Report

# 1. UNSW Lower Campus

## 1.1 Datasets

|  |  |  |  |
| --- | --- | --- | --- |
| Data | Size | #Voxels | Description |
| dtmbot.xyz | 9.22 GB | 641,624,355 | A terrain with holes, in which the buildings fit |
| tree.xyz | 906 MB | 59,640,000 | Tree in lower campus |
| bld1-54.xyz (except for 26 and 46) | 3.52GB | 241,613,693 in total and 4,646,418 per building | 52 buildings in lower campus |
| be.xyz | 249 MB | 17,460,029 | Built Environment (H13) |
| blockhouse.xyz | 45.4 MB | 3,392,202 | Blockhouse (G6) |
| dalton.xyz | 25.5 MB | 1,887,512 | Dalton (F12) |
| quadrangle.xyz | 43.9MB | 3,161,733 | Quadrangle (E15) |
| roundhouse.xyz | 79.9MB | 6,037,174 | Roundhouse (E6) |
| scithe.xyz | 17.2MB | 1,231,821 | Science Theatre (F13) |

Note that:

* For GIS-based voxels, its resolution is 20cm. All voxels are recorded in same relative coordinate with offset (336000, 6245250, 20).
* For BIM-based voxels, its resolution is 10 cm. Each building is in its own relative coordinate with MINXYZ.
  + For be.xyz, the offset is (336300, 6245507, 25).
  + For blockhouse.xyz, the offset is (336042, 6245613, 27).
  + For dalton.xyz, the offset is (336305, 6245569, 29).
  + For quadrangle.xyz, the offset is (336409, 6245580, 31).
  + For roundhouse.xyz, the offset is (336047, 6245651, 25).
  + For scithe.xyz, the offset is (336325, 6245582, 28).

## 1.2 Database Schema

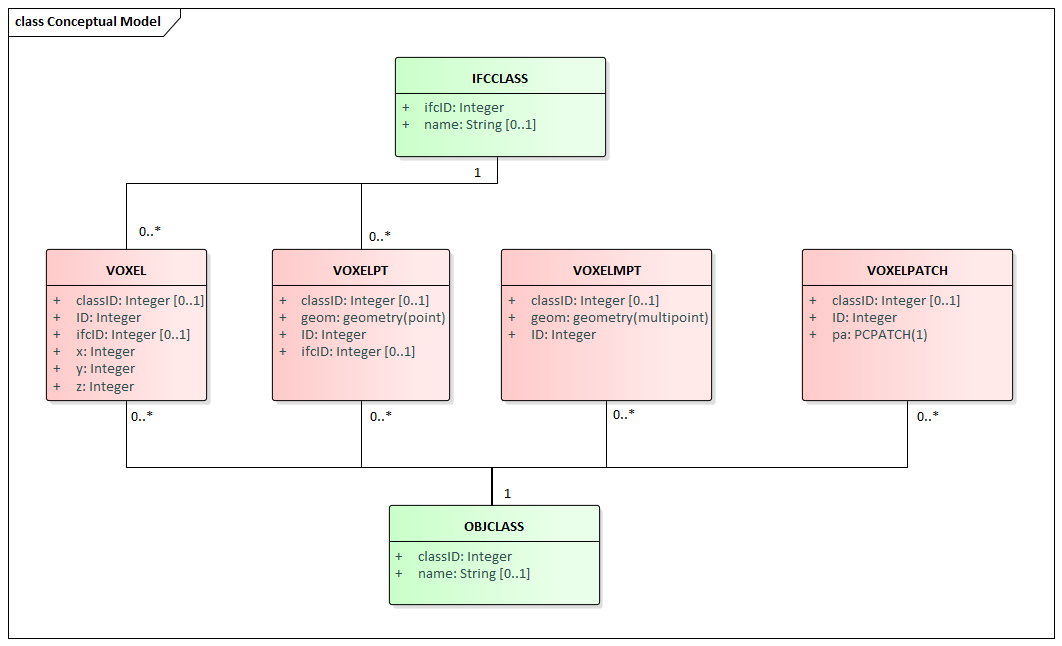


Figure 1. Conceptual Model

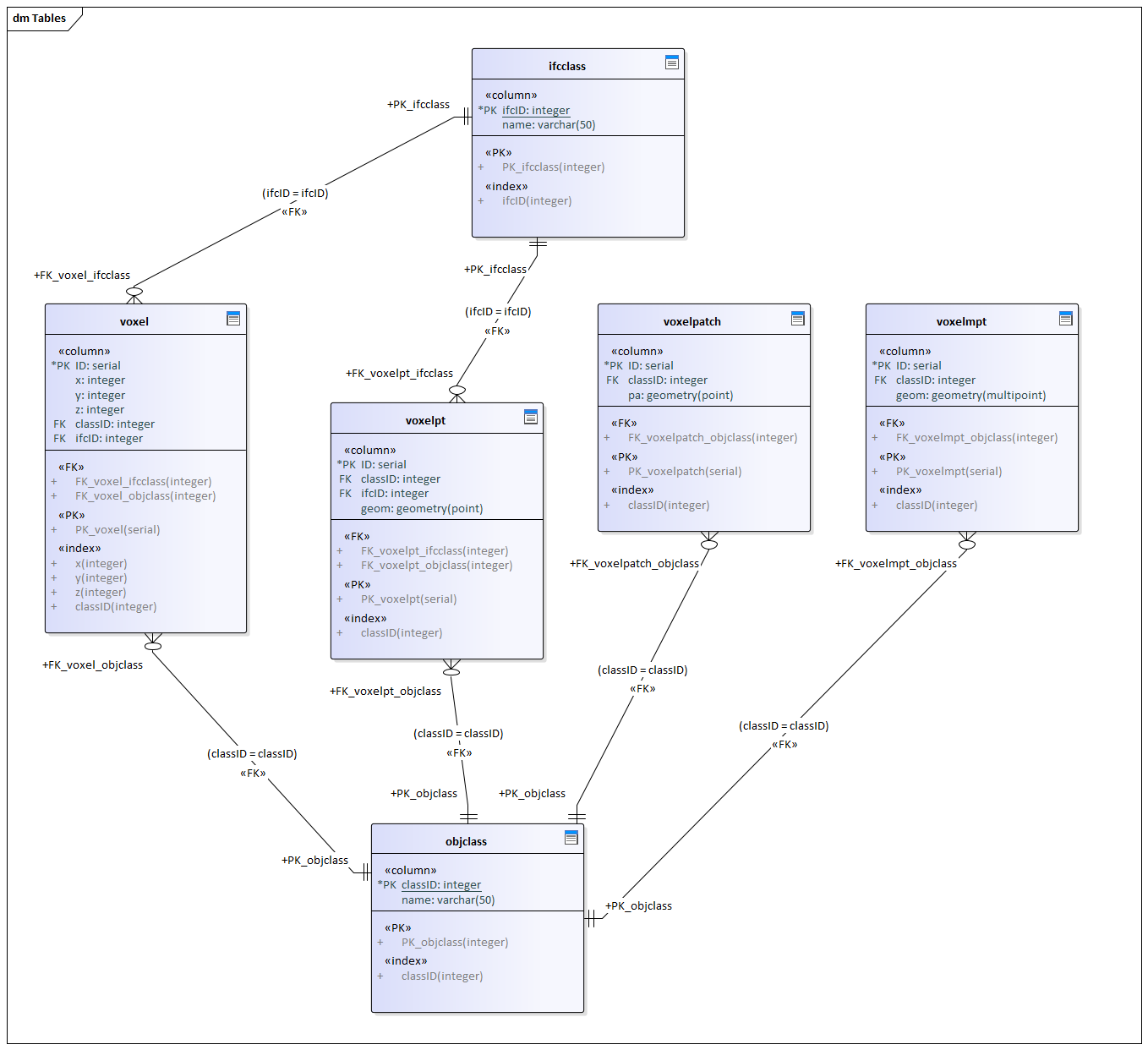


Figure 2. Physical Model

## 1.3 Data Layout in PostgreSQL

Four Main Tables for voxel storage:

|  |  |  |  |
| --- | --- | --- | --- |
| Table | Columns | Geometry | Description |
| voxel | id, x, y, z, classid, ifcid | N/A | One voxel per row/record. |
|  | | | |
| voxelpt | id, classid, ifcid, geom | POINT(x,y,z) | One voxel per row/record with geometry. |
|  | | | |
| voxelmpt | id, classid, geom | MULTIPOINT | One building per row/record.  20,000,000 point per row/record for “tree” and “dtm”. |
|  | | | |
| voxelpatch | id, classid, pa | PCPATCH(1) | One building per row/record.  20,000,000 point per row/record for “tree” and “dtm”. |
|  | | | |

Two Semantic Tables for IFC and class info:

|  |  |  |
| --- | --- | --- |
| Table | Columns | Description |
| ifcclass | ifcid, name | 26 IFC class with corresponding name |
|  | | |
| objclass | classid, name | classid 1-54 (except for 26 and 46) are building ID.  classid 55 is tree ID.  classid 56 is dtmbot ID. |
|  | | |

# 2. QGIS Visualization

The data query is processed using a HP laptop. Its processor is Intel(R) Core (TM) i7-7600 CPU @ 2.80GHz and its installed memory is 16.0 GB. Its operating system is 64-bit Windows 10. And the test is performed on PostgreSQL (11.2), PostGIS (2.5.2), and QGIS (3.6.1).

## 2.1 Sample Visualization

Considering bld52 (we don’t have building name at this moment), 29.6MB and 1,881,847 voxels, we extract “geom” first, and then convert its coordinate into EPSG:28356.

1. **SELECT** ST\_MakePoint(336000+ST\_X(geom)\*0.2, 6245250+ST\_Y(geom)\*0.2, 20+ST\_Z(geom)\*0.2) **AS** geom
2. **FROM** voxelpt
3. **WHERE** classid=52;

Figure 3 shows how to execute query in QGIS.

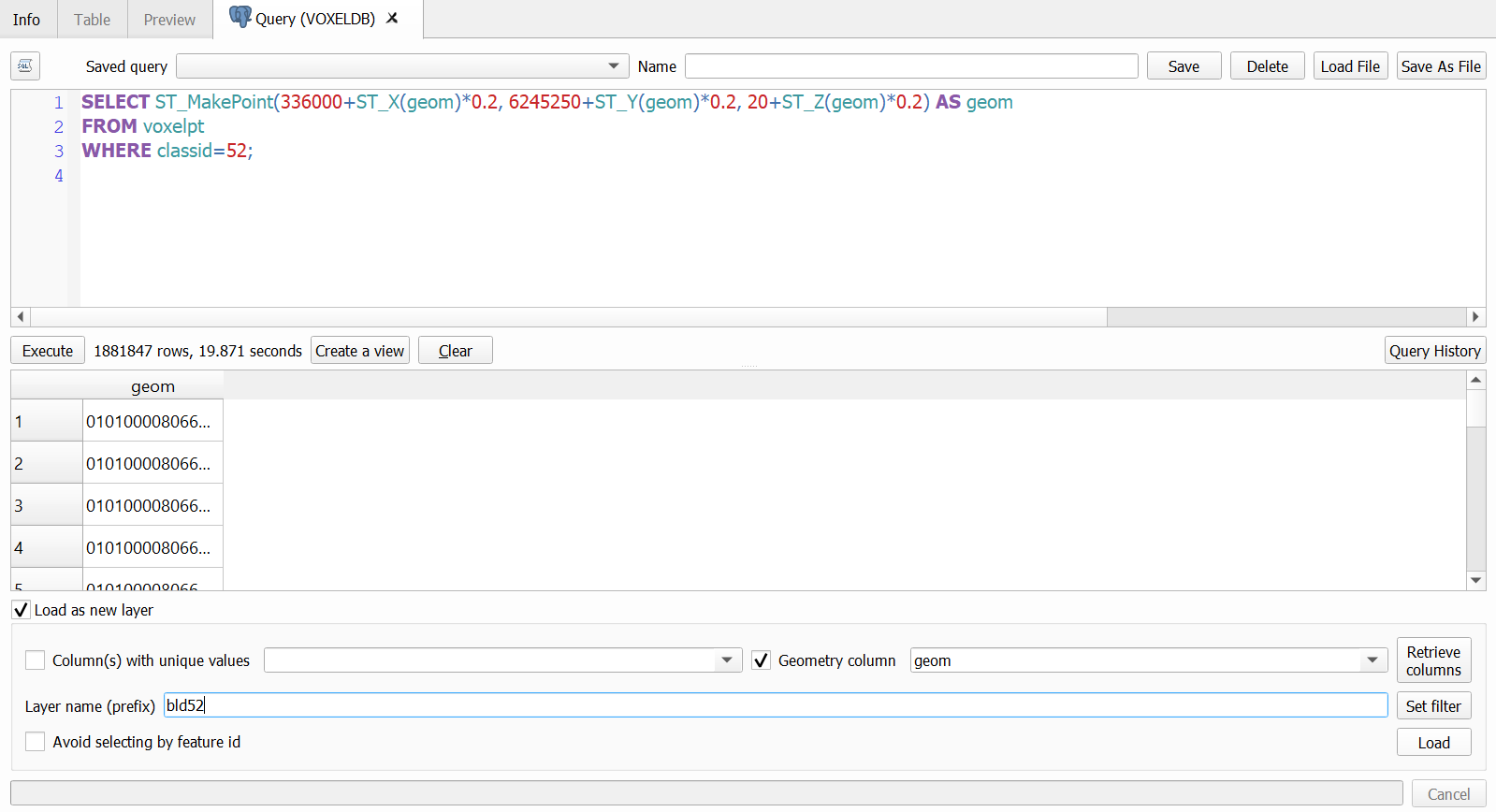


Figure 3. Query execution in QGIS

Note that, loading the above query result as a new layer in QGIS may take several minutes and 3D view is as well. Once choosing 3D view, please keep an eye out for your GPU and memory changes, if you crash, kill the task or stop doing the work at hand and continue to wait patiently. If not necessary or you are not confident in your PC, don’t try 3D view.



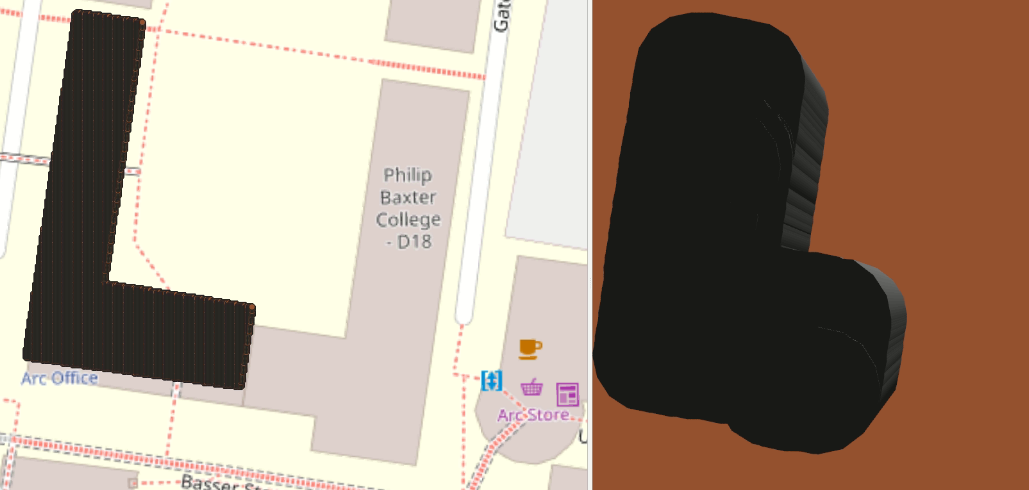


Figure 3. 2D and 3D visualization of “bld52” in QGIS

# 3. Object Matching between Different Data Source

## 3.1 Checking Data Info

|  |  |
| --- | --- |
| Data | #Voxels |
| bld1-54 | 241,613,693 in total and 4,646,418 per building |
| tree | 59,640,000 |
| dtmbot | 641,624,355 |
| Ifcid is not null | 33,170,471 |
| be | 17,460,029 |
| blockhouse | 3,392,202 |
| dalton | 1,887,512 |
| quadrangle | 3,161,733 |
| roundhouse | 6,037,174 |
| scithe | 1,231,821 |

## 3.2 Assign Temporary classID for IFC buildings

In table “voxel” and “voxelpt”, GIS data occupied 944,110,209 rows.

At this moment, we assume the 6 IFC models with classID 57, 58, 59, 60, 61, and 62, respectively.

1. \COPY voxel(x, y, z, ifcID) **FROM** 'C:\Users\z5039792\Documents\Vox3DMod\data\bim\BE\classmodel.xyz' DELIMITER ' ';
2. **UPDATE** voxel **SET** classID=57 **WHERE** classid **IS** NULL;
3. \COPY voxel(x, y, z, ifcID) **FROM** 'C:\Users\z5039792\Documents\Vox3DMod\data\bim\BlockHouse\classmodel.xyz' DELIMITER ' ';
4. **UPDATE** voxel **SET** classID=58 **WHERE** classid **IS** NULL;
5. \COPY voxel(x, y, z, ifcID) **FROM** 'C:\Users\z5039792\Documents\Vox3DMod\data\bim\Dalton\classmodel.xyz' DELIMITER ' ';
6. **UPDATE** voxel **SET** classID=59 **WHERE** classid **IS** NULL;
7. \COPY voxel(x, y, z, ifcID) **FROM** 'C:\Users\z5039792\Documents\Vox3DMod\data\bim\Quadrangle\classmodel.xyz' DELIMITER ' ';
8. **UPDATE** voxel **SET** classID=60 **WHERE** classid **IS** NULL;
9. \COPY voxel(x, y, z, ifcID) **FROM** 'C:\Users\z5039792\Documents\Vox3DMod\data\bim\Roundhouse\classmodel.xyz' DELIMITER ' ';
10. **UPDATE** voxel **SET** classID=61 **WHERE** classid **IS** NULL;
11. \COPY voxel(x, y, z, ifcID) **FROM** 'C:\Users\z5039792\Documents\Vox3DMod\data\bim\SciThe\classmodel.xyz' DELIMITER ' ';
12. **UPDATE** voxel **SET** classID=62 **WHERE** classid **IS** NULL;

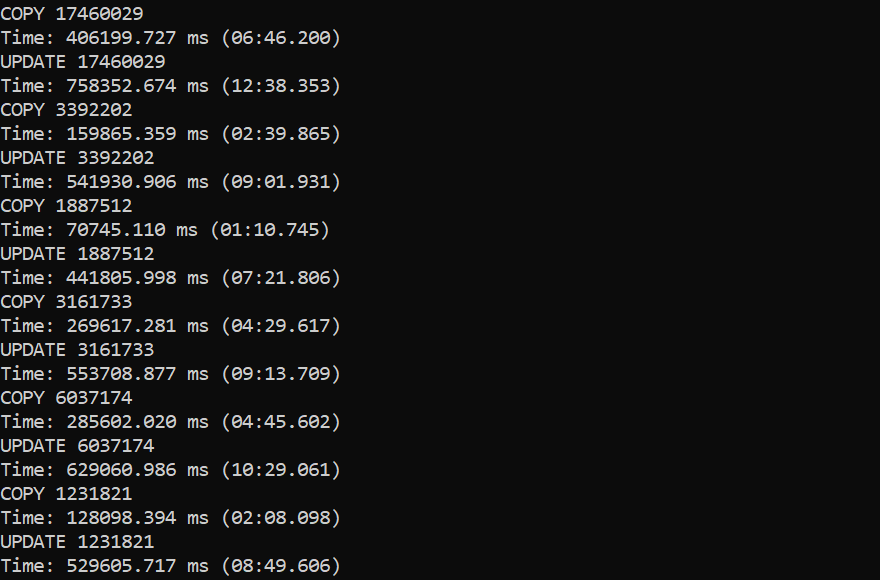


Figure 4. Log info for IFC data importing

## 3.3 Update classID for IFC buildings

For BE building, through computing bld19 and its the MAX & MIN (x,y) range in EPSG:28356 CRS, it is easy to find that they are in high probability the same building.

1. **SELECT** **MAX**(x)\*0.1+336300 **AS** maxx, **MIN**(x)\*0.1+336300 **AS** minx, **MAX**(y)\*0.1+6245507 **AS** maxy, **MIN**(y)\*0.1+6245507 **AS** miny
2. **FROM** voxel
3. **WHERE** classid=57;
5. **SELECT** **MAX**(x)\*0.2+336000 **AS** maxX, **MIN**(x)\*0.2+336000 **AS** minX, **MAX**(y)\*0.2+6245250 **AS** maxY, **MIN**(y)\*0.2+6245250 **AS** minY
6. **FROM** voxel
7. **WHERE** classID=19;





Figure 5. (x,y) range for BE in EPSG:28356 CRS

Then, visualizing above two buildings in CloudCompare, it looks similar, at least in shape.

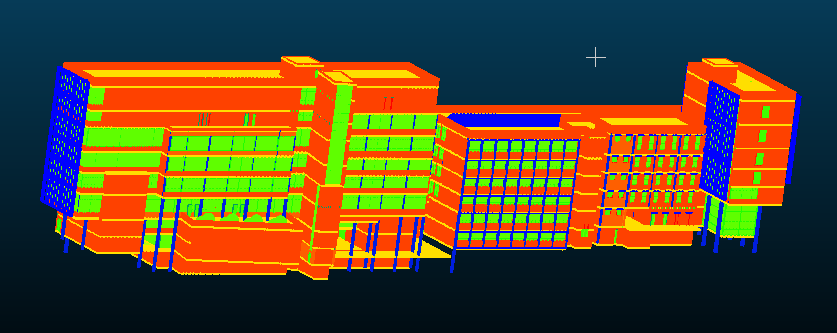


Figure 6. BE building in CloudCompare

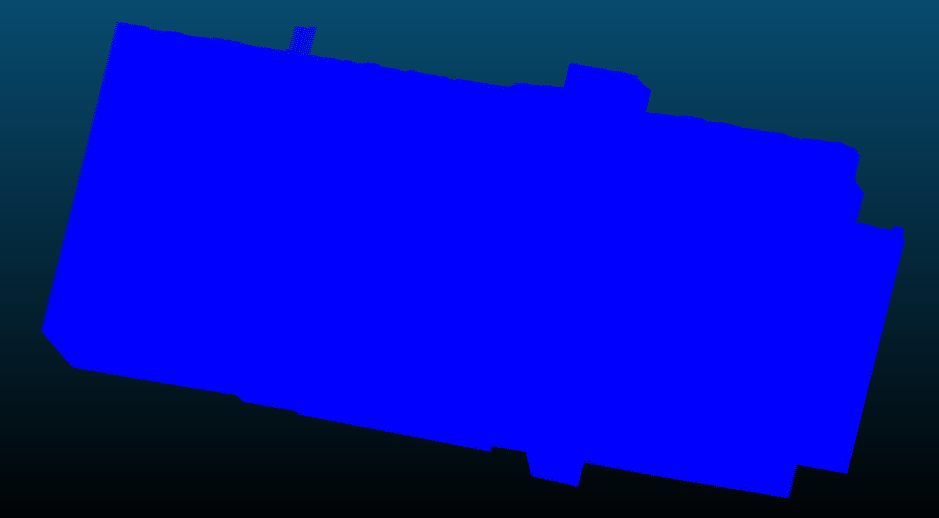
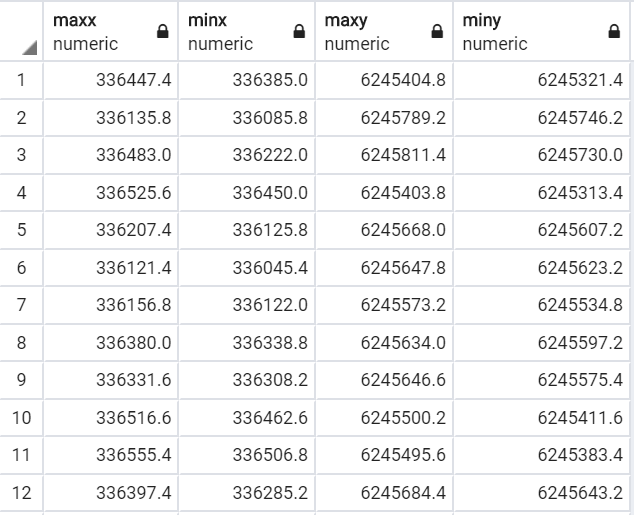


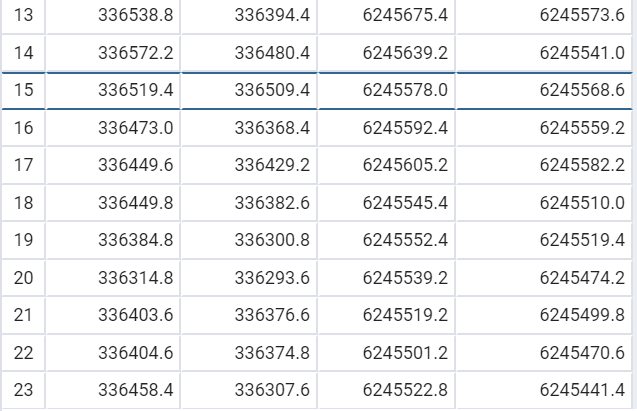
Figure 7. bld19 in CloudCompare

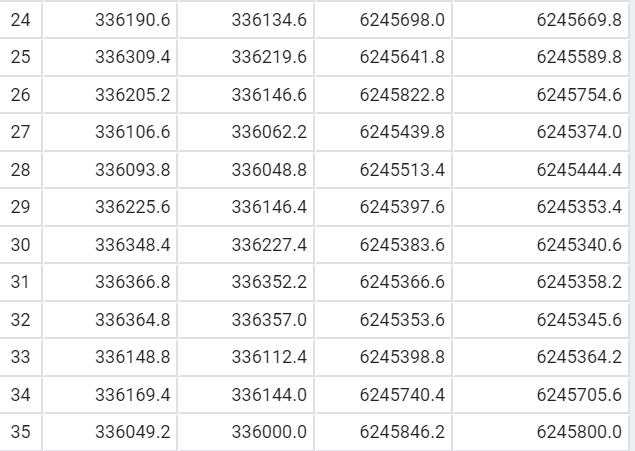
For Blockhouse, Dalton, Quadrangle, Roundhouse, SciThe buildings, calculating (x,y) range for all buildings with classID<=54. And then retrieve same range for above 5 buildings to do matching.

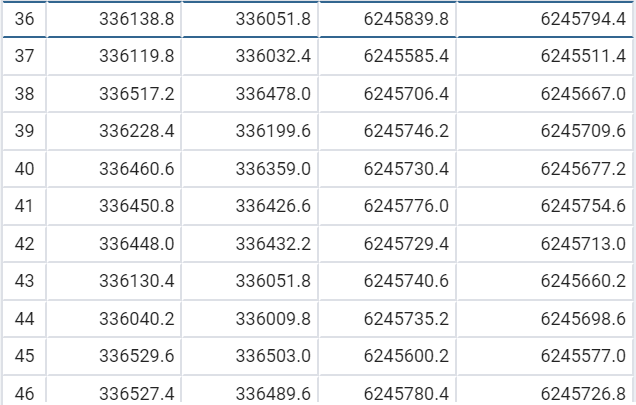
1. **SELECT** **MAX**(x)\*0.1+336042 **AS** maxx, **MIN**(x)\*0.1+336042 **AS** minx, **MAX**(y)\*0.1+6245613 **AS** maxy, **MIN**(y)\*0.1+6245613 **AS** miny
2. **FROM** voxel
3. **WHERE** classid=58;
5. **SELECT** **MAX**(x)\*0.1+336305 **AS** maxx, **MIN**(x)\*0.1+336305 **AS** minx, **MAX**(y)\*0.1+6245569 **AS** maxy, **MIN**(y)\*0.1+6245569 **AS** miny
6. **FROM** voxel
7. **WHERE** classid=59;
9. **SELECT** **MAX**(x)\*0.1+336409 **AS** maxx, **MIN**(x)\*0.1+336409 **AS** minx, **MAX**(y)\*0.1+6245580 **AS** maxy, **MIN**(y)\*0.1+6245580 **AS** miny
10. **FROM** voxel
11. **WHERE** classid=60;
13. **SELECT** **MAX**(x)\*0.1+336047 **AS** maxx, **MIN**(x)\*0.1+336047 **AS** minx, **MAX**(y)\*0.1+6245651 **AS** maxy, **MIN**(y)\*0.1+6245651 **AS** miny
14. **FROM** voxel
15. **WHERE** classid=61;
17. **SELECT** **MAX**(x)\*0.1+336325 **AS** maxx, **MIN**(x)\*0.1+336325 **AS** minx, **MAX**(y)\*0.1+6245582 **AS** maxy, **MIN**(y)\*0.1+6245582 **AS** miny
18. **FROM** voxel
19. **WHERE** classid=62;
21. **SELECT** **MAX**(x)\*0.2+336000 **AS** maxX, **MIN**(x)\*0.2+336000 **AS** minX, **MAX**(y)\*0.2+6245250 **AS** maxY, **MIN**(y)\*0.2+6245250 **AS** minY
22. **FROM** voxel
23. **WHERE** classID<=54
24. **GROUP** **BY** classID;

|  |  |
| --- | --- |
| **Building Name** | **Range in (x,y) EPSG:28356 CRS** |
| **Blockhouse** |  |
| **Dalton** |  |
| **Quadrangle** |  |
| **Roundhouse** |  |
| **SciThe** |  |









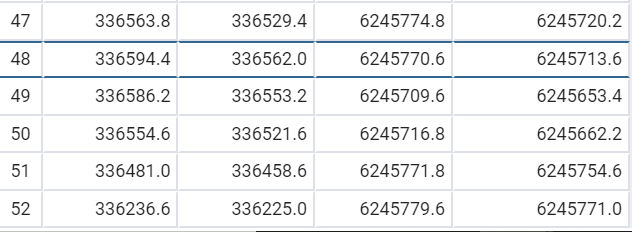


Figure 8. (x,y) range for all 52 building

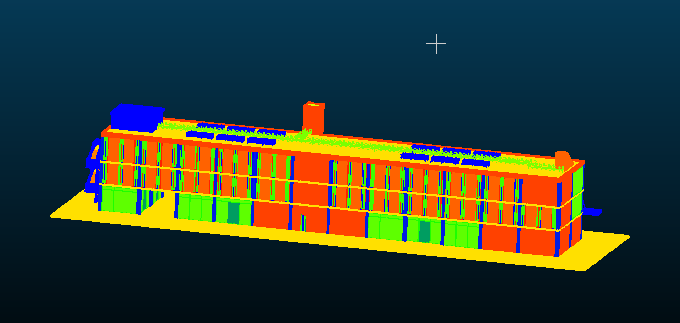
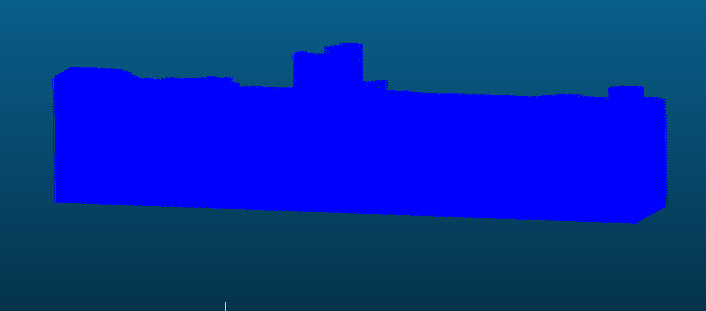
 

Figure 9. Blockhouse

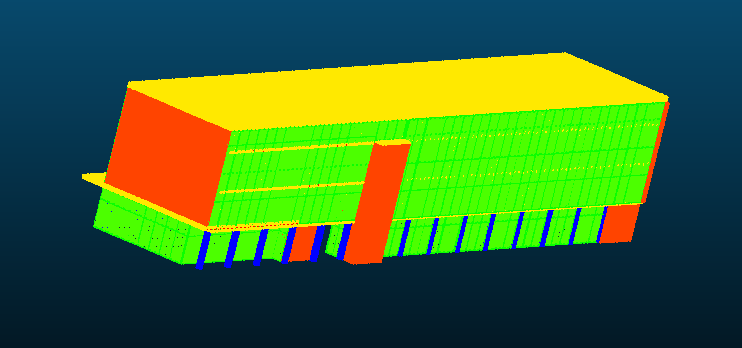
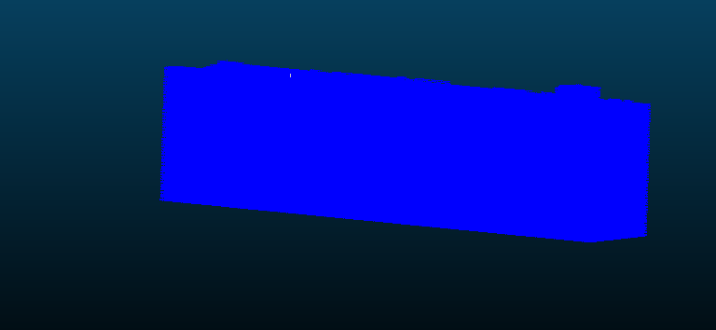
 

Figure 10. Dalton

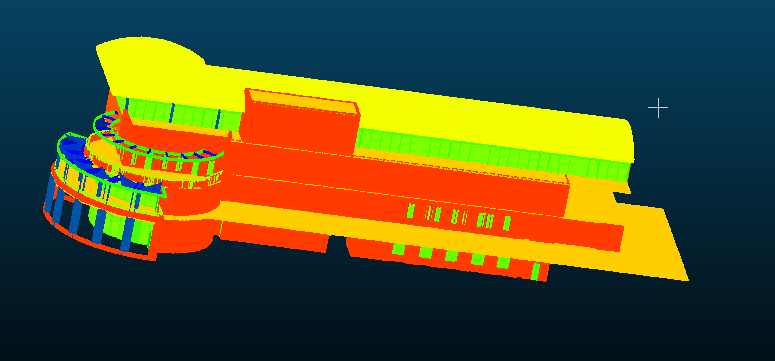
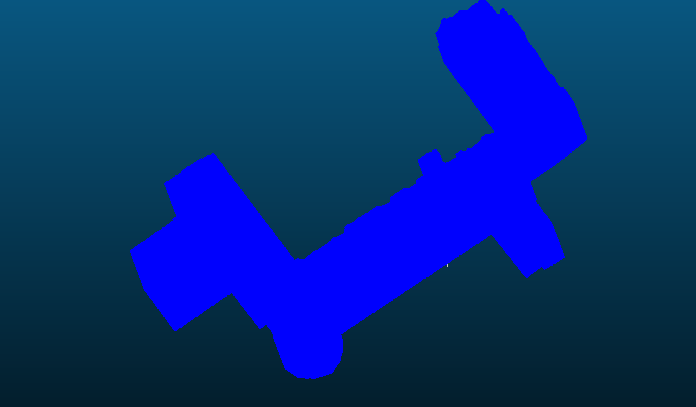
 

Figure 11. Quadrangle

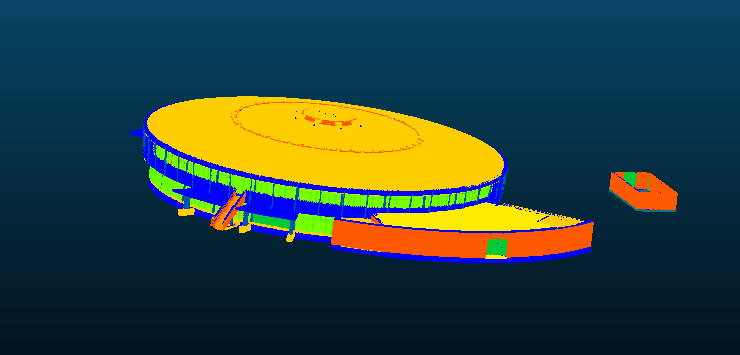
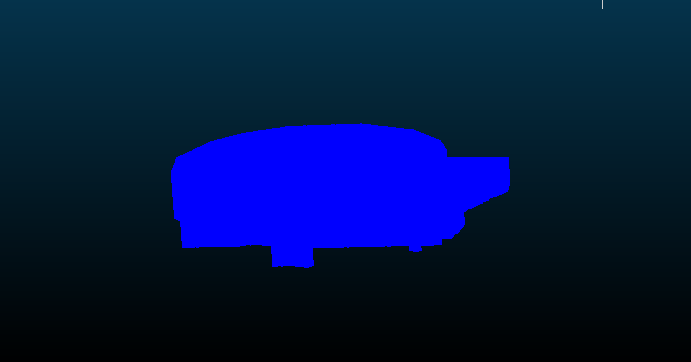
 

Figure 12. Roundhouse

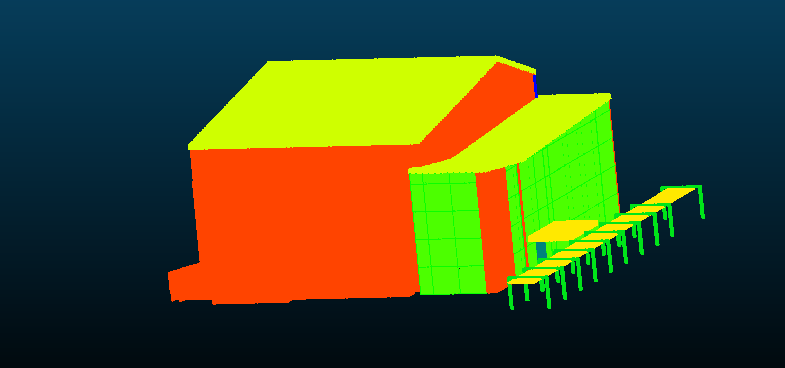
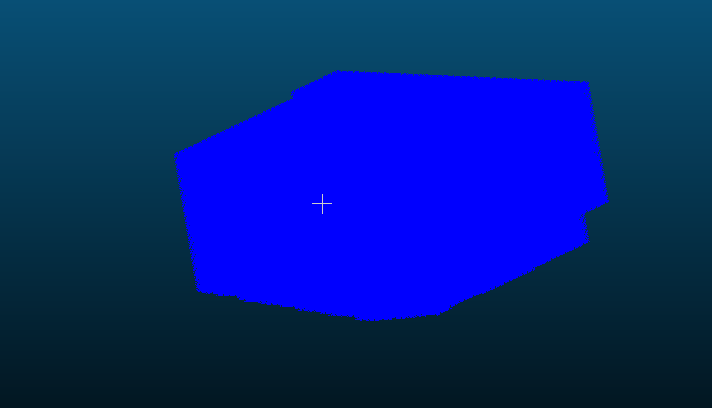
 

Figure 13. Science Theatre

In summary, the corresponding bld are list in below table:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Id | Name | Num | Old classID | New classID |
| Bld19 | Built Environment | H13 | 57 | 19 |
| Bld6 | Blockhouse | G6 | 58 | 6 |
| Bld9 | Dalton | F12 | 59 | 9 |
| Bld13 | Quadrangle | E15 | 60 | 13 |
| Bld44 | Roundhouse | E6 | 61 | 44 |
| Bld8 | Science Theatre | F13 | 62 | 8 |

Update the 6 buildings:

1. **UPDATE** voxel **SET** classID=19 **WHERE** classid=57;
2. **UPDATE** voxel **SET** classID=6 **WHERE** classid=58;
3. **UPDATE** voxel **SET** classID=9 **WHERE** classid=59;
4. **UPDATE** voxel **SET** classID=13 **WHERE** classid=60;
5. **UPDATE** voxel **SET** classID=44 **WHERE** classid=61;
6. **UPDATE** voxel **SET** classID=8 **WHERE** classid=62;

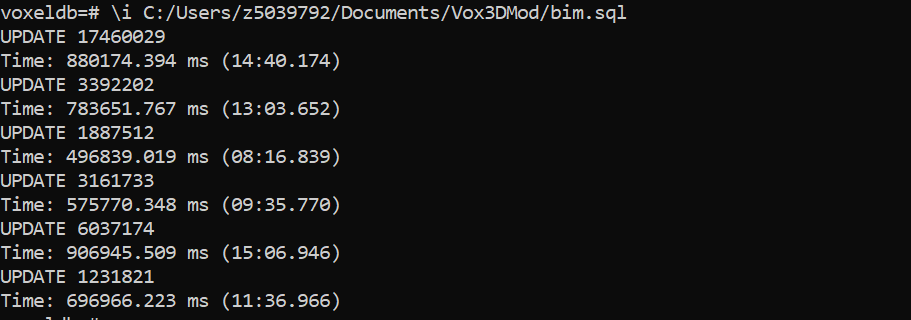


Figure 14. Log info for IFC classID updating

## 3.4 Assign Name for Each Building in Lower Campus

Ignore …

# 4. Conversion to Point

In this section, we consider the second data layout, that stores each voxel as a geometry POINT including (x,y,z).

1. **DELETE** **FROM** voxelpt;
2. **INSERT** **INTO** voxelpt(classID, ifcID, geom) **SELECT** classID, ifcID, ST\_MakePoint(x,y,z) **FROM** voxel **AS** **VALUES**;

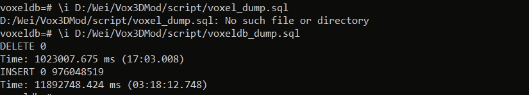


Figure 15. Log info for voxelpt inserting

# 5. Conversion to Multipoint

## 5.1 Create table

DROP TABLE IF EXISTS voxelmpt CASCADE;

CREATE TABLE voxelmpt

(

id serial PRIMARY KEY,

classID INTEGER,

ifcID INTEGER,

geom geometry

);

## 5.2 Partition size

For each IFC object, we regard each IFC object as one Multipoint.

For each GIS building, each building is one Multipoint.

For tree and dtmbot, we combine GIS dataset to decide the patch size.

## 5.3 Data Import

# 6. Conversion to PCPATCH