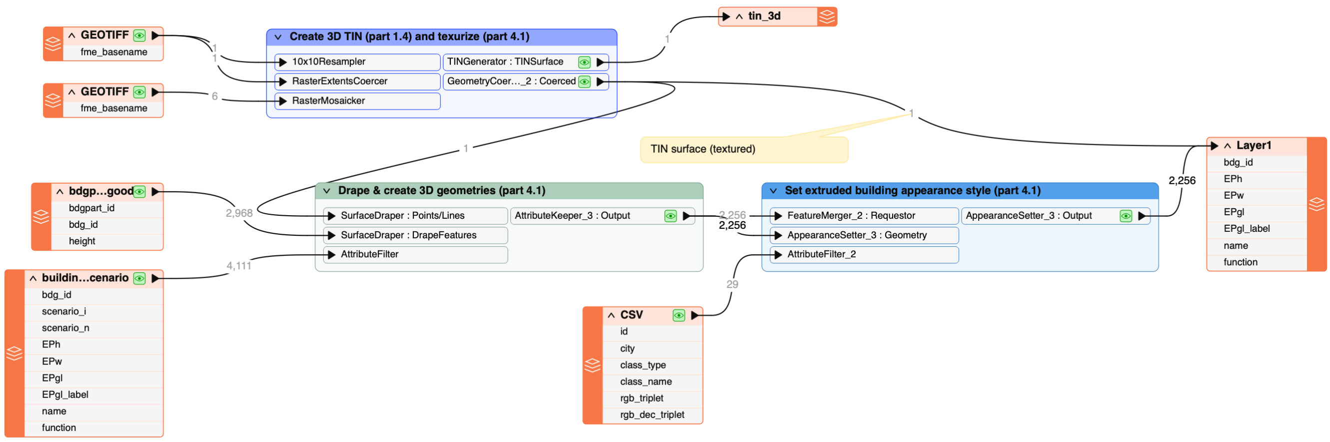


Figure 32: FME workflow overview

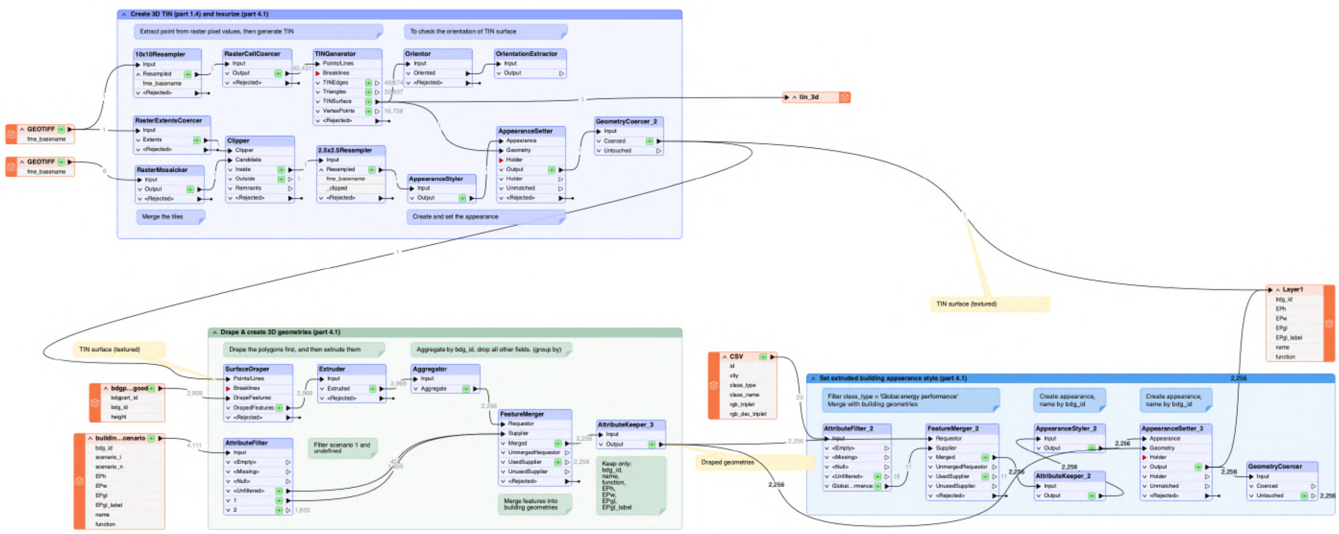


Figure 33: FME workflow overview (expanded)

4.2.1 Set TIN surface appearance style

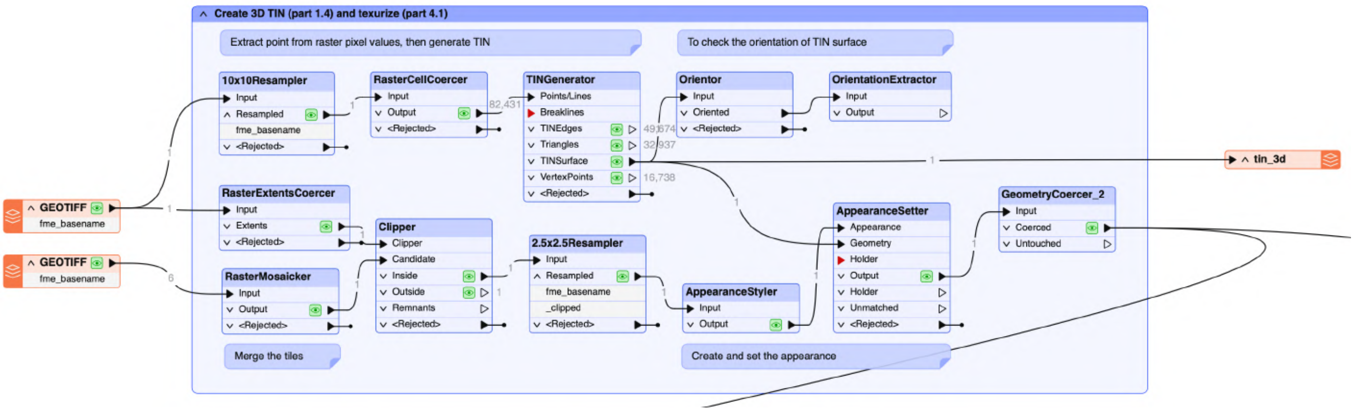


Figure 34: FME subtask 1, Create 3D TIN (part 1.4) and texurize

To obtain a texture within a matching extent, we used `RasterMosaicker` to combine the 6 orthophoto and clipped the combined raster with `Clipper`. We took the dtm used in part1.4 and exrracted its extent by using `RasterExtentsCoercer`. For texturing TIN surface with the orthophoto, we feed the clipped orthophoto into `AppearanceStyler` and then `AppearanceSetter` to obtain the appearance feature we wanted. (The orthophoto is resampled into 2.5m cell with `Resampler`)

4.2.2 Drape and create 3D geometries

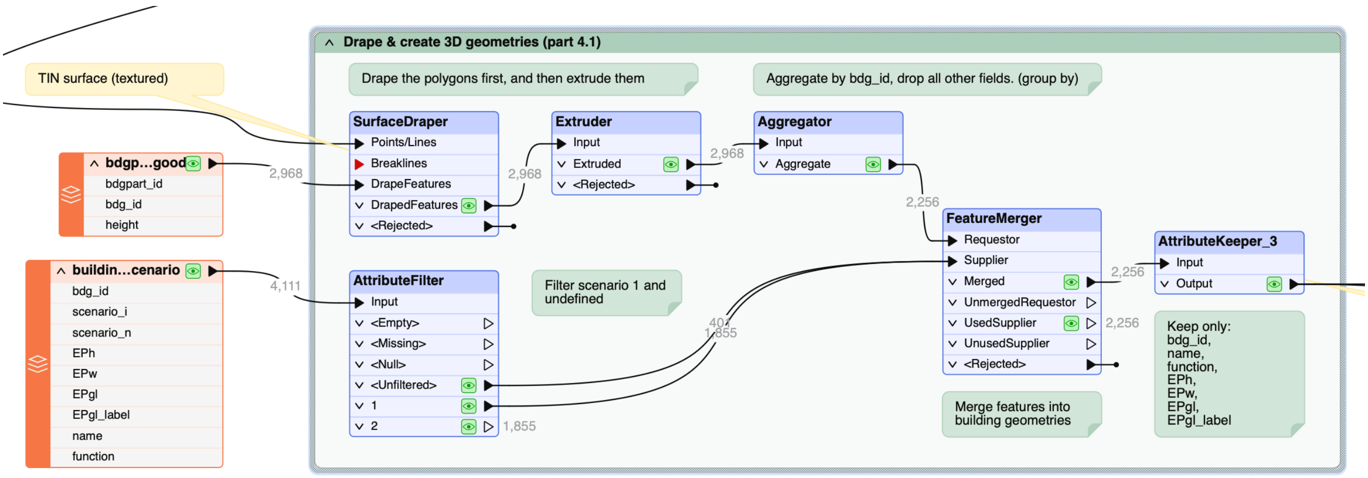


Figure 35: FME subtask 2, Drape & create 3D geometries

Here we take another 2 shapefile from previous exercises as input:

- **bdgpart_good**
- **buildings**

For this subtask 2, we use `SurfaceDraper` and `Extruder` to get the draped solid. Since we cannot take a 3D solid brep as input of `SurfaceDraper`, we must first drape the polygons from bdgpart_good and then extrude them. After the extrusion being generated, we use `Aggregator` with `Group Processing` enabled to aggregate building parts into buildings. We then joined the attributes in buildings layer with `FeatureMerger`.

4.2.3 Set extruded building appearance style

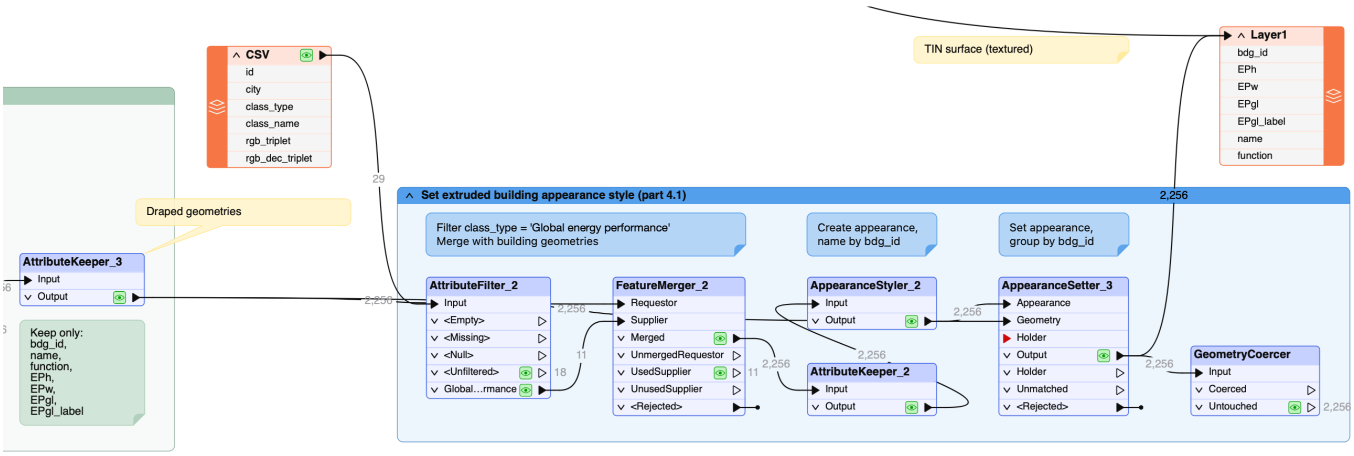


Figure 36: FME subtask 3, Set extruded building appearance style

Another csv file **energy_classes** is used here to set appearance style correspondingly.

The energy classes csv must be filtered to extract the global energy performance color scheme, with filter expression: `class_type = 'Global energy performance'`. In addition, this color scheme should be join into buildings to make the color code attached to bdg_id for further usage (energy label as primary key and class_name as foreign key).

As we did in previous exercise, here we use `AppearanceStyler` and `AppearanceSetter` again.
The following steps have to be done to prevent error while applying color to buildings:

- To set appearance correctly, name appearance by bdg_id.
- In `AppearanceSetter`, enable `Group Processing` and set it to be group by bdg_id.

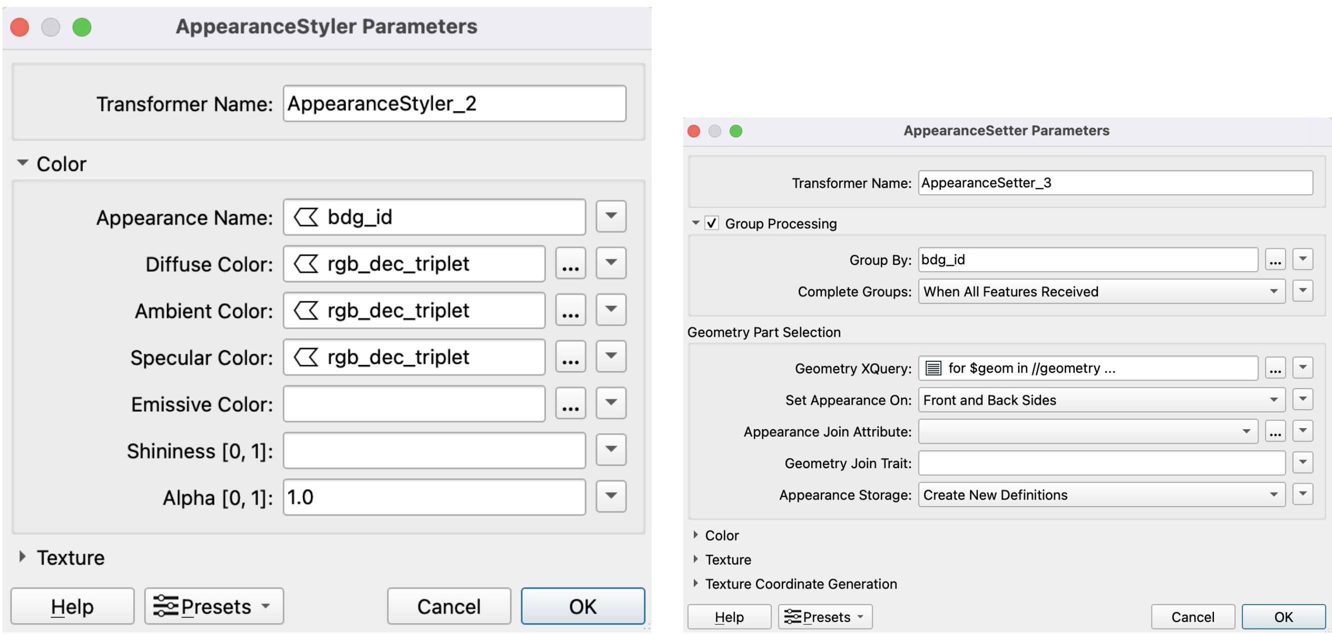


Figure 37: AppearanceStyler and AppearanceSetter parameters