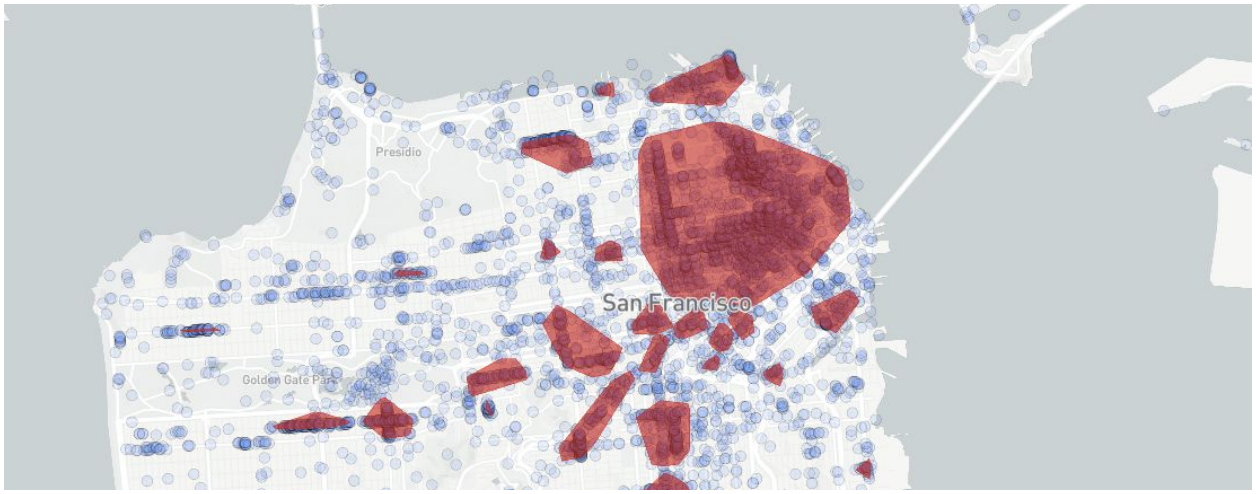


Generating “Interesting Area” Polygons From OSM Data

By Alicia Noel
ajnoel9@gmail.com

Goal

To generate “interesting area” polygons, or Areas of Interest (AOIs) from a database of OpenStreetMap data. Interesting areas consist of a large number of Points of Interest (POIs), like restaurants, cafés, shops, and other things to do.

Introduction

Interesting areas are essentially dense clusters of POIs. We want to consider areas where large numbers of POIs are concentrated, such that area is constrained to a small scale. The ideal scale in this case is on the order of a neighborhood-- within a range that a person could reasonably expect to reach in a few minutes of walking and/or driving.

For this analysis, we will use a PostGIS database and take advantage of the built-in point clustering function ST_ClusterDBSCAN. We can also use ST_ConvexHull to create AOI polygons of each point cluster group.

Requirements

Tools and data

- PostGIS version >= 2.3
- OpenStreetMap data in a PostGIS database

Process

1) Filter OSM data to only include points that we are interested in for POIs (note that this list is broad for this exercise and would need to be filtered more carefully)
    - Amenity
        - Excluding: 'toilets', 'parking', 'bench', 'bicycle parking', 'waste basket'

    - Historic
    - Leisure
    - Shop
    - Sport
    - Tourism
2) Use PostGIS function ST_ClusterDBSCAN to cluster points within a specified distance of each other and give each cluster point a cluster_id.
    - ST_ClusterDBSCAN: to be included in a cluster, a geometry must be within “eps” distance of at least minPoints number of other geometries.
3) Use PostGIS function ST_ConvexHull to create a bounding AOI around each point cluster grouping.

Scaling
This process can be scaled countrywide or even worldwide by constraining it to regional areas and then merging the resulting AOIs. One option would be to just run on all metro areas. Another would be to create a covering grid (with some overlap) and run on each grid. By breaking data down into regions, we can process multiple regions in parallel.

Challenges and Concerns

We can’t ground truth every interesting area in the world. How can we be confident that the areas of interest are accurate?
First, test out the algorithm on an areas that we are familiar with so that we can gauge appropriate threshold values for the cluster function. On a larger scale, I would expect to see larger and/or more frequent AOIs in highly populated areas. Since AOIs are based on the concentration of points of interest, populated areas like dense city centers are more likely to have a greater concentration of POIs. I would overlay the AOIs on a population layer and see how they correlate.

Neighborhoods all over the world are constantly changing. How do these areas of interest keep up with reality?
A static map of AOIs would quickly become outdated. However this process for generating AOIs can be automated and would be continuously updating and refreshing. With this workflow fully automated and scalable, it could be run on a weekly or monthly basis. Or it could be set to run on regions where new OSM updates are detected.

What are the ideal cluster thresholds?
It is challenging to get a cluster size that works for both major cities and smaller less populated areas. Clusters can quickly grow too large in size in major cities. But using lower threshold values results in losing smaller clusters in surrounding suburban areas.
Therefore, clustering threshold values (min_points, distance) for ST_ClusterDBSCAN vary depending on region. For example, we need higher thresholds in San Francisco because there are so many POIs, but a less populated area like Santa Rosa would need lower thresholds. The

clustering values should scale based on total number of POIs in the regional area, so that the most populated areas don't overshadow lower population areas.

Example

This working example shows the process of creating AOIs given an example OSM dataset of the San Francisco Bay Area.

Example OSM data downloaded from Mapzen at:
https://s3.amazonaws.com/metro-extracts.mapzen.com/san-francisco-bay_california.osm.pbf

1. Set up database

```
CREATE DATABASE osm;
\connect osm;
CREATE EXTENSION postgis;
```

2. Load example OSM dataset into postgres database:

```
osm2pgsql -c -d osm -U alicia -W -H localhost
/Users/alicia/Downloads/san-francisco-bay_california.osm.pbf
```

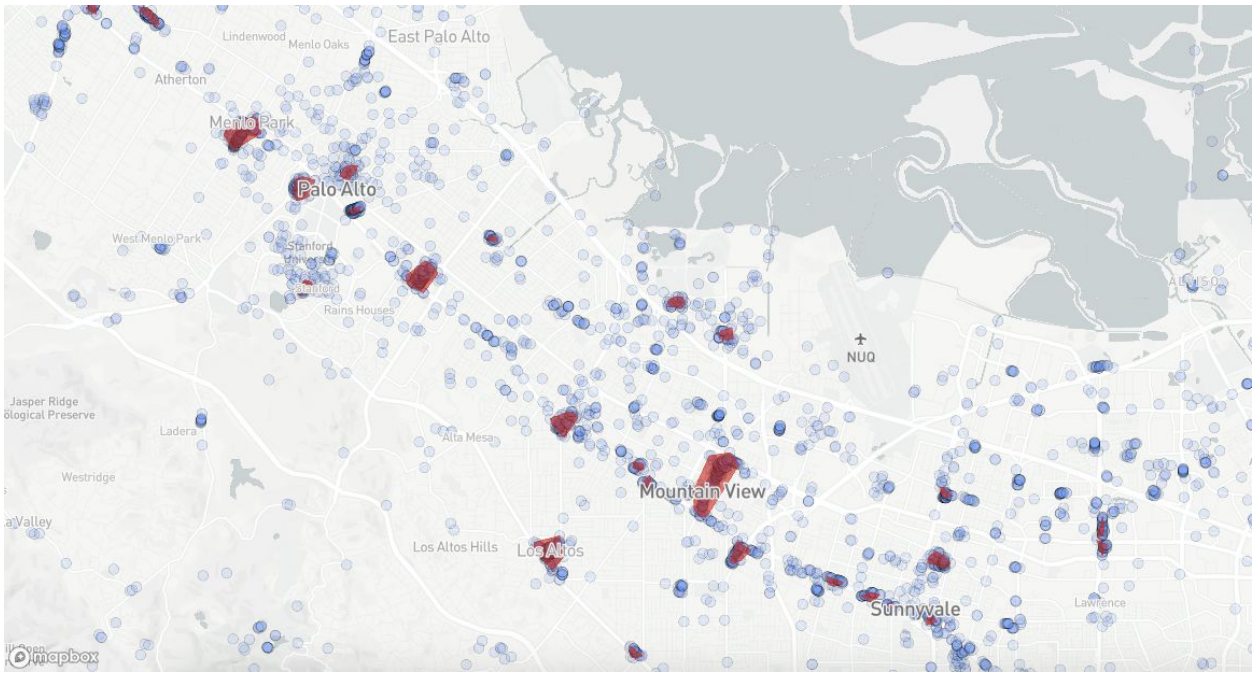
3. Filter OSM data to only include points that we are interested in for POIs

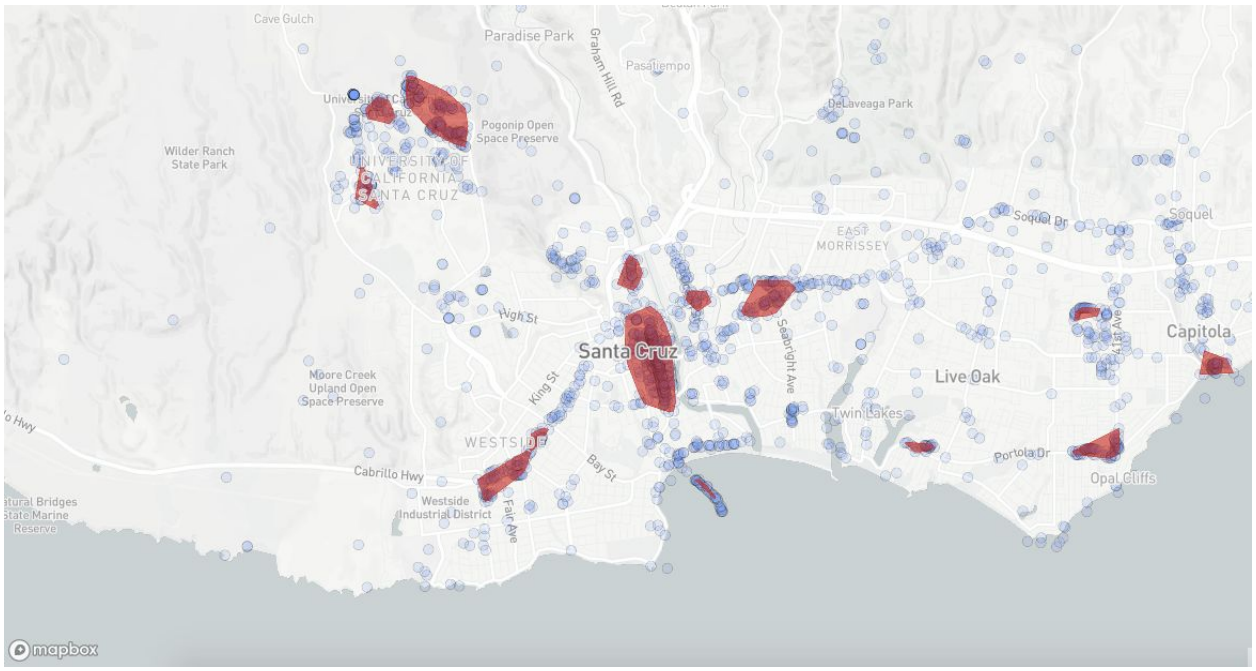
```
CREATE TABLE pois AS (
SELECT amenity, historic, leisure, shop, sport, tourism, way
FROM planet_osm_point
WHERE amenity NOT IN ( 'toilets', 'parking', 'bench', 'bicycle_parking',
'waste basket' )
  OR historic IS NOT NULL
  OR leisure IS NOT NULL
  OR shop IS NOT NULL
  OR sport IS NOT NULL
  OR tourism IS NOT NULL
);
ALTER TABLE pois ADD COLUMN id serial PRIMARY KEY;
```

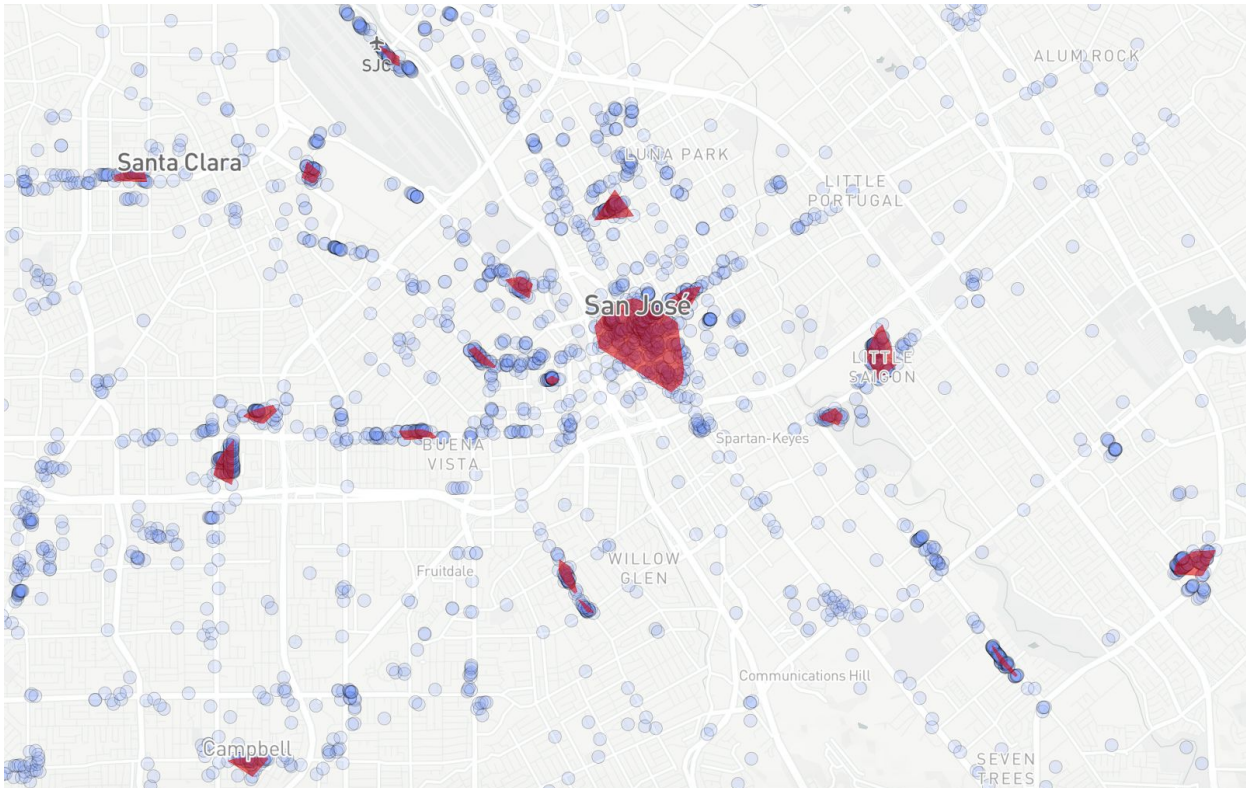
4. Use function ST_ClusterDBSCAN to create point clusters, and function ST_ConvexHull to create an AOI polygon encompassing points within a cluster.

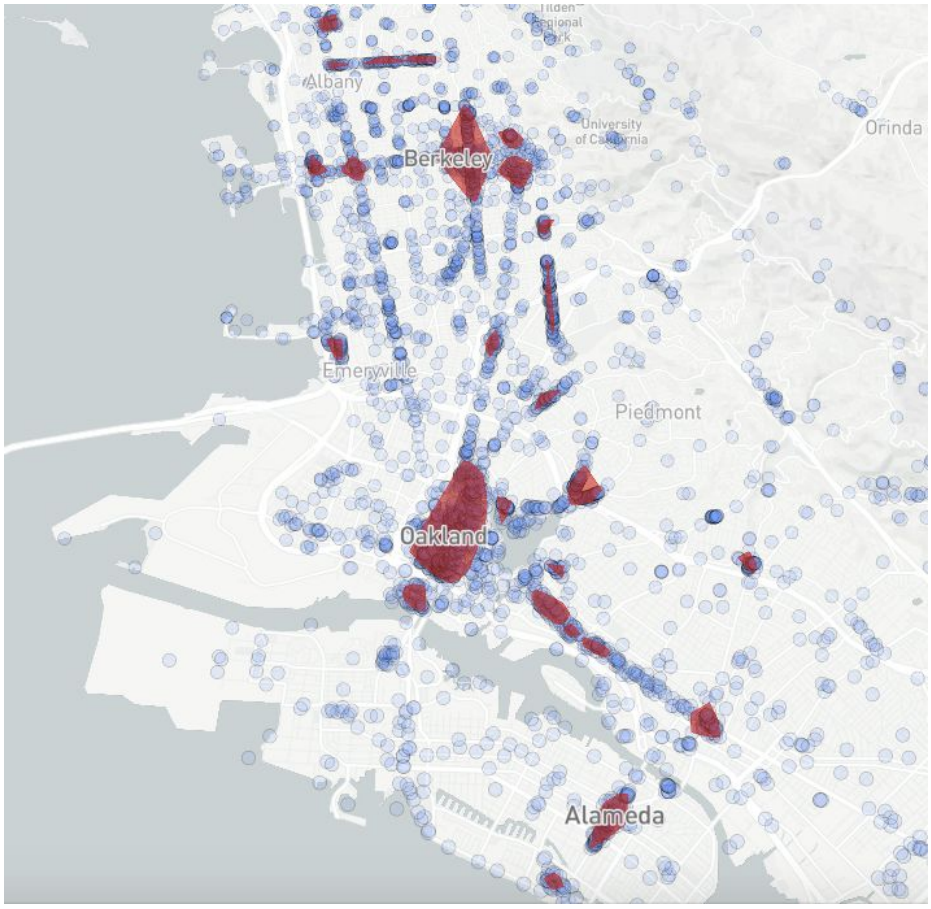
```
CREATE TABLE aois AS (
SELECT ST_ConvexHull(ST_Collect(c.way)), cluster_id
FROM(
  SELECT id, ST_ClusterDBSCAN(way, eps := 200, minPoints := 20)
  OVER () AS cluster_id, way
  FROM pois) c
WHERE cluster_id IS NOT NULL
GROUP BY cluster_id
);
```

5. Publish with Mapbox Studio!
   Explore the map here:
   https://api.mapbox.com/styles/v1/ajnoel/cj7fnqq3g2z982stgw17gm331.html?fresh=true&title=true&access_token=pk.eyJ1IjoiYWpub2VsIiwiYSI6IjQ3NTY4MGNhNGM4M2JmMGE4OTNiYWU2ZjFhN2U5ZDZkIn0.9pxPj81TQmIKBMVkPd_kxg#11.92/37.4041/-122.0842






SJC
Santa Clara
ALUM ROCK
LUNA PARK
LITTLE PORTUGAL
San José
BUENA VISTA
Spartan-Keyes
Fruitdale
WILLOW GLEN
Communications Hill
Campbell
SEVEN TREES


Albany
University of California
Berkeley
Orinda
Emeryville
Piedmont
Oakland
Alameda