A VECTOR ANALYTICAL FRAMEWORK FOR POPULATION MODELING

Jessica J. Moehl[1], Eric M. Weber[1],Jacob J. McKee[1]

[1]Oak Ridge National Laboratory, Oak Ridge, TN, USA (moehjj weberem mckeejj)@ornl.gov

KEY WORDS: Population Modeling, PostgreSQL, PostGIS, raster processing, vector processing, gridded population

ABSTRACT:

We propose a vector alternative to the typical raster based population modeling framework. When compared with rasters, vectors are more precise, have the ability to hold more information, and are more conducive to areal constructs such as building and parcel outlines. While rasters have traditionally provided computational efficiency, much of this efficiency is reduced at finer resolutions and computational resources are more plentiful today. Herein we describe the approach and implementation methodology. We also describe the output data stack for the United States and provide examples and applications.

1. INTRODUCTION

High resolution mapping of human populations is often achieved through the disaggregation of aggregate counts (e.g. census tabulations) from tabulation areas (source zones) to smaller areas (target zones), with the aid of covariate spatial data characterizing the natural or built environment (e.g. land cover/use, building footprints) with some known or presumed functional relationship with population density. Source zones and natural/built environment data, found in a variety of raster-/vector formats and resolutions, are often converted to a common raster resolution for analysis (Lloyd et al., 2017, Mennis, 2003, Bhaduri et al., 2007, Freire et al., 2016). This approach is computationally efficient at coarse resolutions and existing software and methods facilitate modeling for those with an understanding of raster-based spatial analysis (Leyk et al., 2019), but has potential shortcomings due to limitations of raster data formats. When compared to a vector data model, rasters are less precise, usually hold less information, and are less conducive to smaller area constructs, such as building outlines and parcels. Given these shortcomings, we propose a vector analytical framework for population modeling. The framework is designed to combine all of the lines defining the input layers so that fields enclosed by those lines (i.e. polygons) are uniformly attributable to each of the input layers. This richer data stack allows for the development of models with more complex logic that are straightforward to implement and explain, as well as potentially increasing the accessibility of modeled estimates and intermediate layers to a broader audience. Furthermore, by embedding grid cell boundaries into the vector framework from the outset, we maintain the ability to generate raster layers (e.g., gridded population estimates) using this framework.

2. MAIN BODY

2.1 Approach

Within our framework, capturing all relevant built environment attributes as well as all source zone identifiers at the finest resolution requires calculating the spatial intersections of all input layers as a first step. Unlike approaches that convert all vector data to raster or that simply join attributes on polygon centroids, our approach maintains all attribute information from the input layers with their original spatial precision. We thereby retain the flexibility to aggregate at subsequent steps according to modeling assumptions. This method also allows for the inclusion of a regular grid for aggregation, preserving our ability to aggregate back to the raster format familiar to users of high resolution population estimates. Simply put, the vector analytical framework is designed to combine all of the lines defining the input layers so that fields enclosed by those lines (i.e. polygons) are uniformly attributable to each of the input layers.

2.2 Implementation

Our population models rely on myriad datasets, some with point/polygon geometries natively in shapefiles, geodatabases, geojsons, etc, some in tabular form, such as from csvs. We have found PostgreSQL with PostGIS to be an ideal central storage point for extract, transform, and load (ETL) processes from the many data sources. While using this as storage and service for our previous raster based modeling efforts, we found the PostGIS processing capabilities to be impressive and thus developed our vector based framework to run within the PostgreSQL/PostGIS environment.

While SQL is more accessible as a descriptive language, some understanding of the underlying PostgreSQL query engine and configuration settings is required for performant outcomes. We arrived at the following order of operations:

1. Index all geometries and id columns to optimize evaluations and joins.

2. Assemble all linear boundaries from polygon inputs, census blocks and parcels, as well as the regular grid bounding lines, into a single ‘blades’ table; this prevents duplication of features when cutting building polygons.

3. Identify and union all blades that intersect each building and join that line to the building geometry by **buildingid** so that the resulting table has a record for each building with a column for the single building polygon and a single blade that intersects it. This table can be iterated over and the building polygon split by the blade.

4. The resulting table of split geometries is then merged into a new table with the building geometries that had no intersections with the blades.

5. A point within the polygon is calculated for joining all input parameter polygons. This point is the centroid if the centroid is within, otherwise it is a point on surface so that spatial joins are guaranteed to be within the building parts.

6. The final step is a query to join all the parameter polygons. Where multiple distinct values are possible, variables are stored in arrays. The result is stored in a table structured as shown:

```
uid bigint,
buildingid bigint,
censusblock character varying(15),
parcelids array,
landuses array,
area_m2 double precision,
point_geom geometry,
polygon_geom geometry,
grid_row integer,
grid_column integer
```

   The final two columns are required only to facilitate the conversion to a raster grid.

We have implemented these concepts in SQL orchestrated through Python 3.x. We have run these scripts on various versions of PostgreSQL/PostGIS from PostgreSQL 11.12 with PostGIS 2.5 to PostgreSQL 13.1 with PostGIS 3.0. All processing times and discussion references herein were performed in a centos 8 environment with 128 cores and 500 GB of RAM using PostgreSQL 13.1 with PostGIS 3.0. USA Structures building outlines (Oak Ridge National Laboratory, 2021) are stored in one table per state. Lightbox parcel data (DMP/Lightbox, 2020) and US Census TIGER data (U.S. Census Bureau, 2019) are stored in one table each, with partial indexes on geometries per state.

2.3 Processing times

The processing times shown in Table 1 are for the polygon to line conversions and the creation of the 3 arcs-econd grid cell bounding lines, described in section 2.1, operation 2. As such, the time is a function of the total rows. Larger extents have more grid lines and more populated states have more blocks and parcels.

Table 2 lists processing times for operations 3 to 6 in section 2.1, as they are scripted together. Variations from state to state are not simply dependent on the number of rows in the input tables; they depend on the complexity of polygons in the building and land use layers and the spatial relationships between those layers. Operation 3, for example, takes longer for a state with more building/blade intersections. However, the relationship between number of building polygons and processing time is roughly linear (Figure 2). With regard to operation 2, the number of parcel polygons is moderately predictive of processing time; the number of census blocks and the total land area to be diced into grid cells also contribute to the processing time variation (Figure 1).

2.4 Discussion

The vector analytical framework has been transformative in our population modeling work for LandScan USA (Moehl et al., 2020, Weber et al., 2019). It allows us to move the heaviest

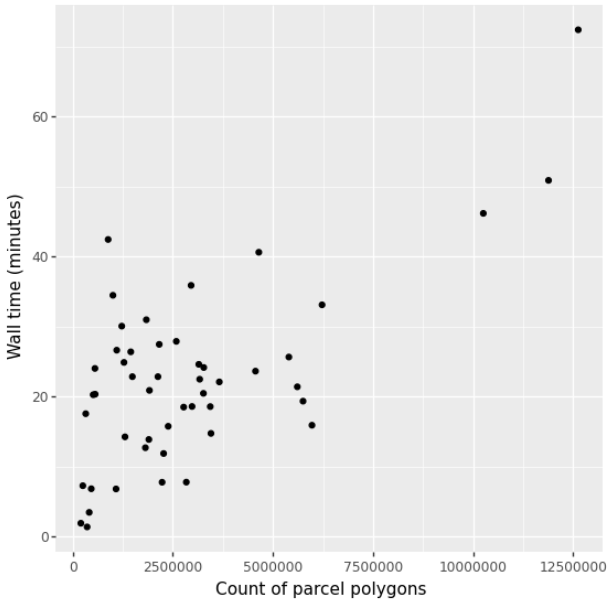

Figure 1. Processing times for blade generation by number of parcel polygons.

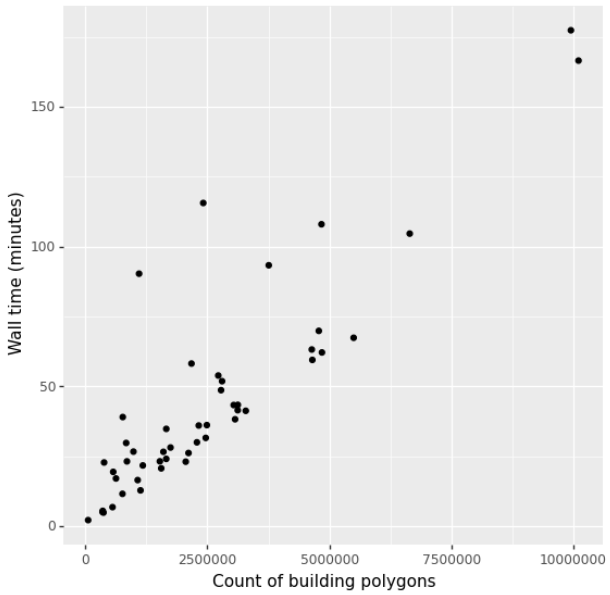

Figure 2. Processing times for generation of building parts by number of building polygons.

Table 1. Processing time for generating the blades from parcels, census blocks, and grid cell boundaries (step 2 in section 2.1). States are run in parallel; total processing time limited by longest-running state (California in this case).

State	Wall Time	State	Wall Time
AL	0:18:35.760	NC	0:25:39.478
AR	0:13:52.950	ND	0:24:01.913
AZ	0:24:09.857	NE	0:26:38.188
CA	1:12:24.087	NH	0:06:50.751
CO	0:27:53.953	NJ	0:07:47.448
CT	0:06:49.054	NM	0:26:24.210
DC	0:01:55.219	NV	0:30:04.362
DE	0:03:28.271	NY	0:33:07.005
FL	0:46:10.952	OH	0:15:55.075
GA	0:23:38.582	OK	0:27:28.125
IA	0:15:46.051	OR	0:30:59.154
ID	0:34:28.764	PA	0:19:21.387
IL	0:21:24.412	RI	0:01:23.531
IN	0:14:45.101	SC	0:18:30.442
KS	0:22:51.305	SD	0:20:20.866
KY	0:20:53.742	TN	0:20:28.197
LA	0:22:51.269	TX	0:50:53.683
MA	0:07:46.557	UT	0:24:53.473
MD	0:11:52.809	VA	0:22:06.382
ME	0:20:16.344	VT	0:07:16.702
MI	0:40:37.441	WA	0:22:29.309
MN	0:35:53.678	WI	0:18:34.736
MO	0:24:36.971	WV	0:14:15.496
MS	0:12:42.376	WY	0:17:33.924
MT	0:42:27.089		

computation to the early stages of production, before many decisions, removing barriers to iterating and adjusting implemented decisions. Conversely, raster based methods have the heaviest computations at the penultimate step. The US analytical data stack herein consists of 270 million plus rows of building parts resulting from running 123 million building outline polygons through our framework, using over 152 million parcel polygons, over 11 million census blocks, and over 65 million unique grid cells also embedded; all stored and calculated within PostGIS. These 270 million building parts having no loss of information and linking back to the source datasets, become the basis for all subsequent decisions and analysis in the workflow, including handling of overlapping parcel polygons, interpreting other confounding land use information, imputation of null land uses, as well as summaries of area by land use by source zone for further statistical analyses.

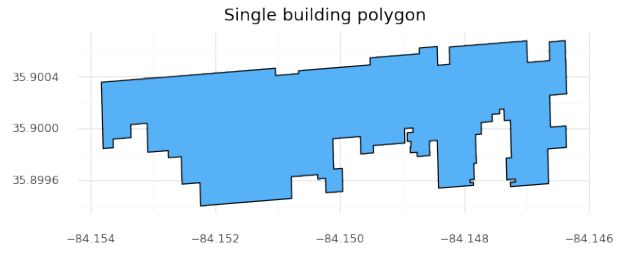

Figure 3. A single building polygon.

Figures 3 - 5 show how our vector analytical framework starts with a single building polygon (3), representing a row of several shops in Knoxville, TN, and results in 29 records (5) after being split by parcels and grid lines. An example row, with the polygon_geom highlighted in Figure 5, is shown below:

Table 2. Processing time for generating building parts from buildings, parcels, and census blocks (3 to 6 in Section 2.1). States are run in parallel; total processing time limited by longest-running states (California and Texas at close to three hours).

State	Wall Time	State	Wall Time
AL	0:36:06.501	NC	0:59:26.013
AR	0:20:38.310	ND	0:19:22.768
AZ	0:53:50.692	NE	0:21:40.890
CA	2:57:20.312	NH	0:06:45.709
CO	0:58:07.492	NJ	0:31:32.009
CT	0:12:45.892	NM	0:26:37.462
DC	0:02:07.410	NV	0:29:43.461
DE	0:04:48.488	NY	1:02:05.974
FL	1:44:36.450	OH	1:07:20.716
GA	1:33:17.016	OK	0:35:57.850
IA	0:26:08.041	OR	0:34:45.796
ID	0:23:08.115	PA	1:47:57.058
IL	1:03:09.031	RI	0:05:03.301
IN	0:41:13.737	SC	0:29:57.128
KS	0:26:33.453	SD	0:17:01.052
KY	1:55:33.864	TN	0:41:25.756
LA	0:28:05.476	TX	2:46:29.628
MA	0:23:01.636	UT	1:30:16.304
MD	0:24:01.672	VA	0:43:21.270
ME	0:11:31.417	VT	0:05:25.795
MI	1:09:49.176	WA	0:48:36.747
MN	0:51:49.121	WI	0:43:17.230
MO	0:38:11.729	WV	0:16:26.612
MS	0:23:09.423	WY	0:22:44.552
MT	0:38:59.357		

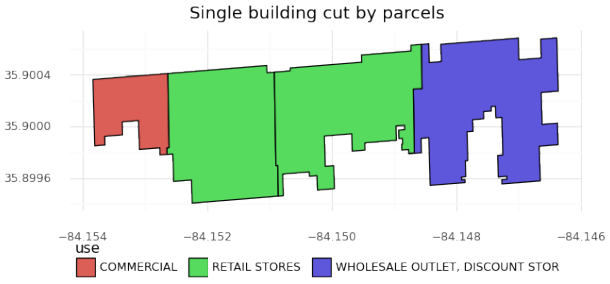

Figure 4. Four parcels intersect this single building polygon. Three land uses are present.

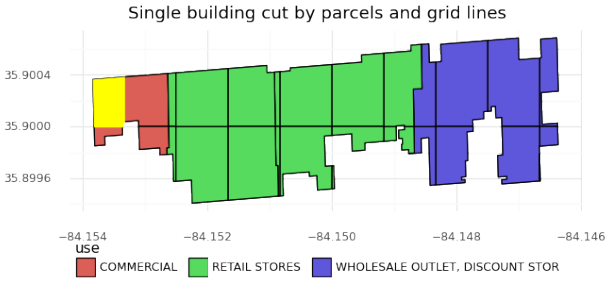

Figure 5. The single building polygon, when split by parcels and a regular grid, becomes 29 records. A single record is highlighted in yellow.

```
uid: 12016302
buildingid: 1280868
censusblock: '470930058031033'
parcelids: {534343}
land_uses: {'COMMERCIAL'}
area_m2: 1816.42
point_geom: SRID=4326;POINT(...)
polygon_geom: SRID=4326;POLYGON(...)
grid_row: 16320
grid_column: 49016
```

Spatially joining attributes from one polygon layer to another is a common procedure. This is often done using the centroid of the target layer to ensure one record in the resultant table for each of the target layers. An example using building outlines as the target layer for parcel land uses, would result in a 1 to 1 ratio of records between the input buildings layer and the output buildings + land use layer. Figures 3 and 4, which show one building polygon intersecting four parcels, illustrate how a spatial join of parcels to a building using centroids might be insufficient. With the building parts, we can calculate the ratios of building/parcel intersections to buildings by any any census geography. A ratio of 1:1 would be equivalent to the centroid based join, with no building intersecting parcel boundaries. Figure 6 shows that a centroid based spatial join approach would not be sufficient in many counties. The highest ratios are found in the New York City counties. Queens County is one of these extreme examples with a ratio of 3.075:1. Figure 7 shows the 38 buildings before being split by parcels, as shown in Figure 8. These figures illustrate that while the county level ratio at 3:1 is extreme relative to the rest of the US, there are even more extreme ratios at the tract scale; 12.94:1 in this example. The vector analytical framework and the building parts allow us to understand these ratios across our entire study area at any scale, not just for a sample. Otherwise, it would be very difficult to know and explain where and how a centroid method would be insufficient.

Figure 8 also illustrates where the vector framework allows for the precision of the underlying land use data to be maintained throughout the modeling process. In a raster based workflow, rasterization would inevitably distort the relative distributions of land use areas at all scales and depending on the cell size and alignment some land uses might be lost entirely.

2.5 Applications

2.5.1 Overlap Handling Vector polygon datasets often have overlapping features. This is sometimes the result of misalignment, but is also often a true representation, a building outline and a land use polygon within OpenStreetMap, for example. Measuring the impact of these overlaps on population models can be difficult. In a raster framework, calculating the impact of overlaps requires one to rasterize for each possible handling scenario. This impact can also change depending on spatial resolution, i.e. cell size, which would require further rasterization for evaluation. Additional complexity is added for evaluation at different scales, such as state, county, or tract. With the vector analytical framework, all decisions are downstream of the heaviest computations, thereby encouraging and facilitating iterative analysis and refinement of methods. With the building parts, we have no information loss and all area and land use information is in the context of the building outline polygons. This allows us to calculate overlap in terms of area of the building parts at various scales rapidly. As specified in

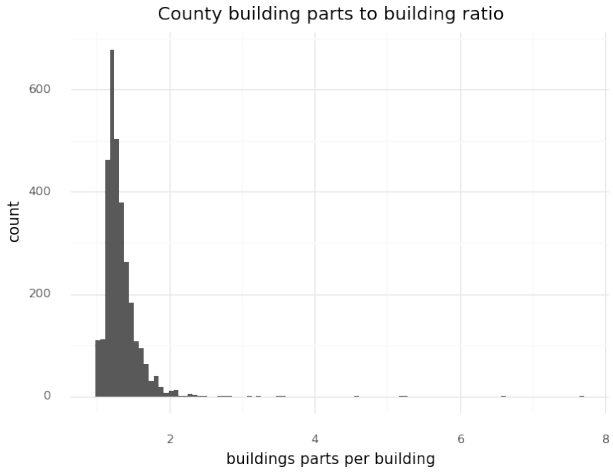

Figure 6. Most counties have ratios between 1 and 2, though some have much higher ratios.

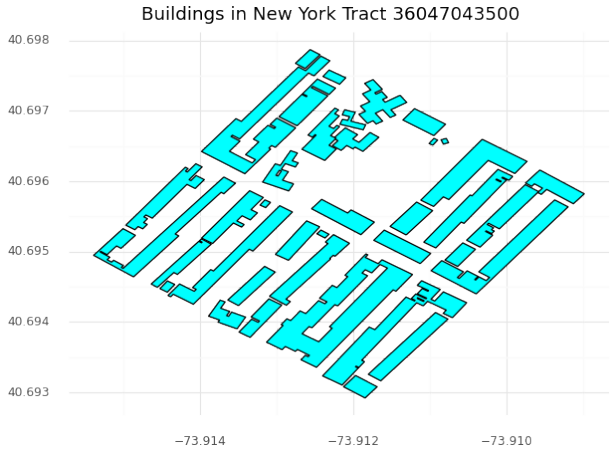

Figure 7. Tract 36047043500 has 38 buildings

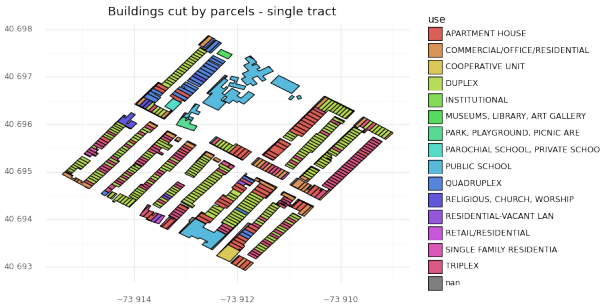

Figure 8. Tract 36047043500 has 492 building/parcel records.

the 2.2, when multiple values occur in the parameter polygons, arrays are used to capture the distinct values present. We can use this in the sql logic to find the building area with multiple parcels by any level of census geography.

```
SELECT sum(area_m2),
   substring(censusblock, 1, 5) as county
FROM table
WHERE array_length(a.parcelids, 1) > 1
GROUP BY county
```

It takes about a minute to process the 270 million building parts and find that the building area overlap across the states ranges from 0.75% in Indiana to 16.15% in Washington, D.C. as shown in Figure 9. It takes another minute each to process the 70,000 tracts and 3,000 counties in Figures 10 and 11, respectively. These calculations show that the chosen method(s) for overlap handling, such as 'prefer residential' or 'take the smallest parcel' can greatly impact population models at finer scales as there are some geographies with large amounts of overlapping information. These calculations can also aid in identification of any systematic issues that might be present in the input datasets.

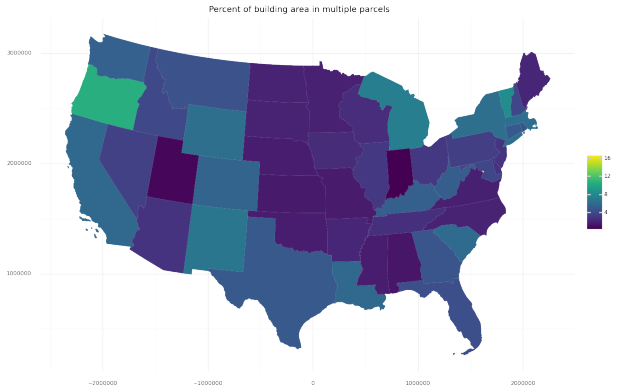

Figure 9. The building area overlap across the States ranges from 0.75% in Indiana to 16.15% in Washington, D.C.

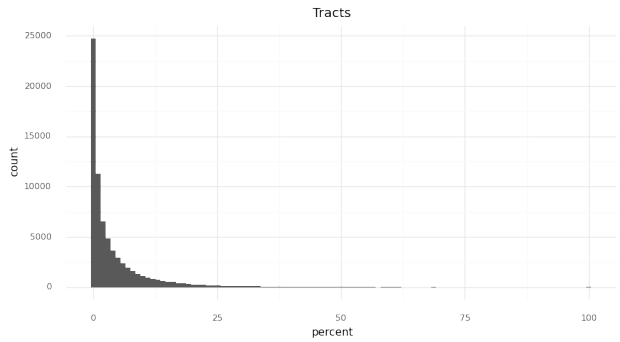

Figure 10. Count of tracts by percentage of building area with overlapping parcels

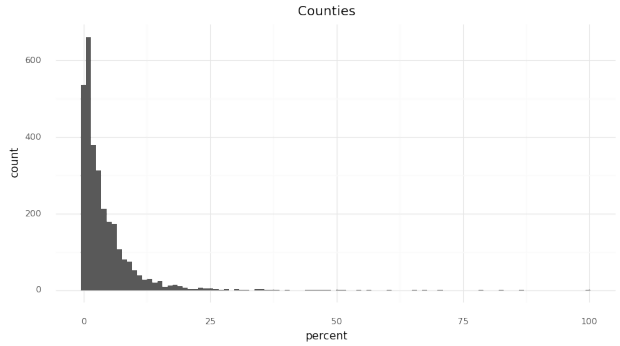

Figure 11. Count of counties by percentage of building area with overlapping parcels

2.5.2 Subpopulation Modeling For this demonstration, let's assume we have a polygon layer representing the extents of the grounds of colleges in the contiguous US. This source zone layer has a unique identifier for each campus and a population to be distributed within the zone. To distribute the population with the zone, we first select the building parts that intersect each zone, along with zone id and population so that the resultant weights table has a row with the uid, area_m2, and land_use attributes from the building parts and the zone id and population from the zone layer. After selection, population is distributed to each of the building parts in the weights table. Any building parts that intersect multiple source zones receive reduced portions of the contributing source zones' populations so that they don't receive extra population. This is calculated by counting the occurrences of each building part, with one occurrence indicating that building part is within one college zone, and then dividing that building part's area by the occurrences count. The area per school is calculated and each building part is given it's portion of the college population according to its portion of of the total school area. The building parts with nonresidential land uses have their areas set to a tiny value (0.0000001), so that a building part with a nonresidential land use receives no population unless all the building parts for that zone have nonresidential land uses. Figure 12 shows the building parts within two campus zone polygons (blue). The darker green building parts within the smaller college zone denotes the presence of two building parts polygons, one for each college. That building would receive all of small source zone population and a fraction of the population from the larger college at half the density of the buildings which only intersect the larger source zone.

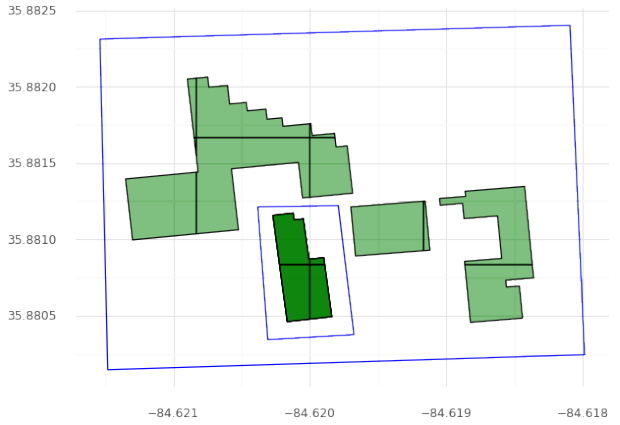

Figure 12. Two college zones (blue outline) share building parts.

3. CONCLUSIONS

We believe this framework, developed in the context of population modeling, is extensible to many other spatial analysis problems. Also, it is an excellent example use of Postgres/PostGIS beyond storing and serving data and is thus of interest and relevance to the FOSS4G community. To GIS practitioners accustomed to a geoprocessing workflow in desktop GIS software (input layers process output layer), our data flow diagrams are familiar, but the scale and performance achieved is not. Many practitioners of traditional GIS spatial analysis can benefit from making their data more analytically accessible to a wider array of emerging data science techniques.

COPYRIGHT

REFERENCES

Bhaduri, B., Bright, E., Coleman, P., Urban, M. L., 2007. LandScan USA: a high-resolution geospatial and temporal modeling approach for population distribution and dynamics. *GeoJournal*, 69(1-2), 103–117. http://link.springer.com/article/10.1007/s10708-007-9105-9.

DMP/Lightbox, 2020. SmartParcels. Licensed through HIFLD Secure Data.

Freire, S., Doxsey-Whitfield, E., MacManus, K., Mills, J., Pesaresi, M., 2016. Development of new open and free multi-temporal global population grids at 250 m resolution. *AGILE 2016*, Helsinki, Finland.

Leyk, S., Gaughan, A. E., Adamo, S. B., Sherbinin, A. d., Balk, D., Freire, S., Rose, A., Stevens, F. R., Blankespoor, B., Frye, C., Comenetz, J., Sorichetta, A., MacManus, K., Pistolesi, L., Levy, M., Tatem, A. J., Pesaresi, M., 2019. The spatial allocation of population: a review of large-scale gridded population data products and their fitness for use. *Earth System Science Data*, 11(3), 1385–1409. https://www.earth-syst-sci-data.net/11/1385/2019/.

Lloyd, C. T., Sorichetta, A., Tatem, A. J., 2017. High resolution global gridded data for use in population studies. *Scientific Data*, 4(1), 1–17. https://www.nature.com/articles/sdata20171.

Mennis, J., 2003. Generating Surface Models of Population Using Dasymetric Mapping. *The Professional Geographer*, 55(1), 31–42. http://www.tandfonline.com/doi/abs/10.1111/0033-0124.10042.

Moehl, J. J., Weber, E. M., Sims, K. M., Trombley, N. E., Weston, S. T., Rose, A. N., 2020. LandScan USA 2019. https://hifld-geoplatform.opendata.arcgis.com/datasets/landscan-usa.

Oak Ridge National Laboratory, 2021. USA Structures. https://disasters.geoplatform.gov/publicdata/Partners/ORNL/USA_Structures/.

U.S. Census Bureau, 2019. TIGER/Line Shapefiles. https://www.census.gov/geographies/mapping-files/time-series/geo/tiger-line-file.2019.html.

Weber, E., Moehl, J., Rose, A., 2019. Areal interpolation of population in the USA using a combination of national parcel data and a national building outline layer. *GeoComputation 2019*, Queenstown, NZ.