



Advanced OpenStreetMap Cartography II

“Dealing with Detail”

Andy Allan

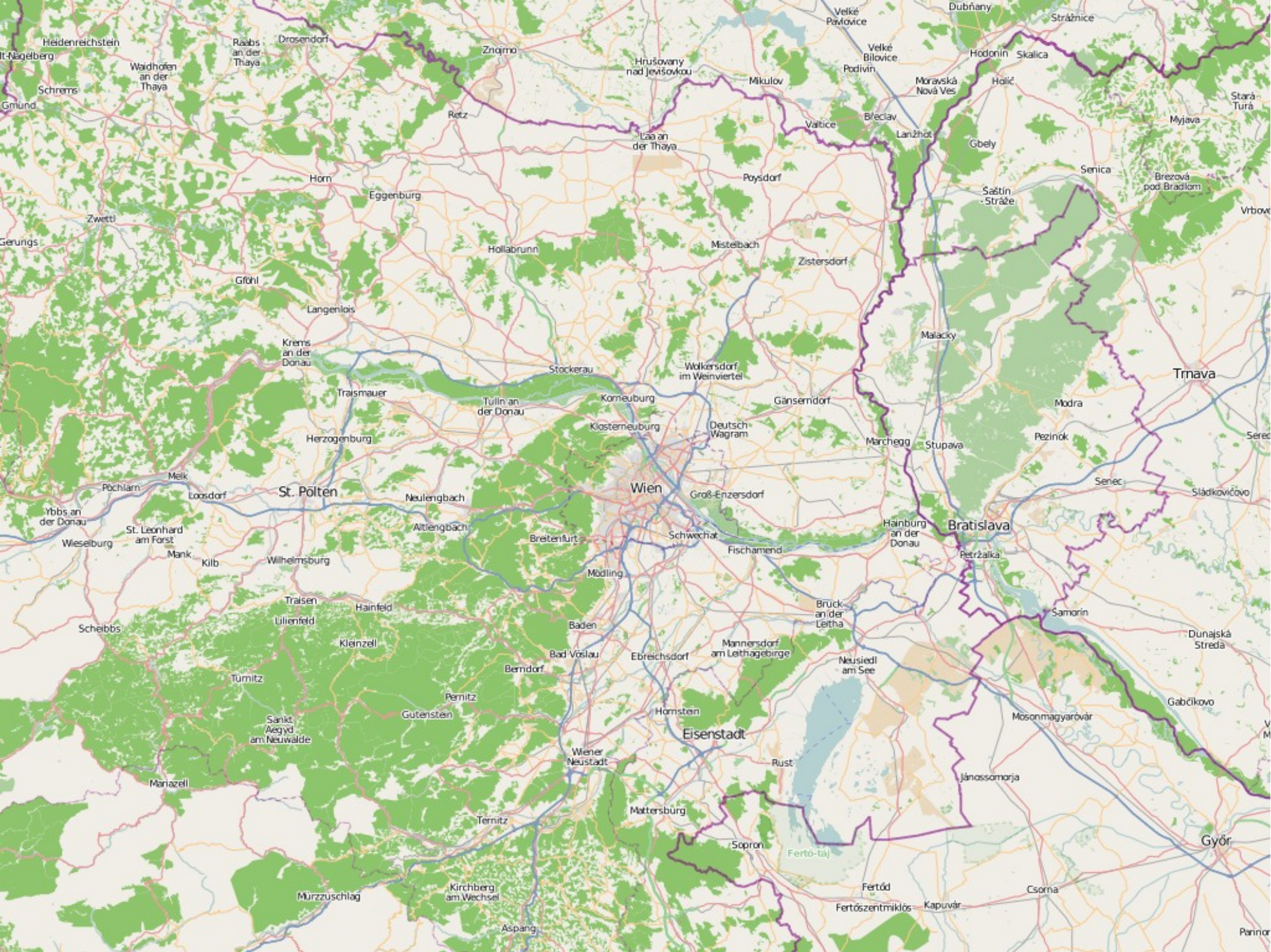
www.gravitystorm.co.uk

www.opencyclemap.org

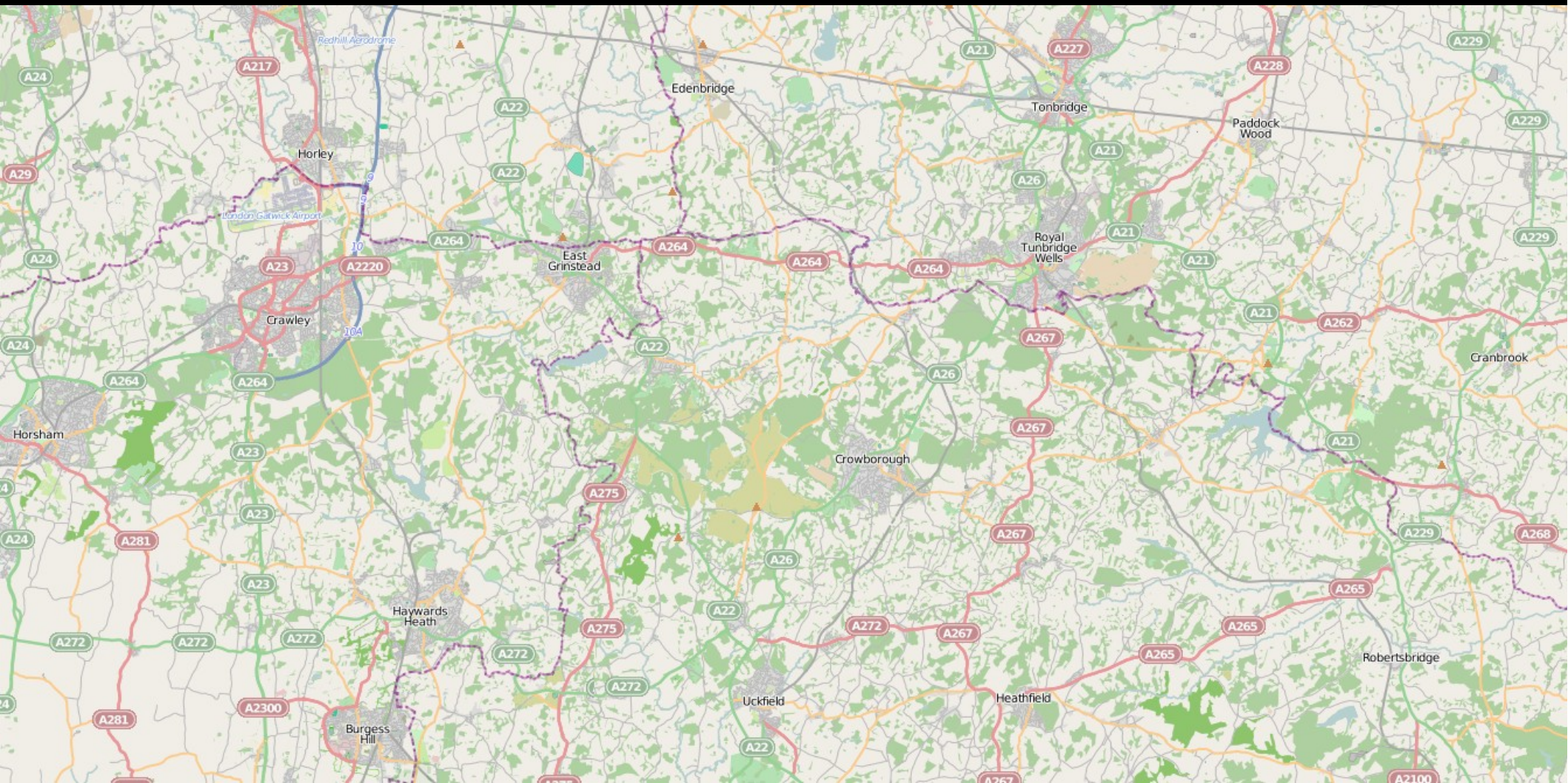


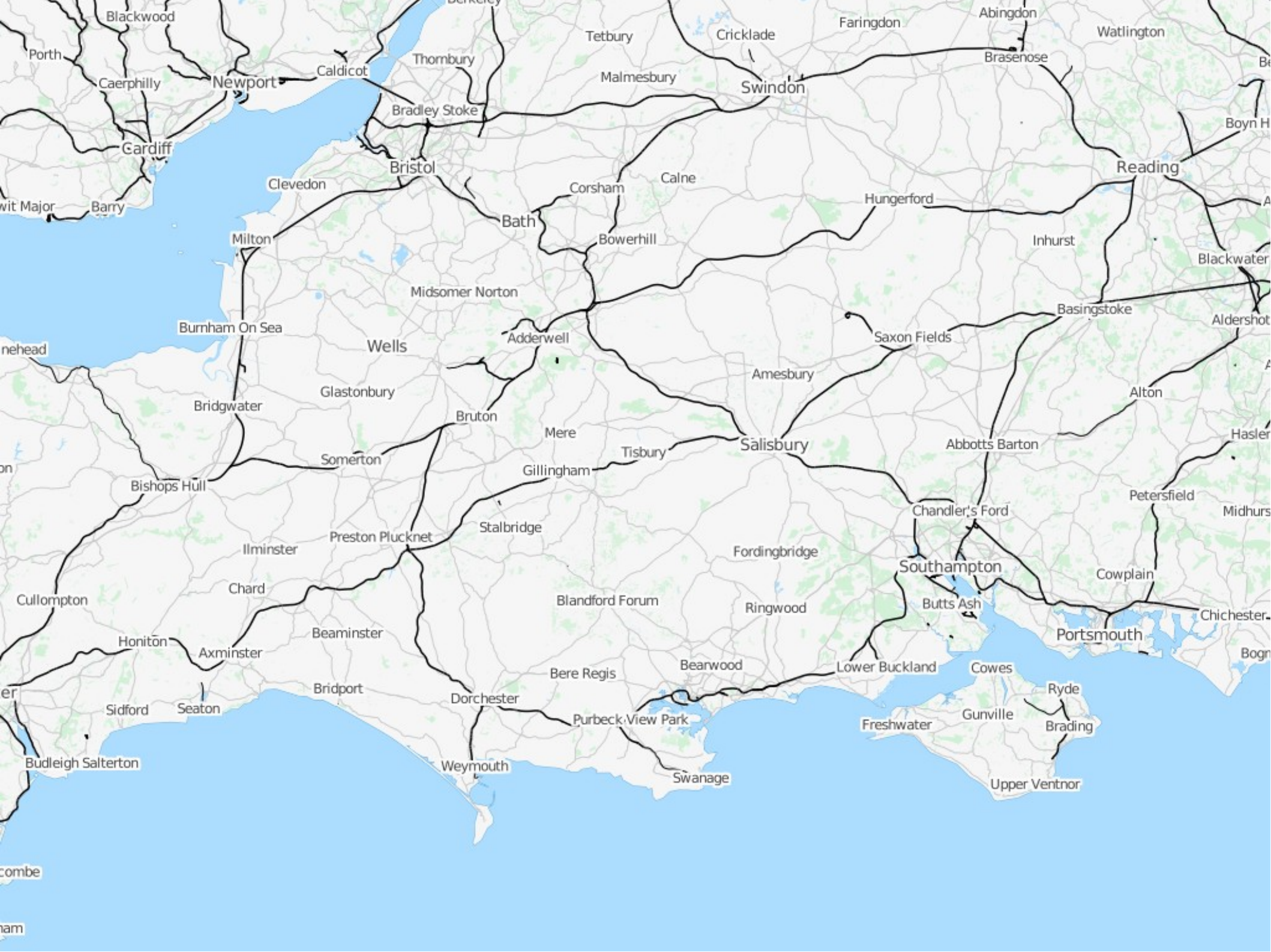
The Basics

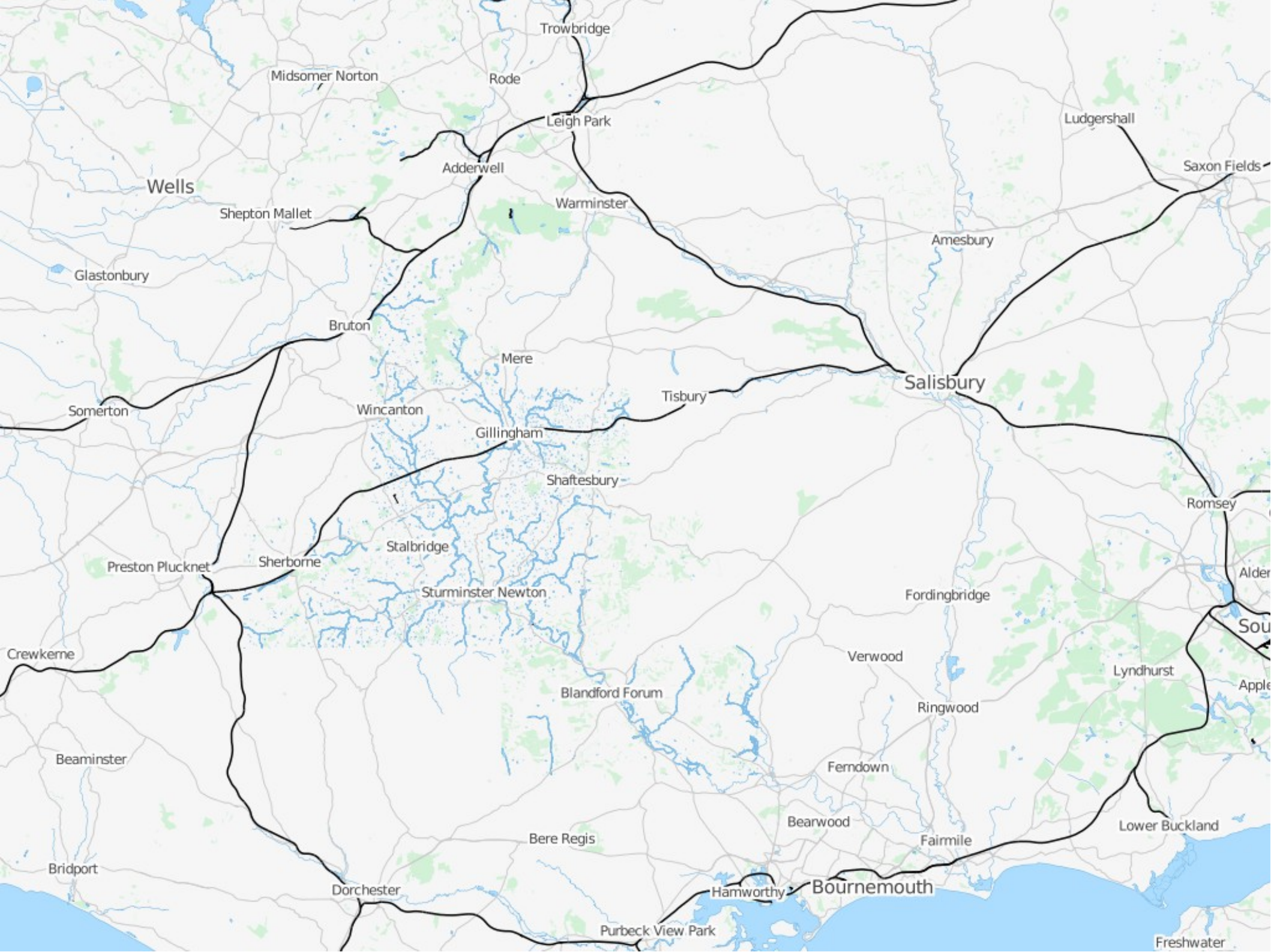




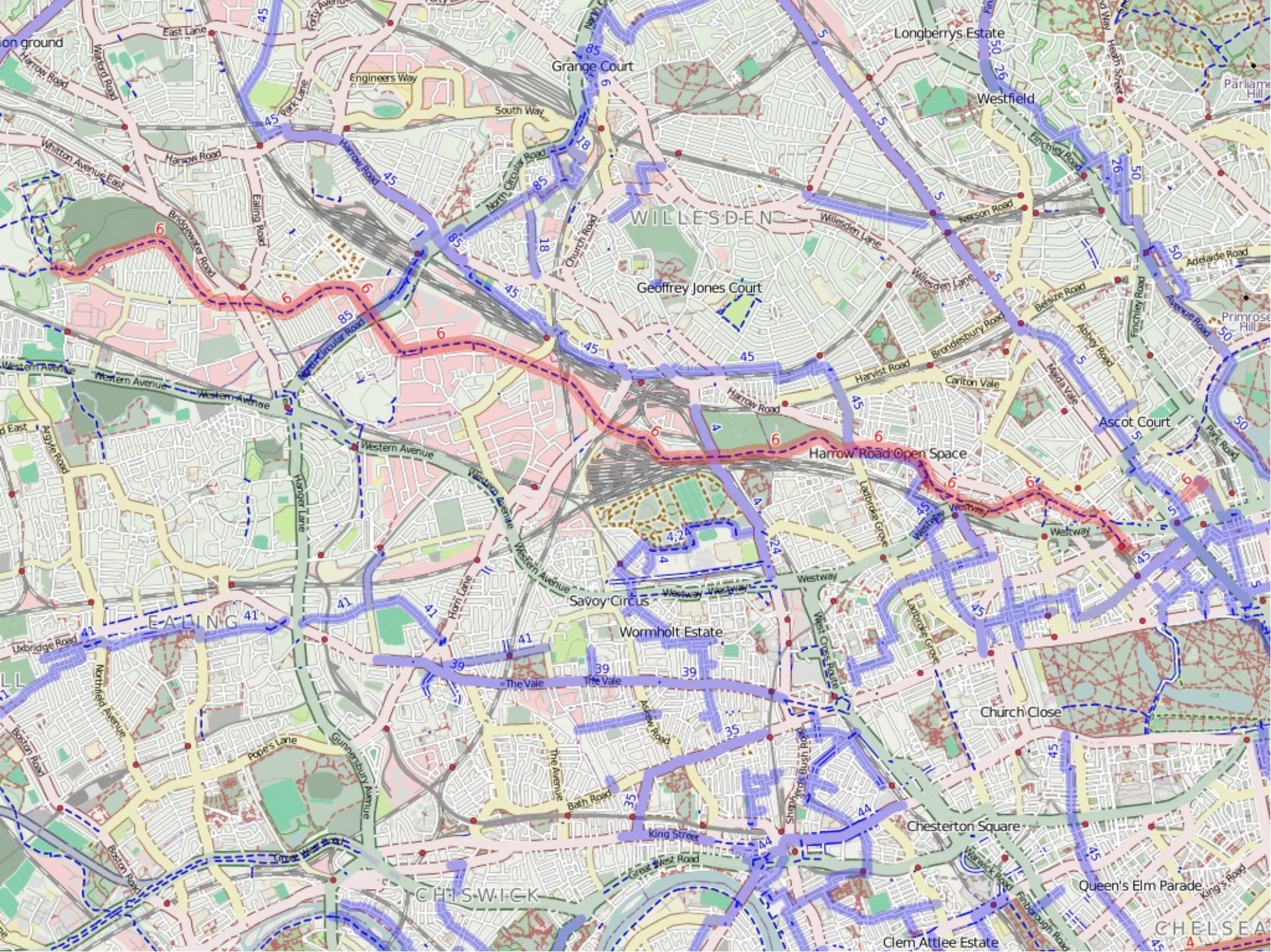
Small Polygons

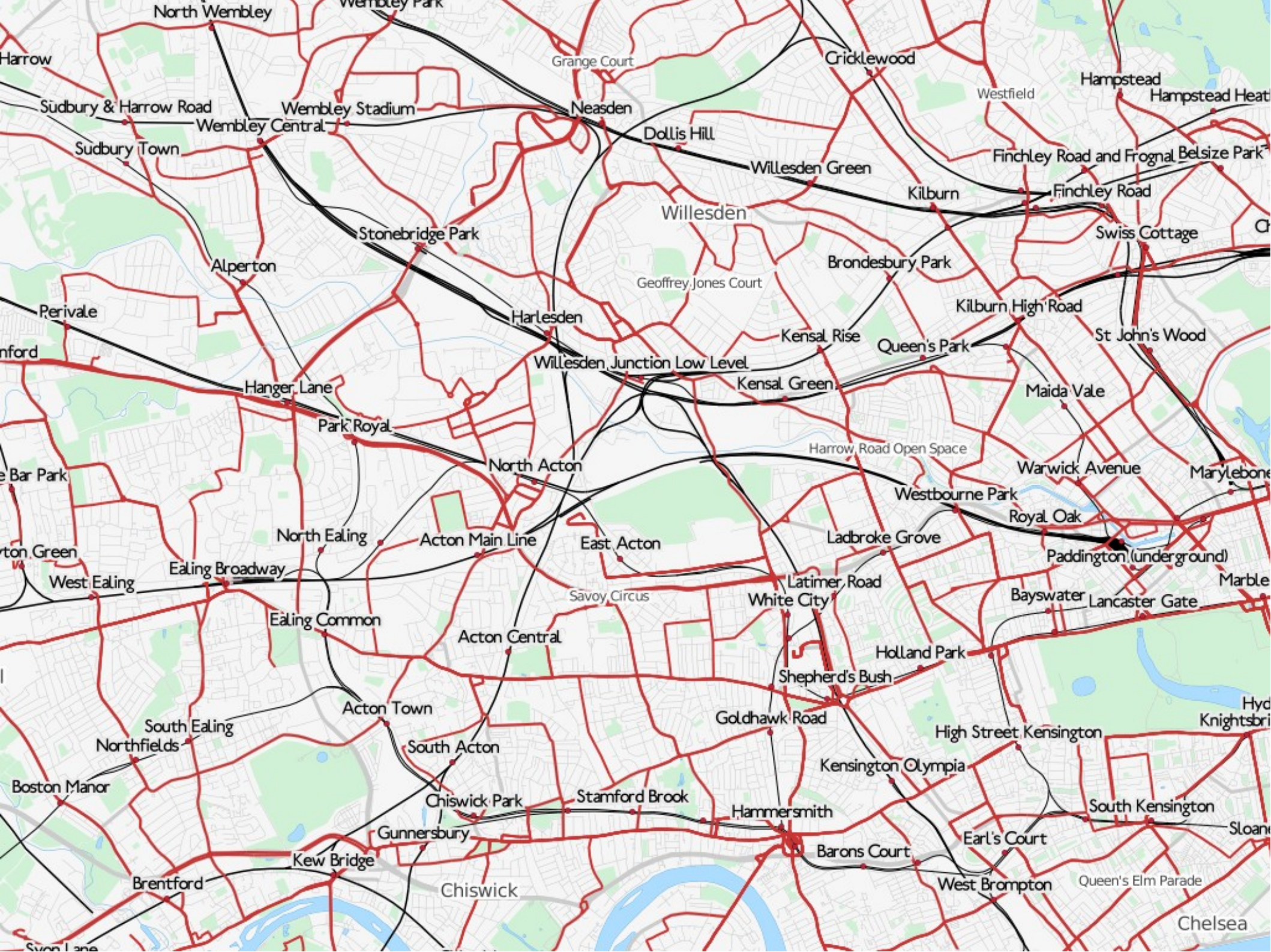


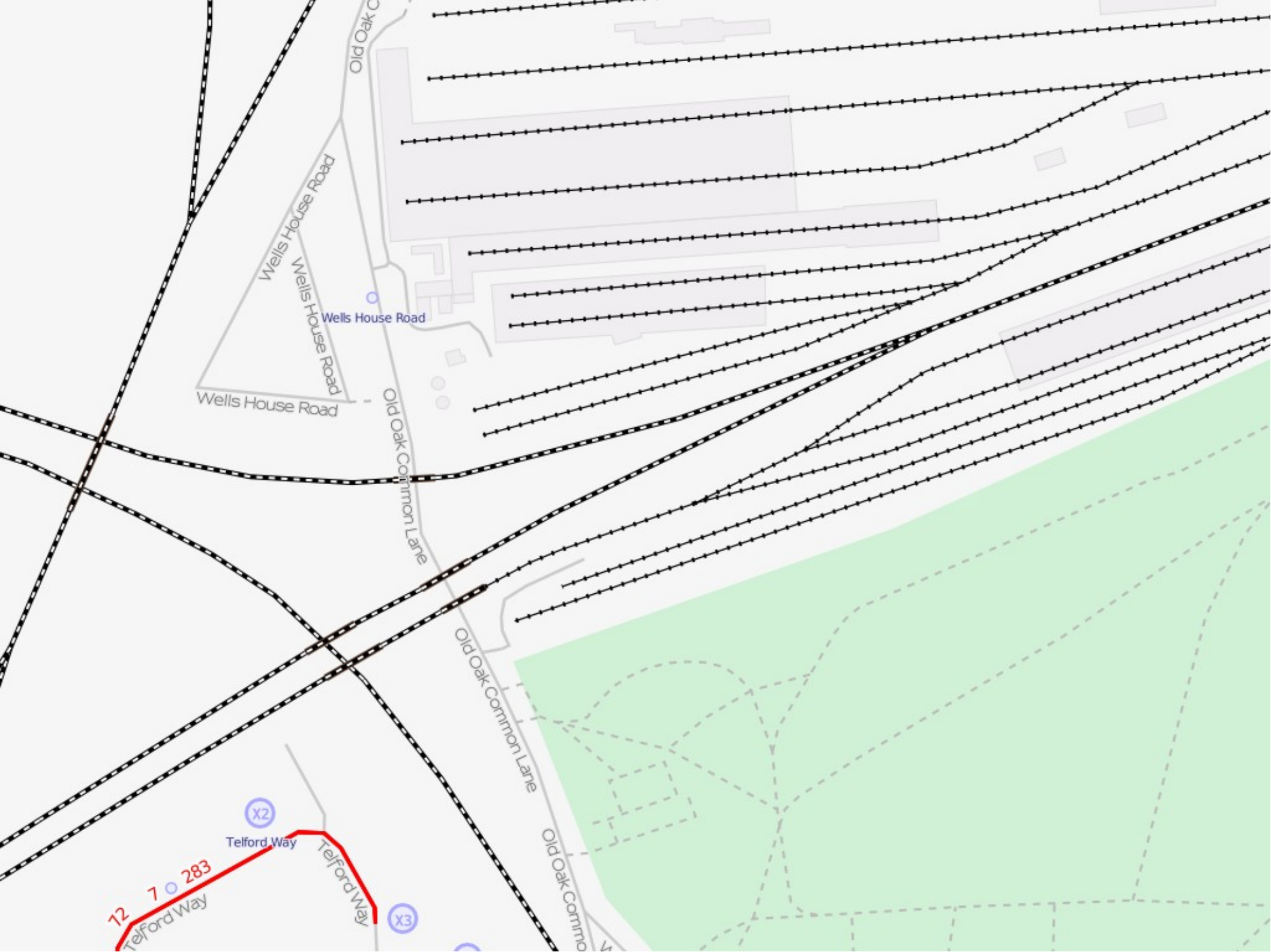




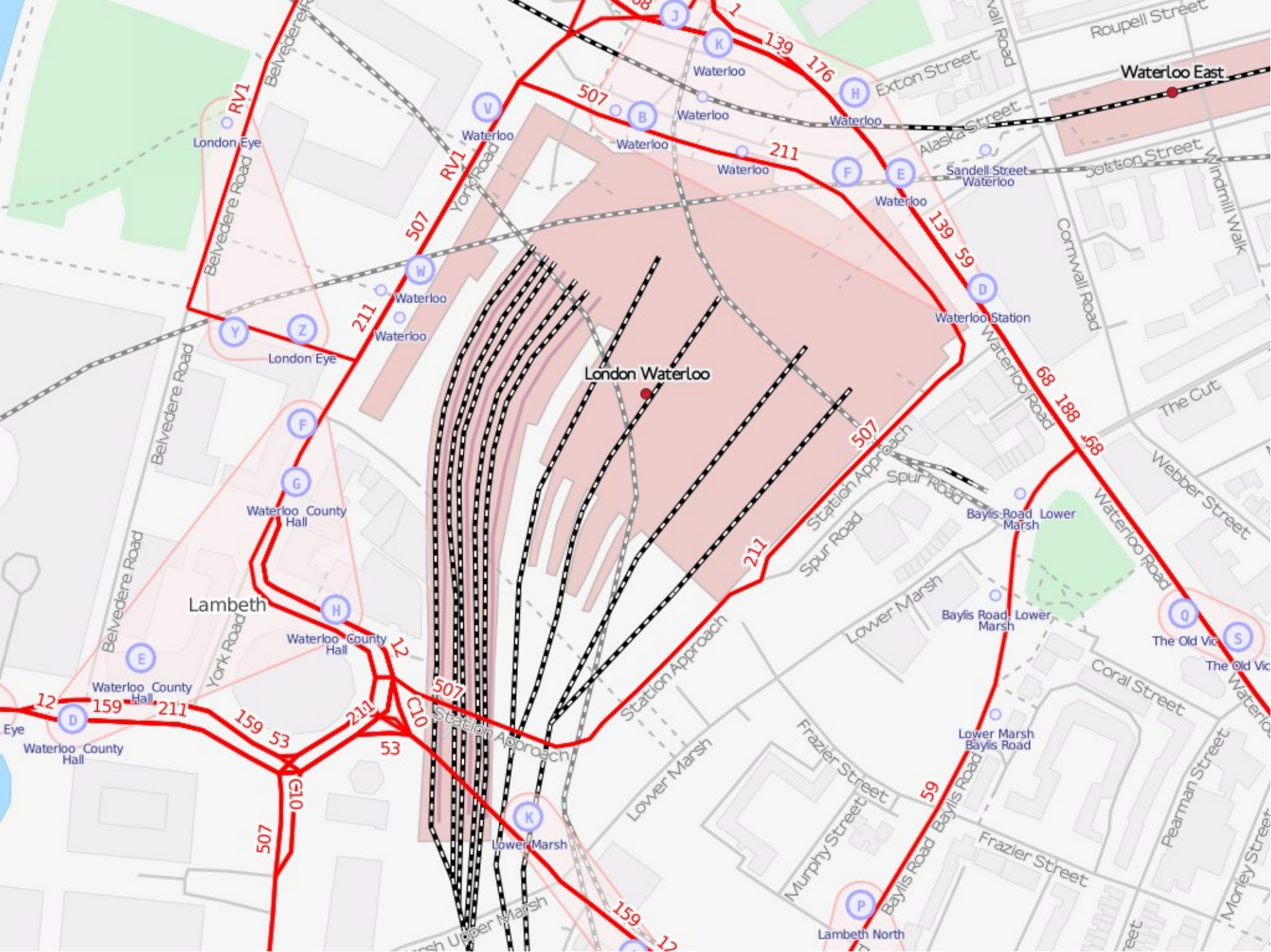
Extra detail for better rendering

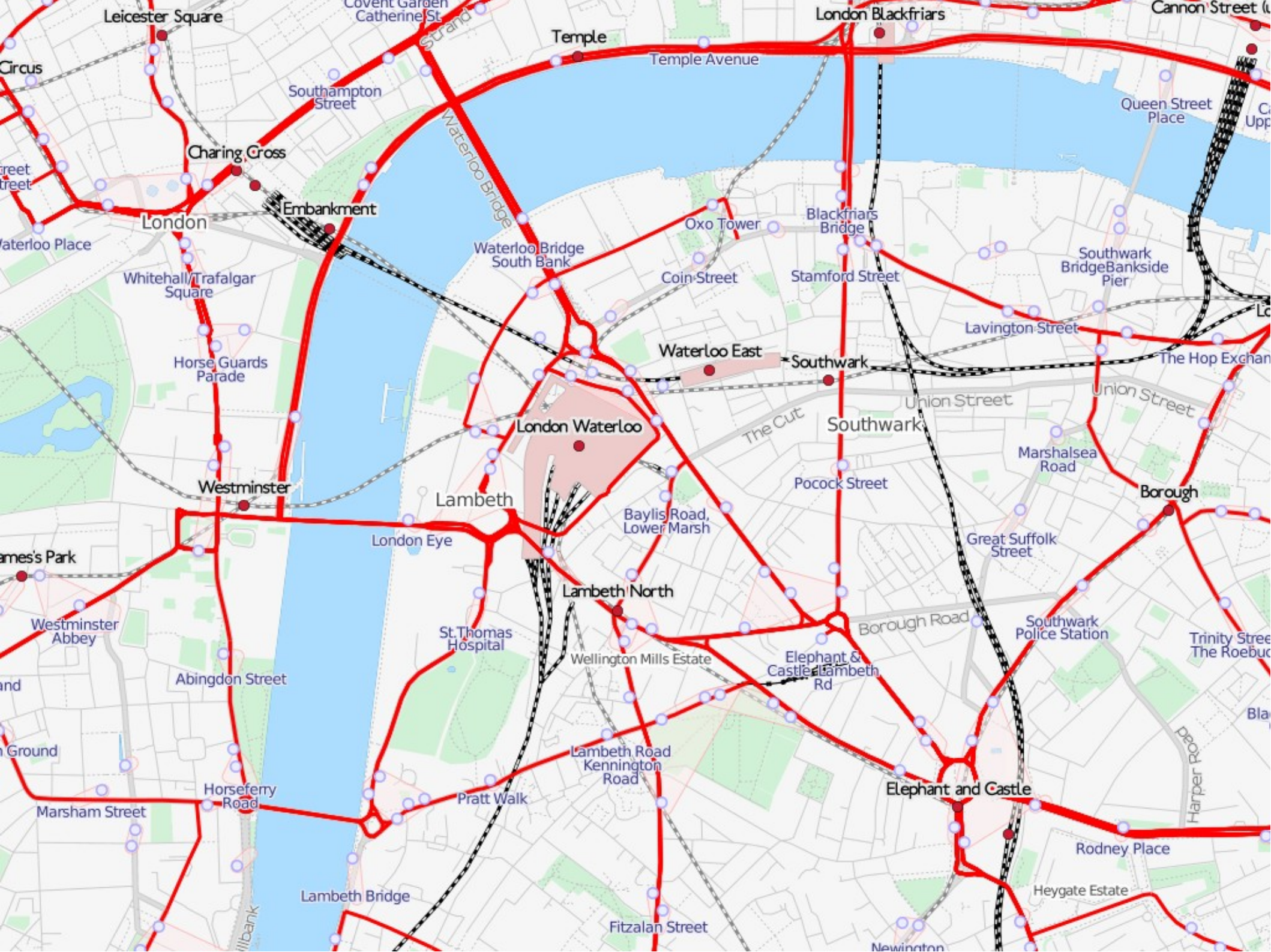




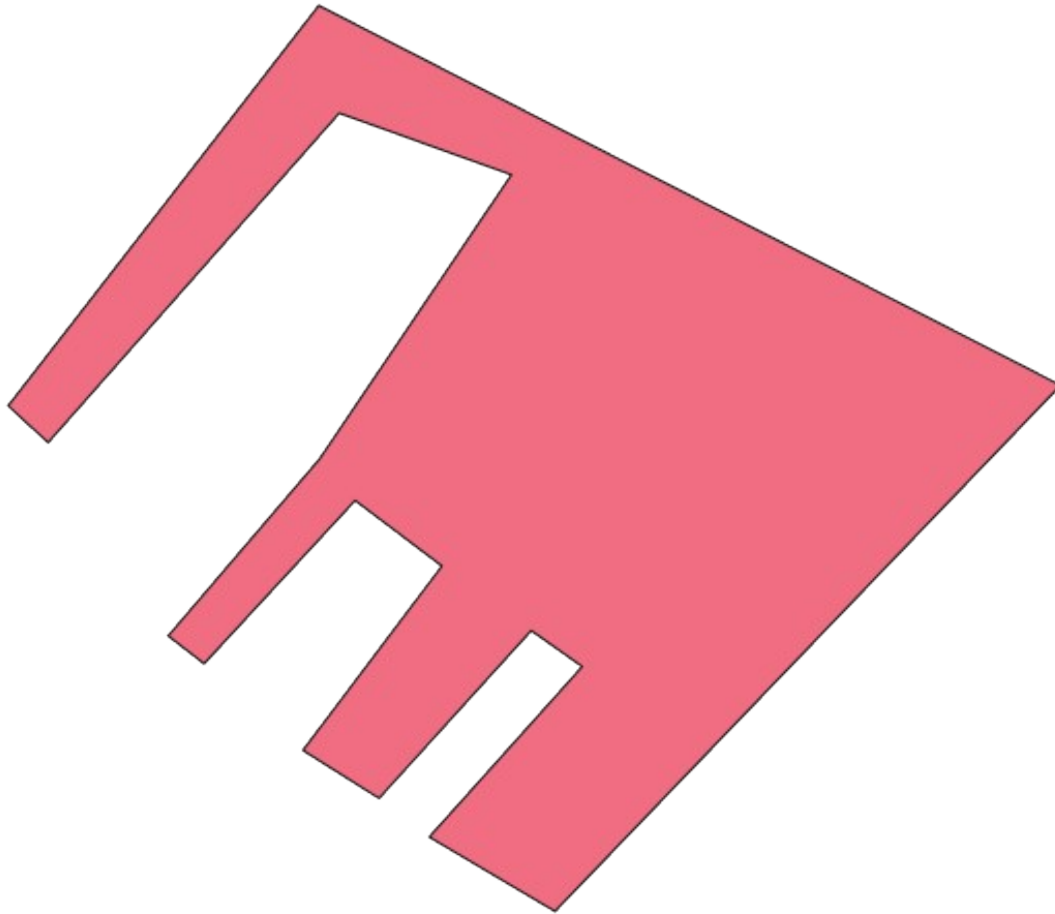


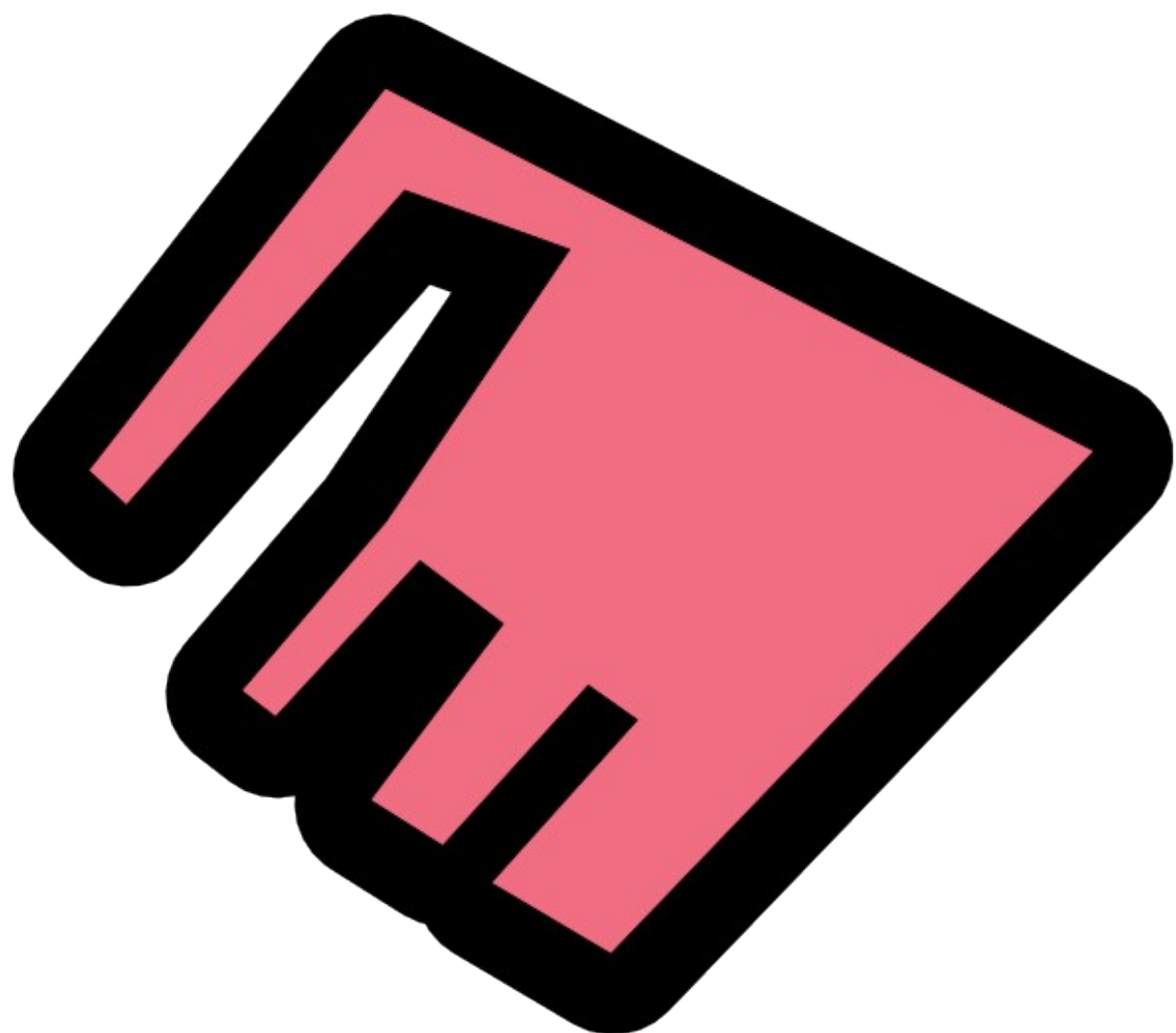
Buffering and Suchlike



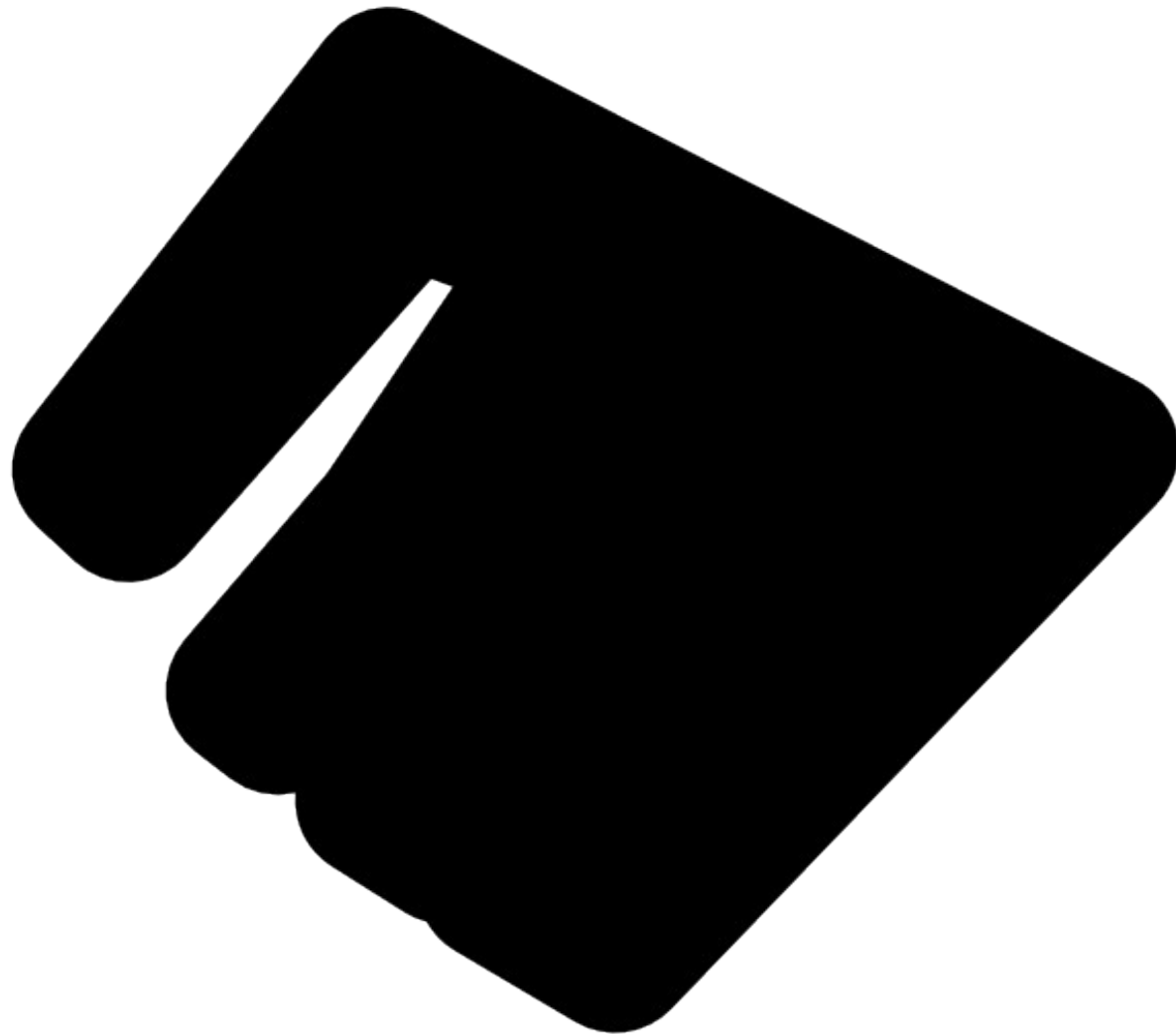


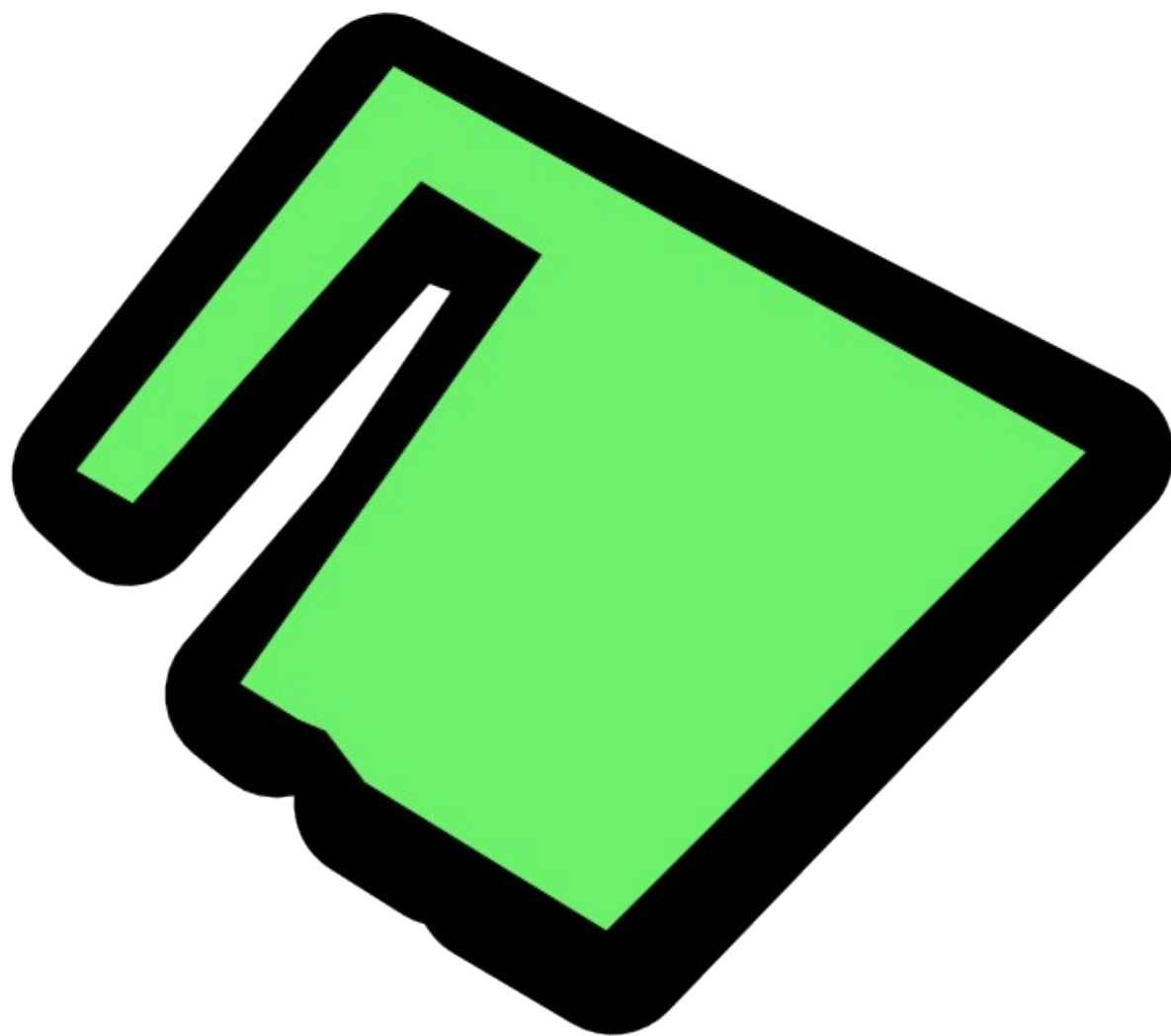
```
SELECT way FROM planet_osm_polygon WHERE building = 'train_station'
```



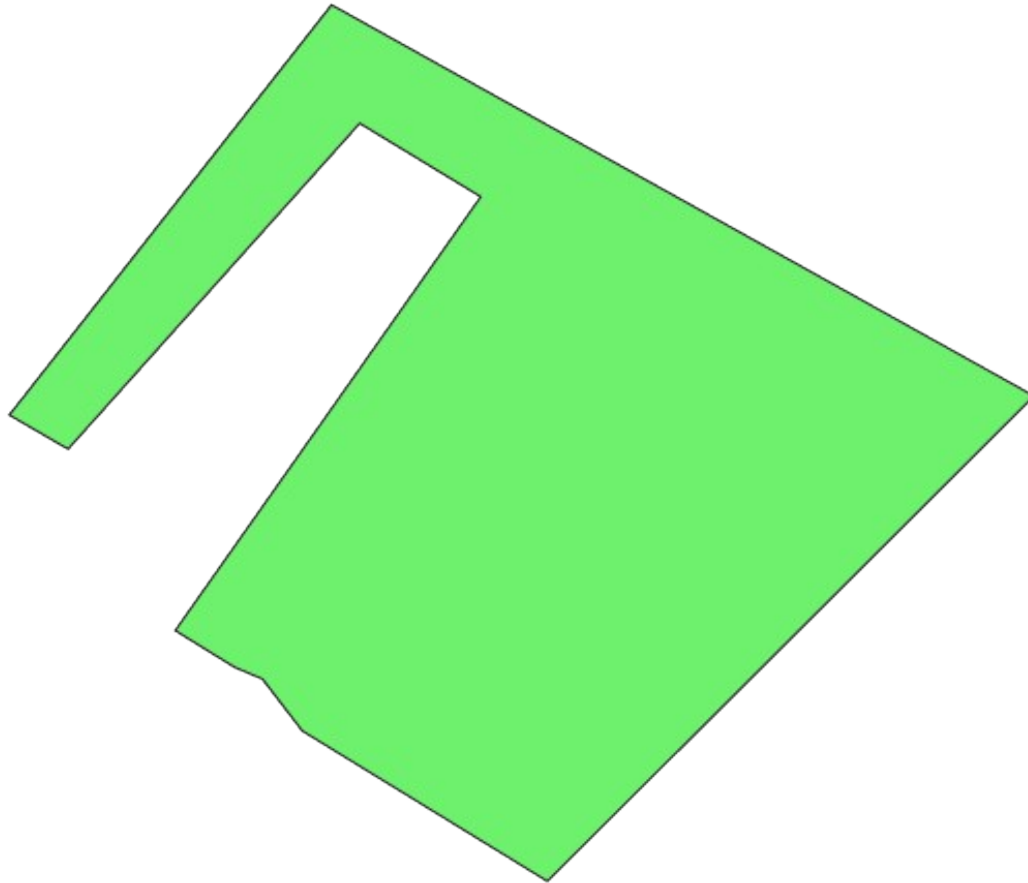


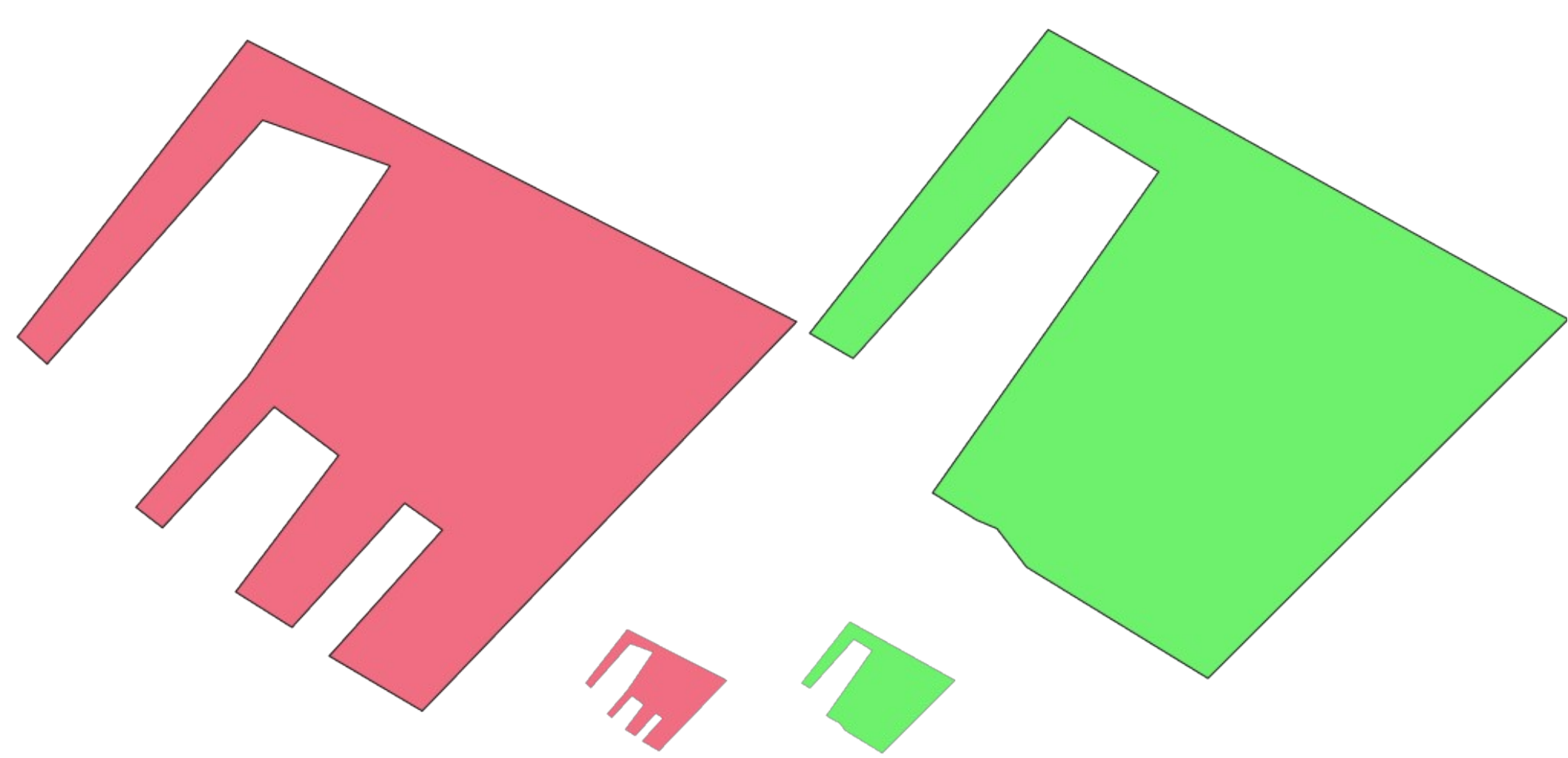
```
SELECT ST_Buffer(way, 10) FROM ...
```

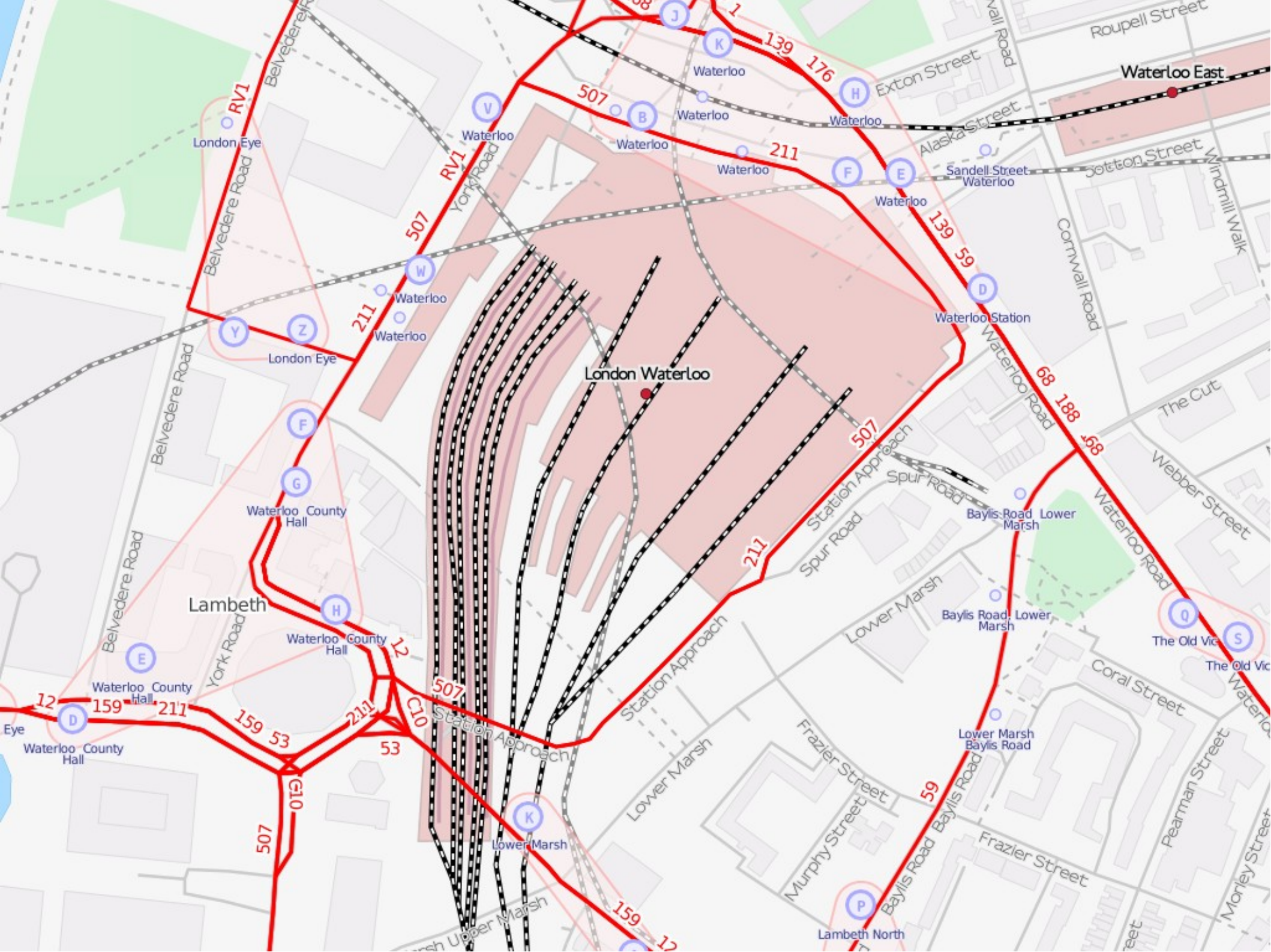


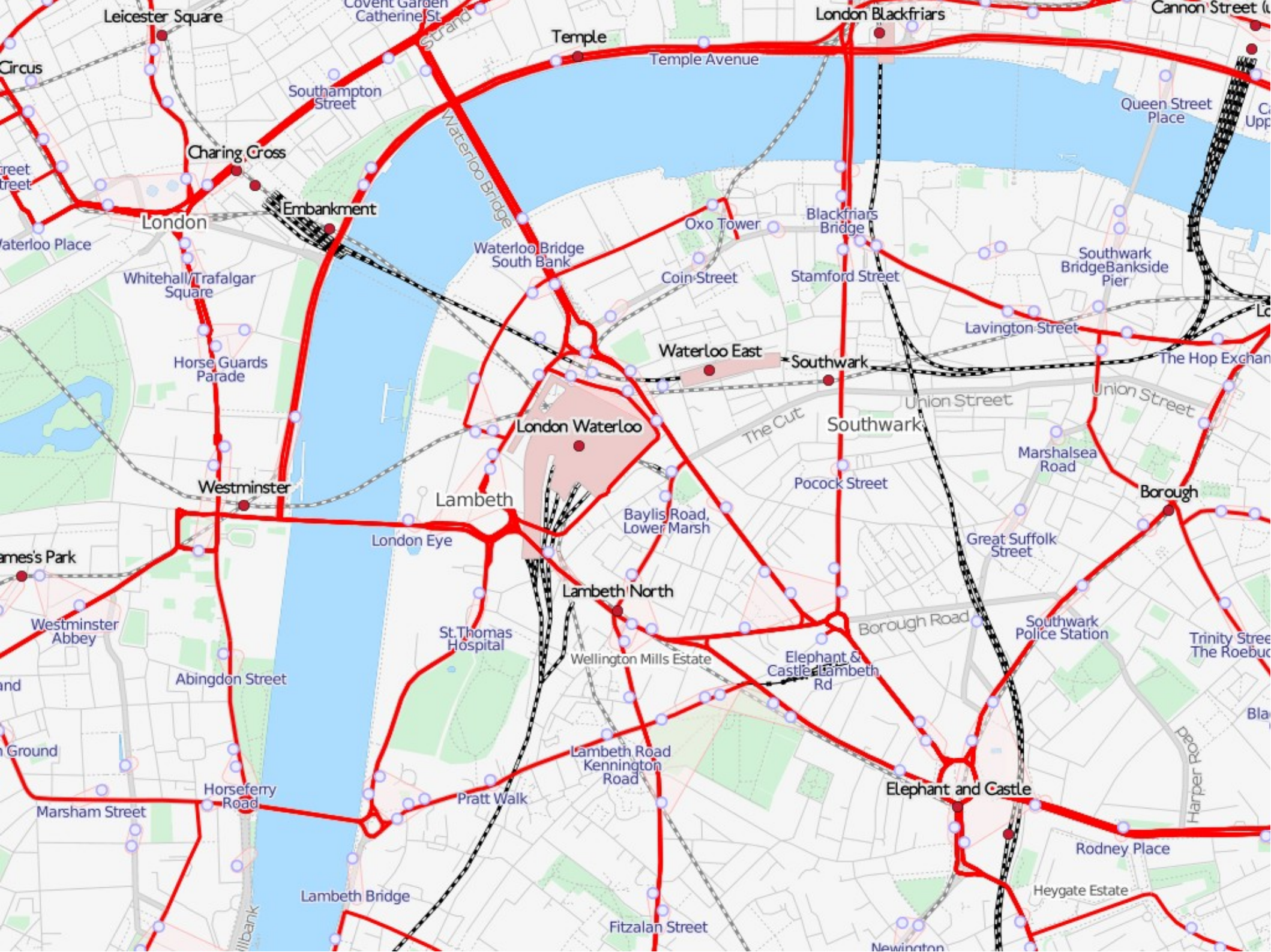


```
SELECT ST_Buffer(ST_Buffer(way, 10), -10) FROM ...
```

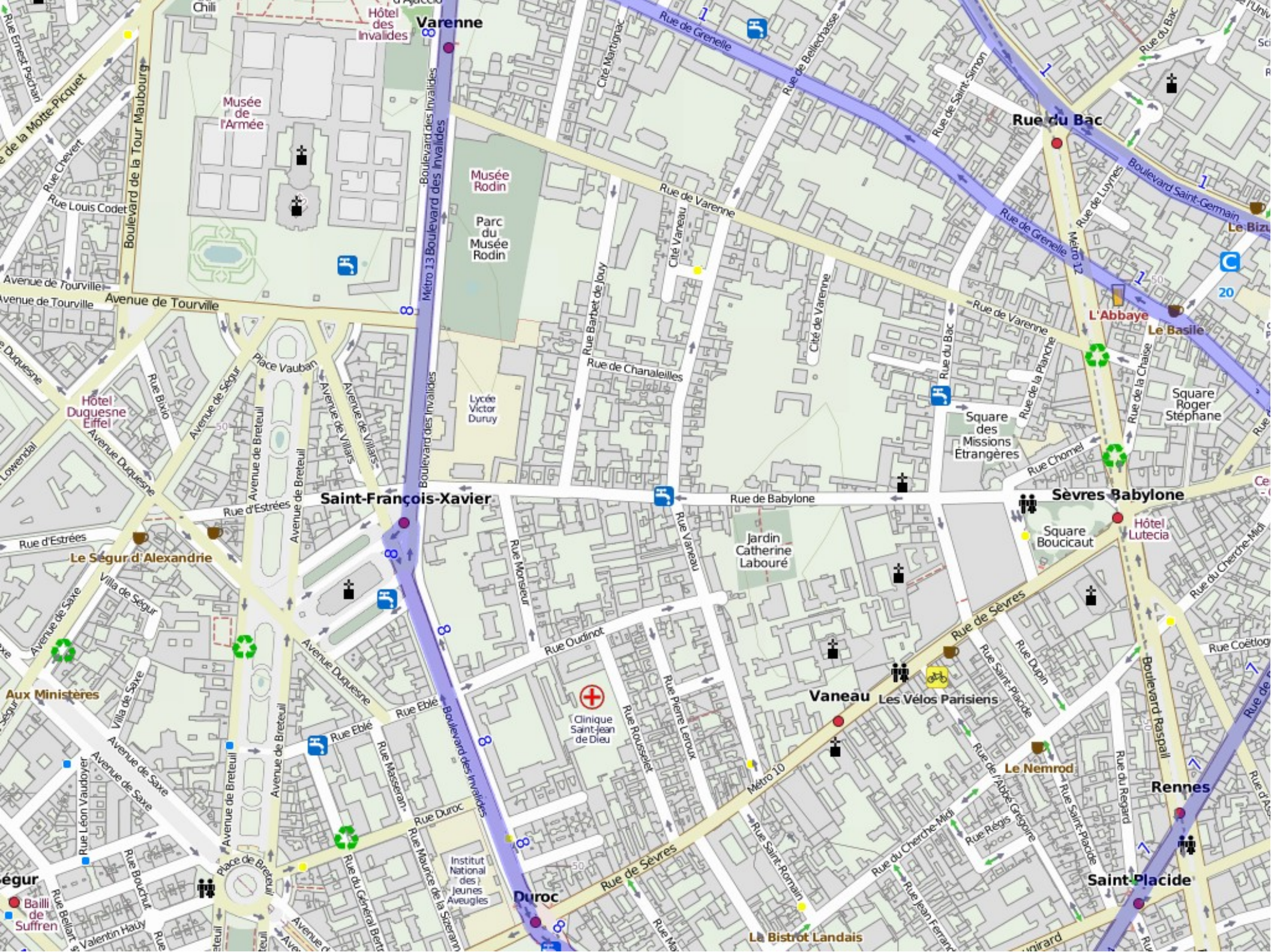




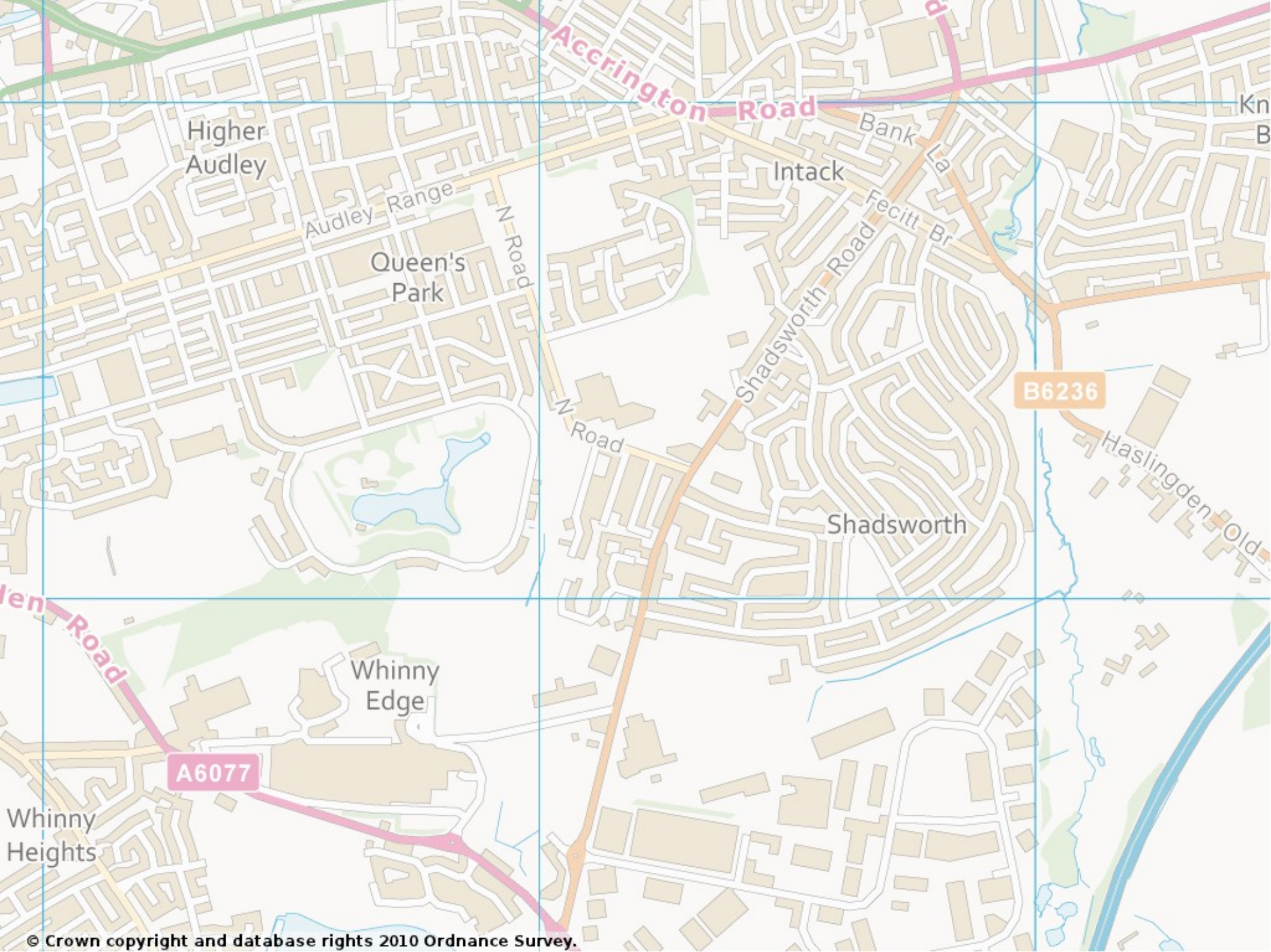




Merging Adjacent Polygons and stuff







Higher
Audley

Audley Range

Queen's
Park

Accrington Road

Bank

Intack

Fecitt Br

Shadsworth Road

Z Road

B6236

Haslingden Old

Shadsworth

Whinney Road

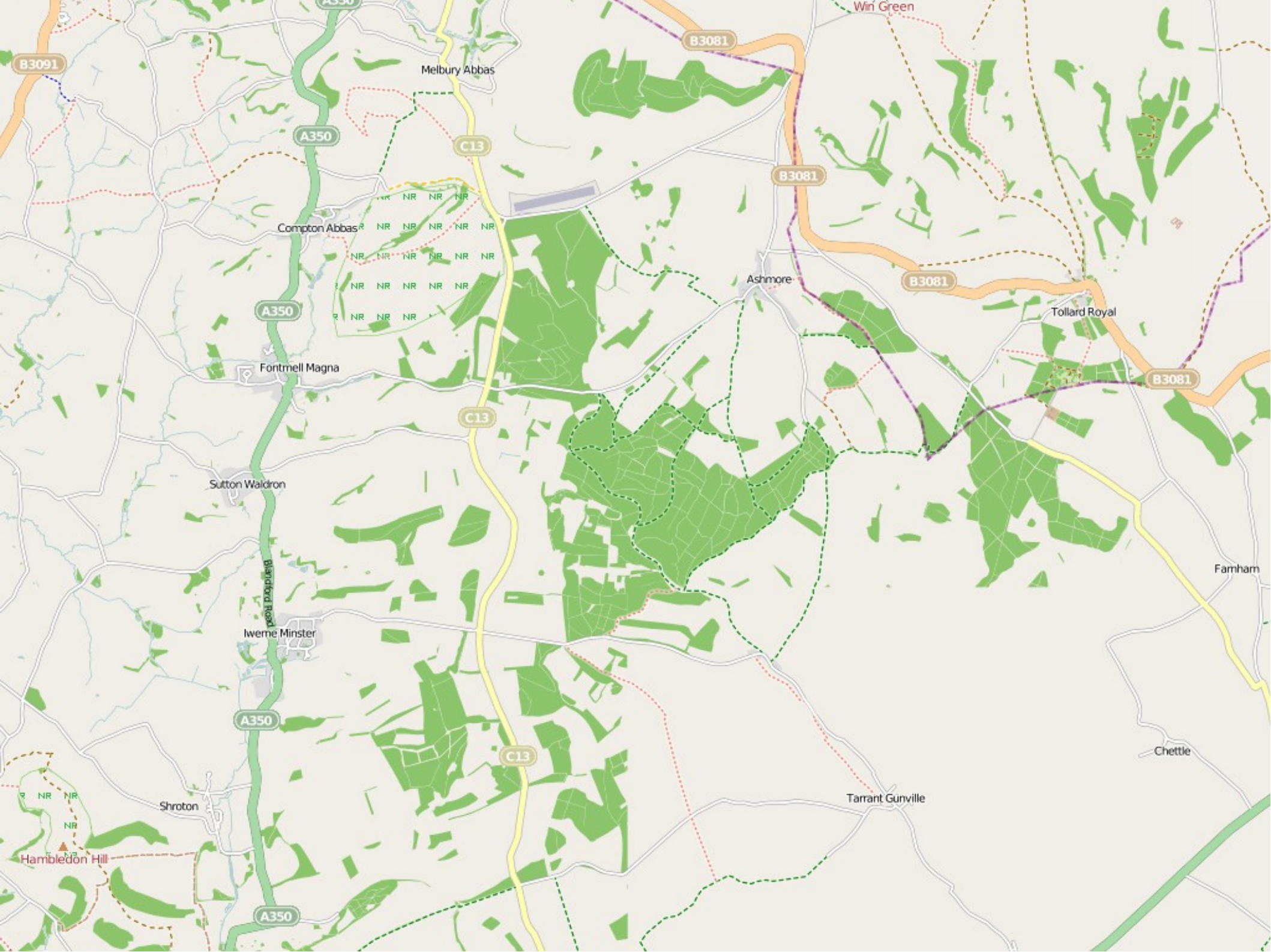
Whinny
Edge

A6077

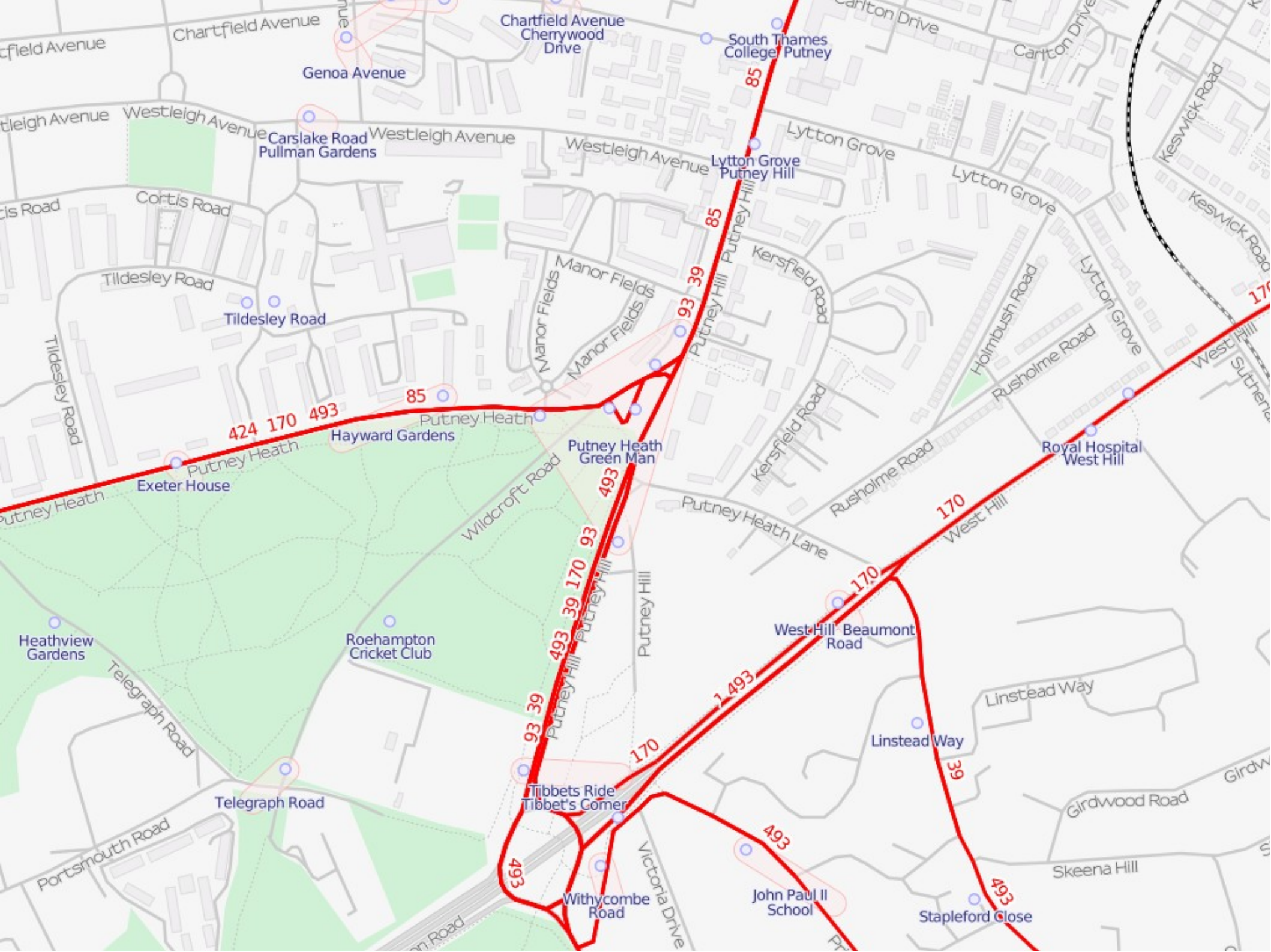
Whinny
Heights

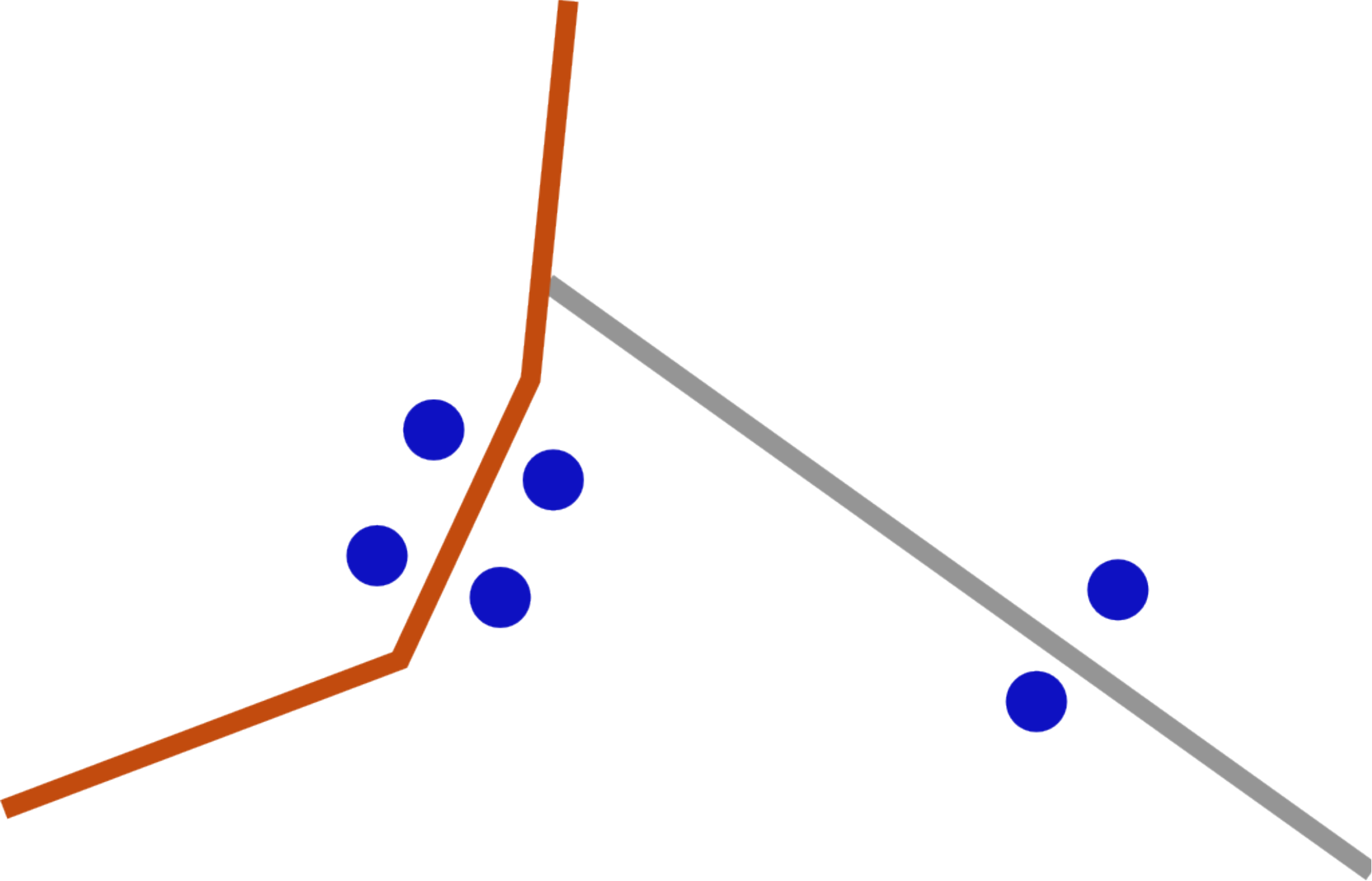
Dealing with import artefacts





bus stop zones





```
SELECT way FROM planet_osm_point  
WHERE highway = 'bus_stop'
```



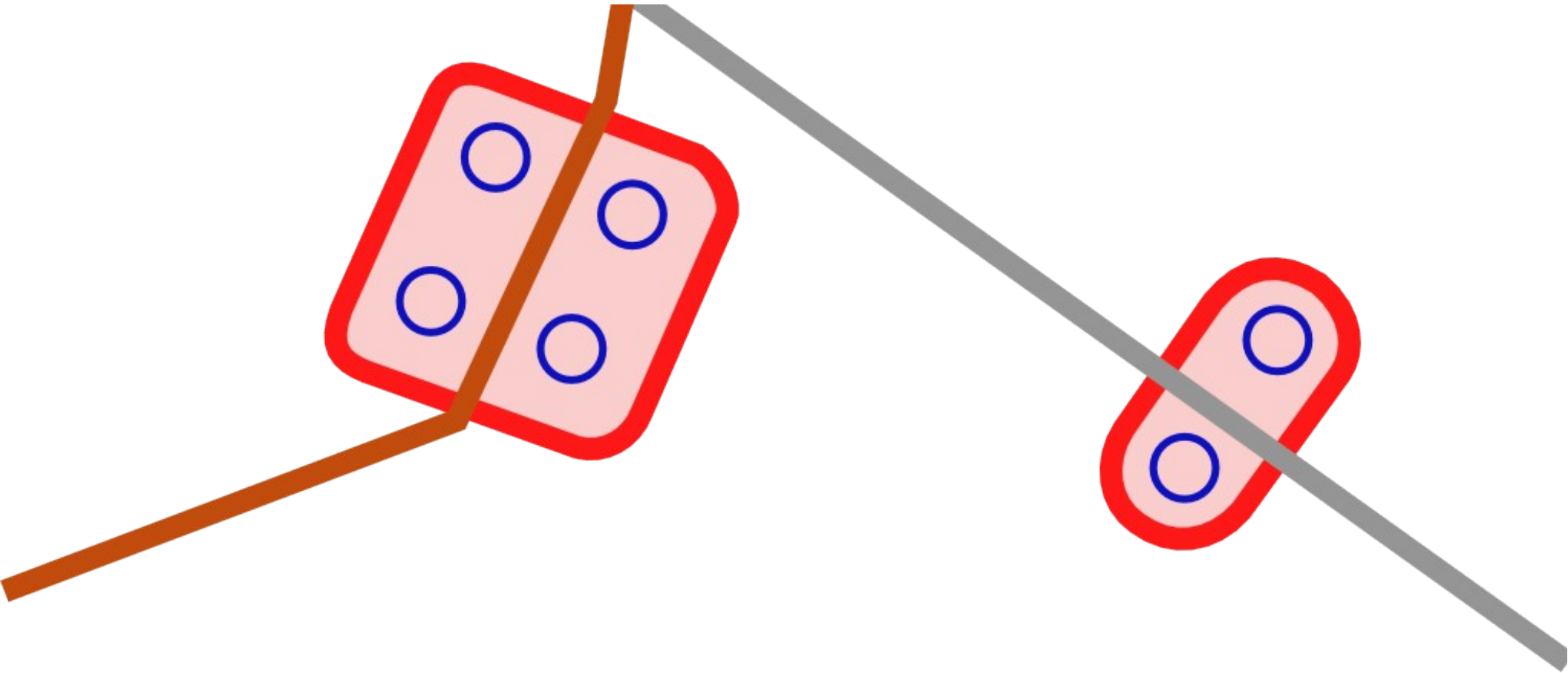
```
SELECT ST_Collect(way) FROM planet_osm_point  
WHERE highway = 'bus_stop'  
GROUP BY name
```

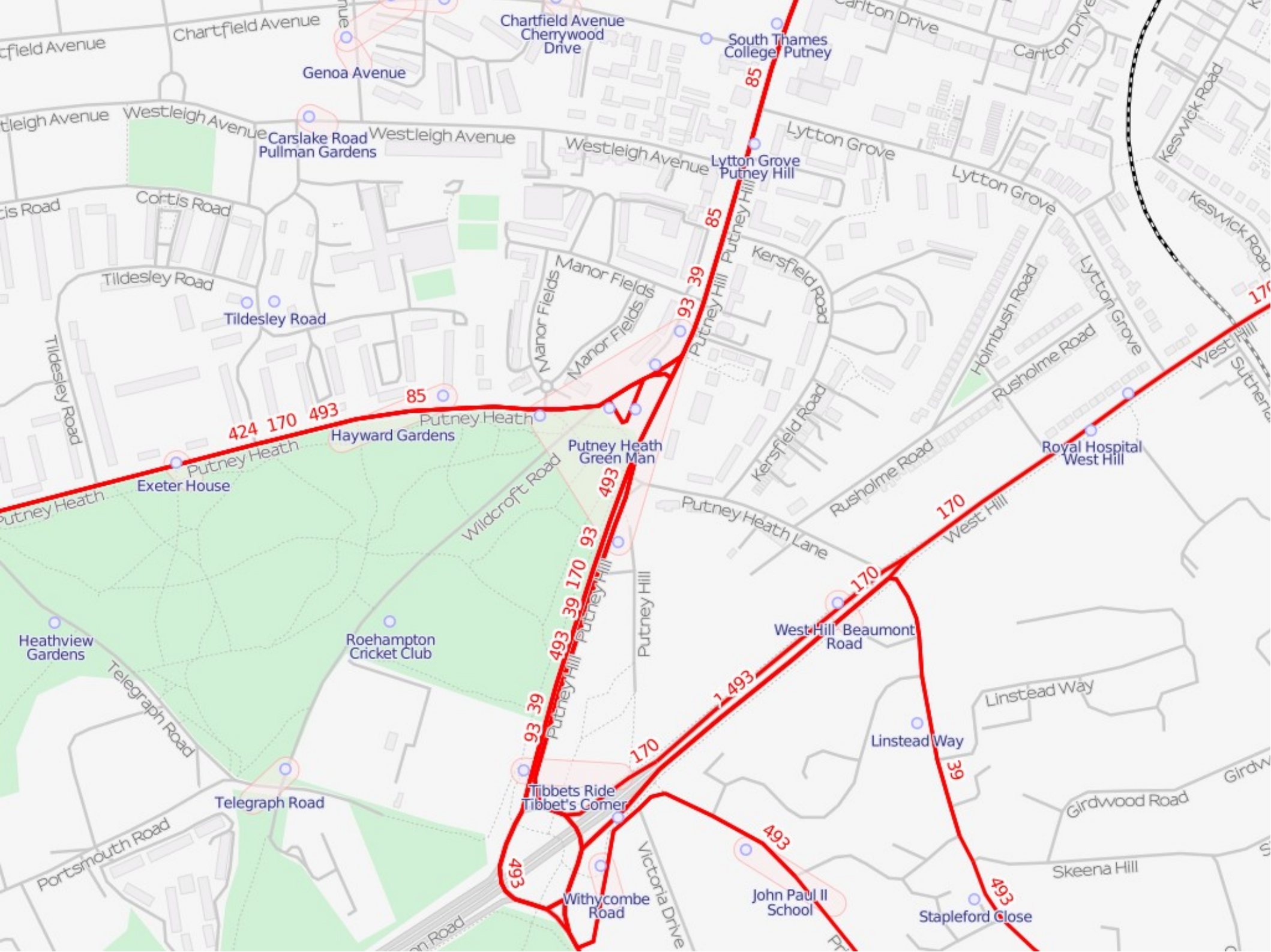


```
SELECT ST_ConvexHull(ST_Collect(way))  
FROM planet_osm_point  
WHERE highway = 'bus_stop'  
GROUP BY name
```



```
SELECT ST_Buffer(ST_ConvexHull(ST_Collect(way)),25)  
FROM planet_osm_point  
WHERE highway = 'bus_stop'  
GROUP BY name
```



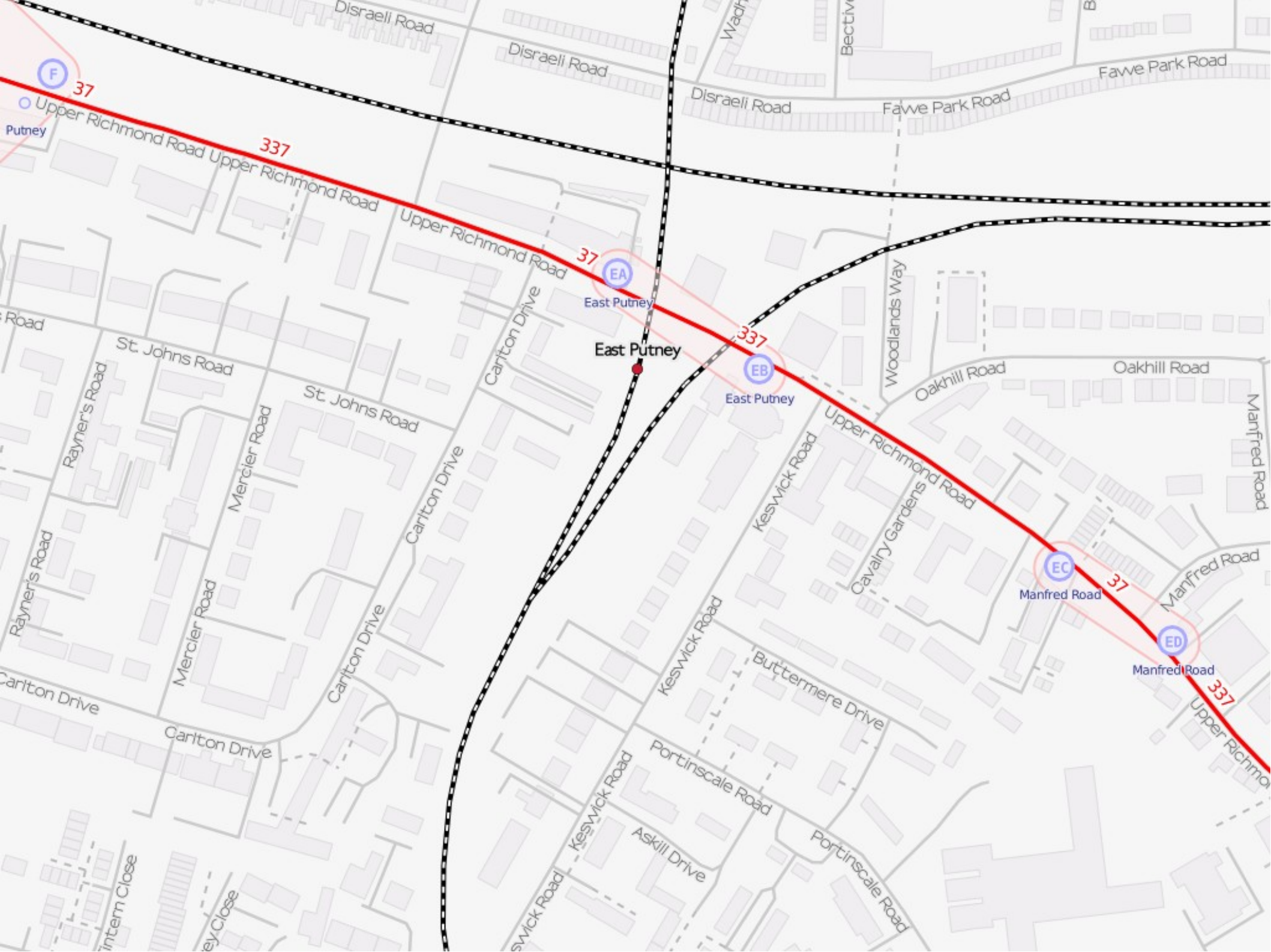


Convex Hull vs. Concave Hull

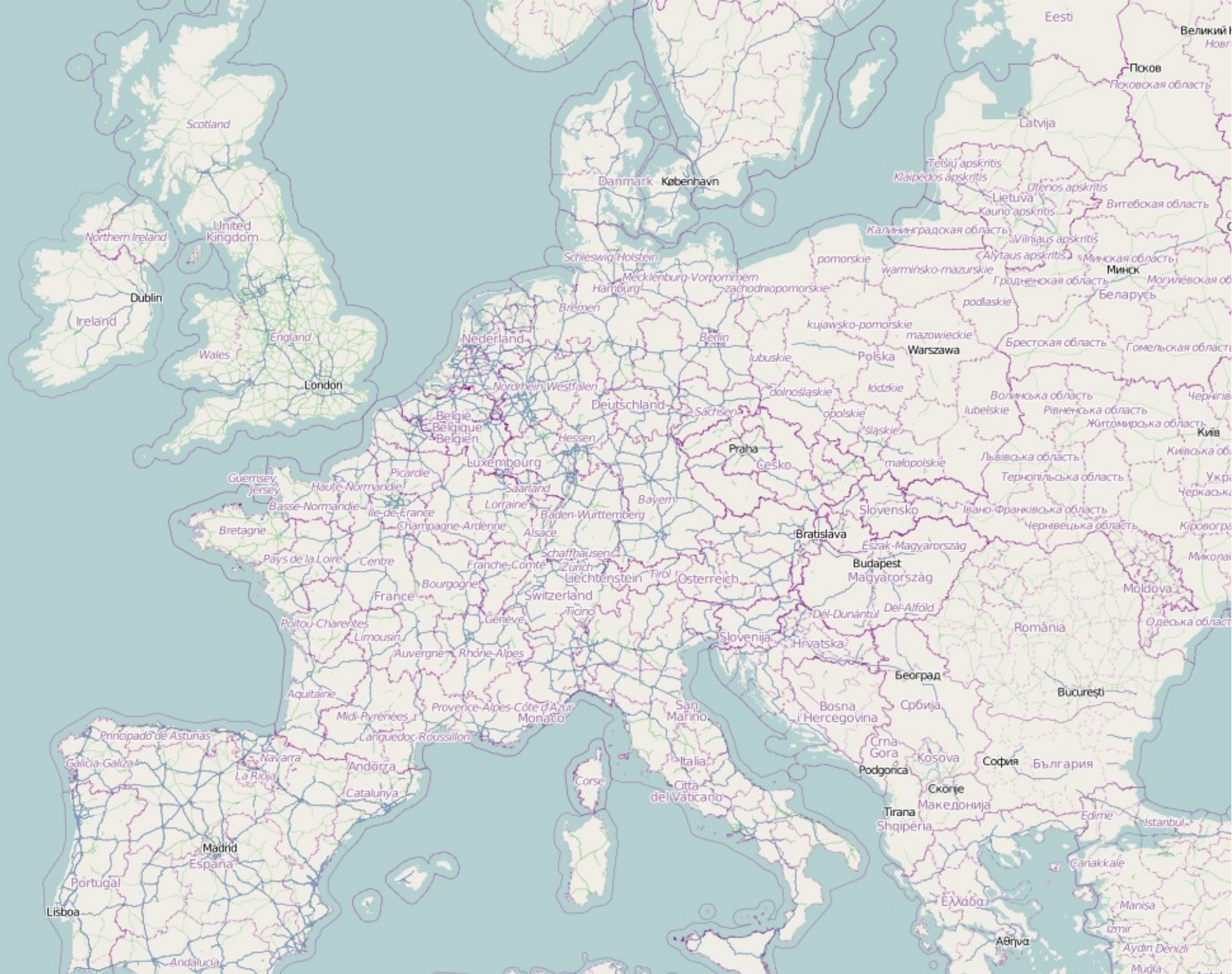


http://www.bostongis.com/postgis_concavehull.snippet

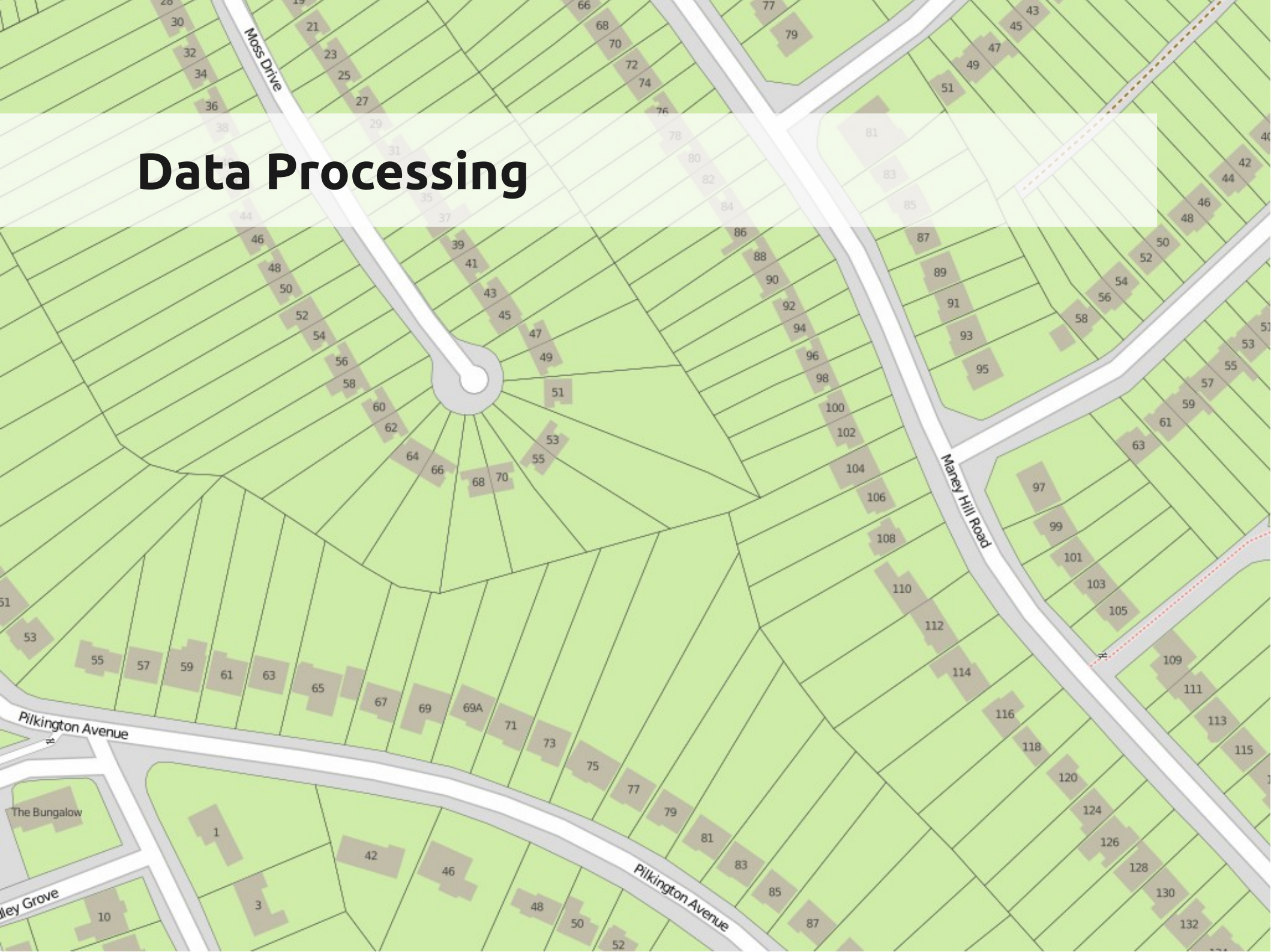
bus routes names

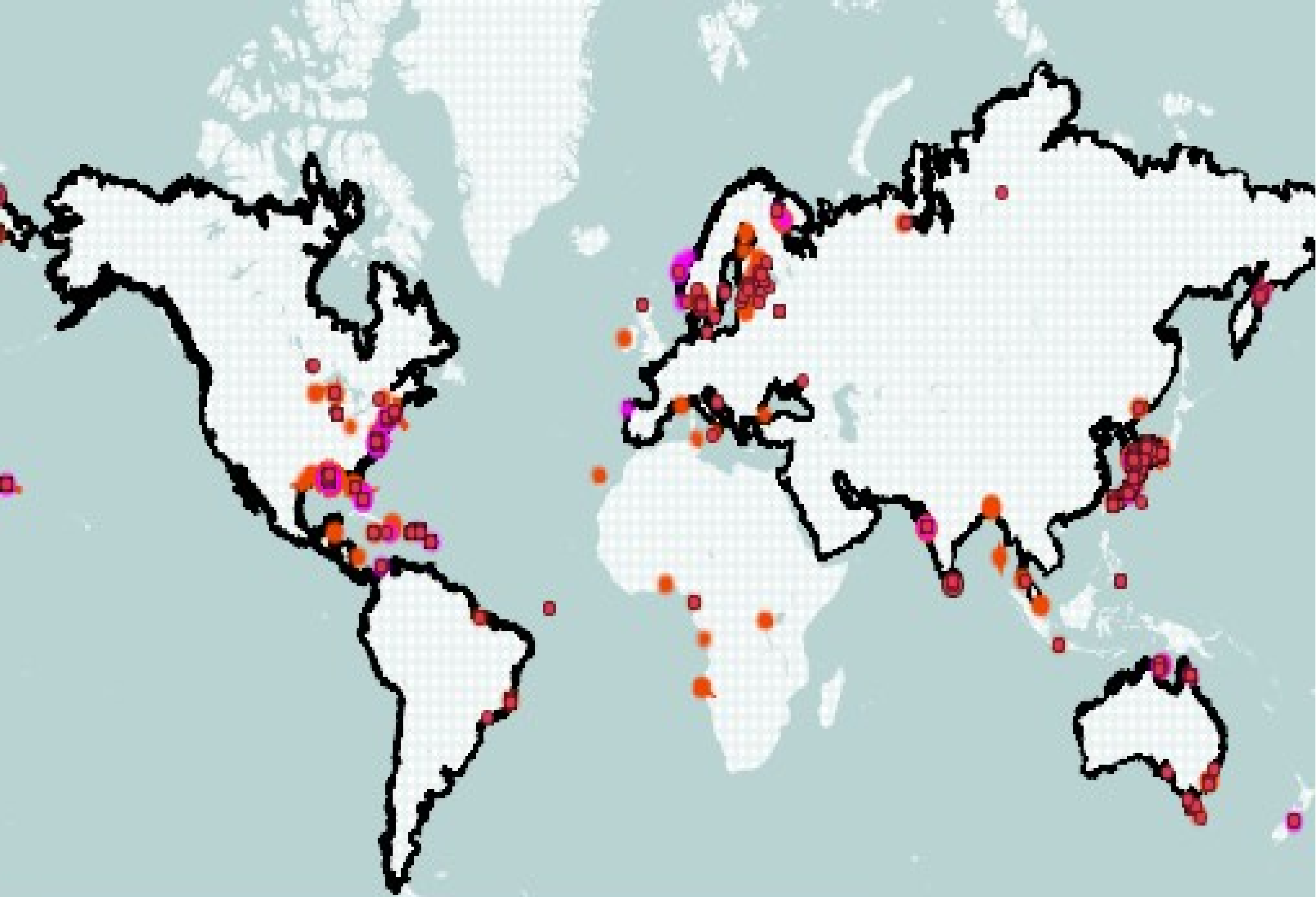


Squiggles hate dashes and transparency



Data Processing



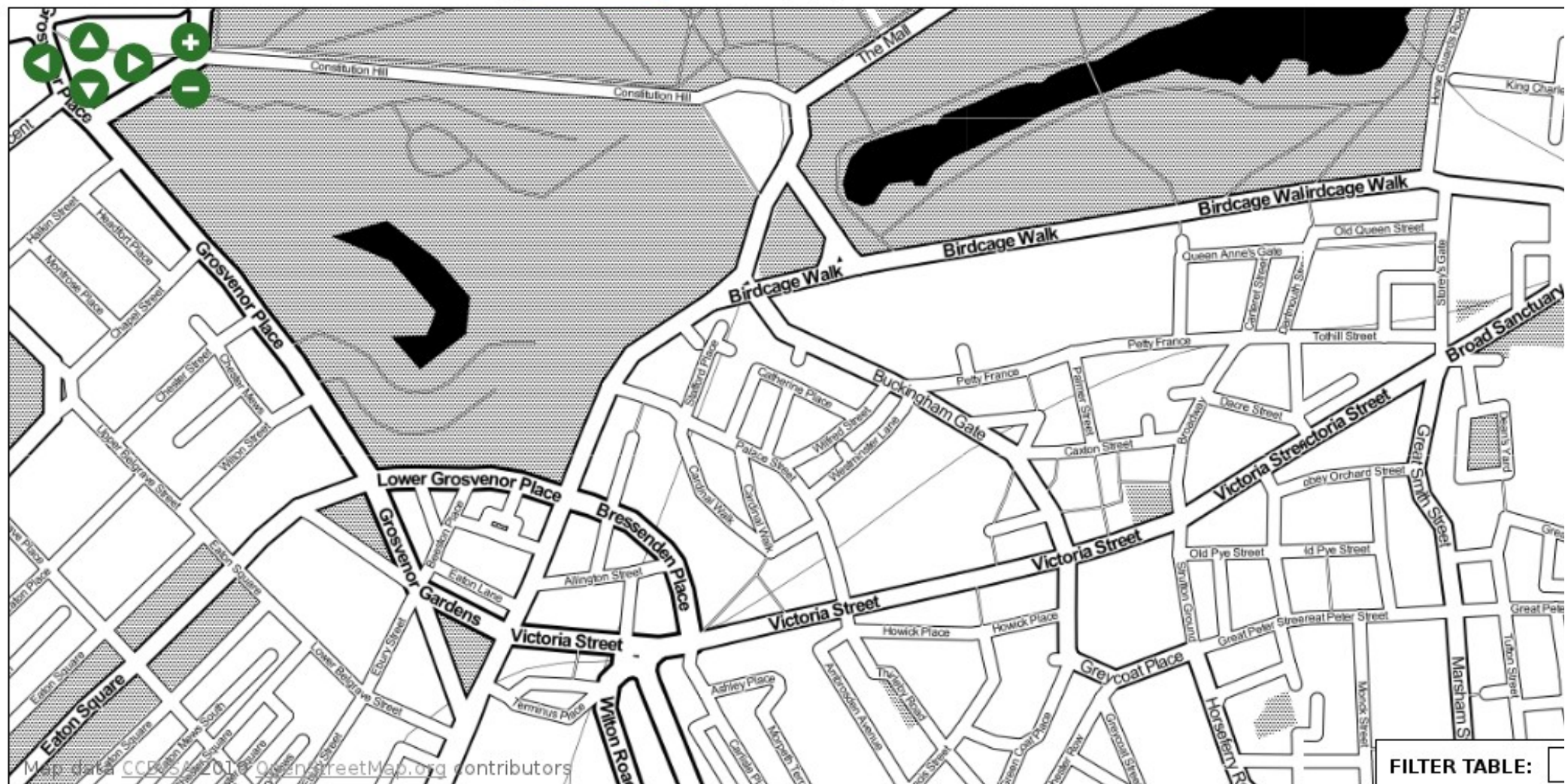


Legend: ● Big gap, ● Small gap, ● Self-intersection, ● Inverted polygon

Dotspotting

Dots on the pavement from flickr by [ericrodenbeck](#) created on July 8, 2011 at 02:43 PM

▼ Em



Map data © OpenStreetMap contributors

FILTER TABLE:

Dot	Coordinates	flickr:id	tags	ownername	datetaken	description	woeid	accuracy	tit
206733	37.768499,-122.417167	5810413500	blue dot	dotspotting	2011-05-13 20:01:51	—	55970976	16	IMG_2
206735	37.765166,-122.418834	5809848809	blue dot	dotspotting	2011-05-11 12:58:11	—	23512048	16	IMG_2



[map=yes] mapquest m^a

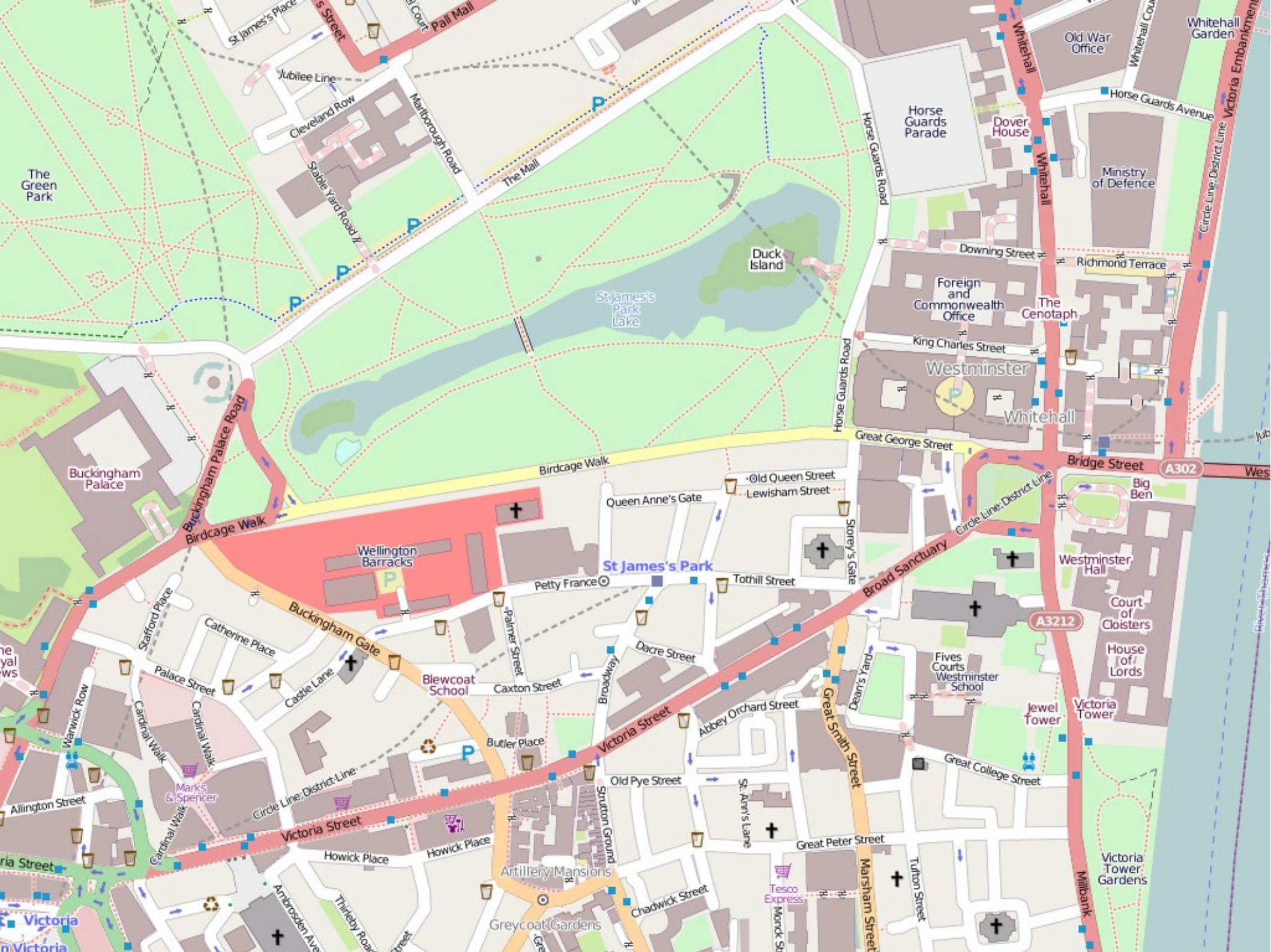
A new kind of map based on open data



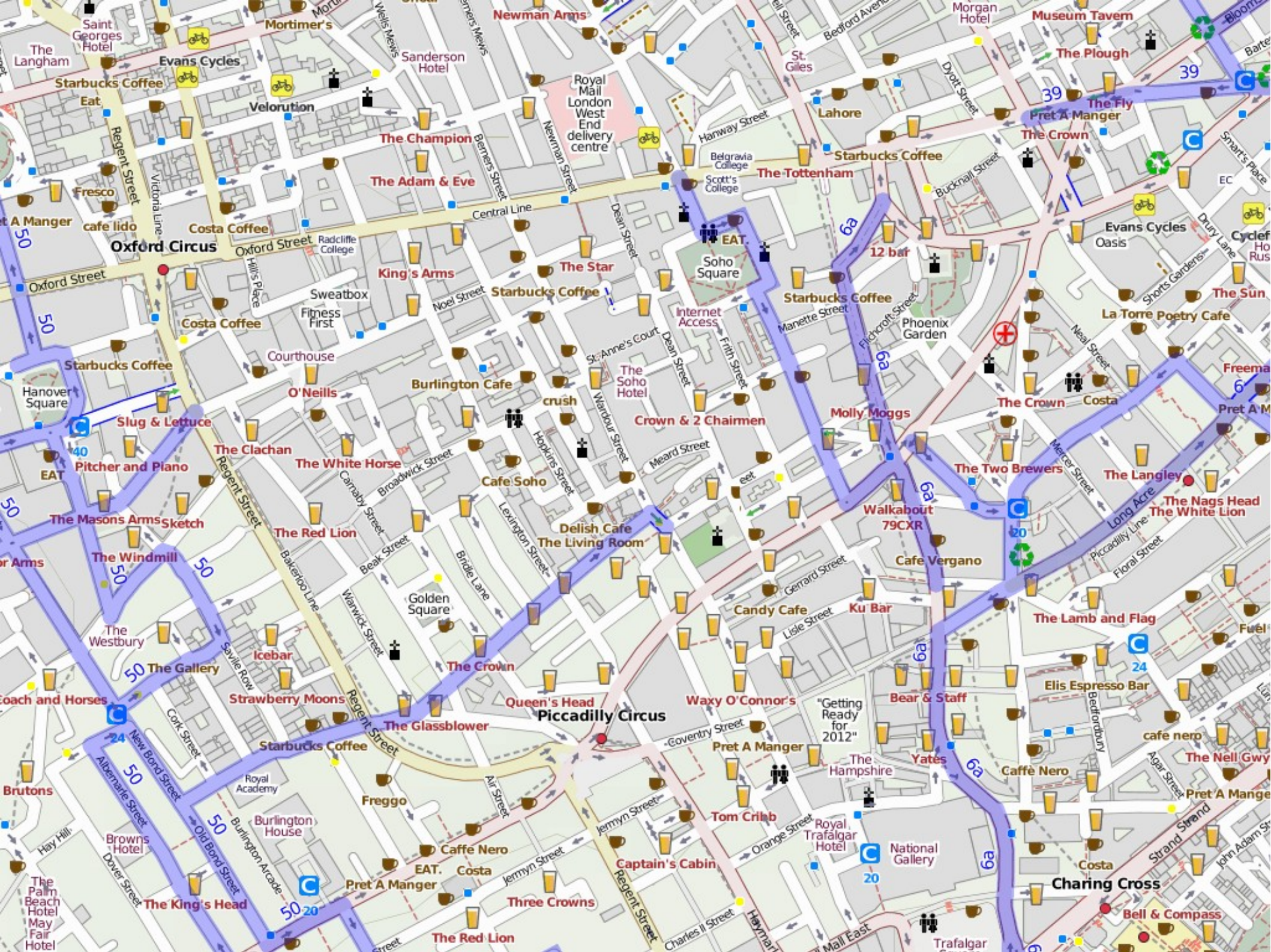
Most online maps are designed to help you get around in a car. This generally means displaying: roads, businesses, buildings, on-ramps, parks, oceans and traffic congestion. Nothing wrong with that! Designers get handed a tool kit that has as many tools as a good swiss army knife, and the maps reflect these tools. Millions of people use them to make appointments across town, find restaurants, and drive home for the holidays.

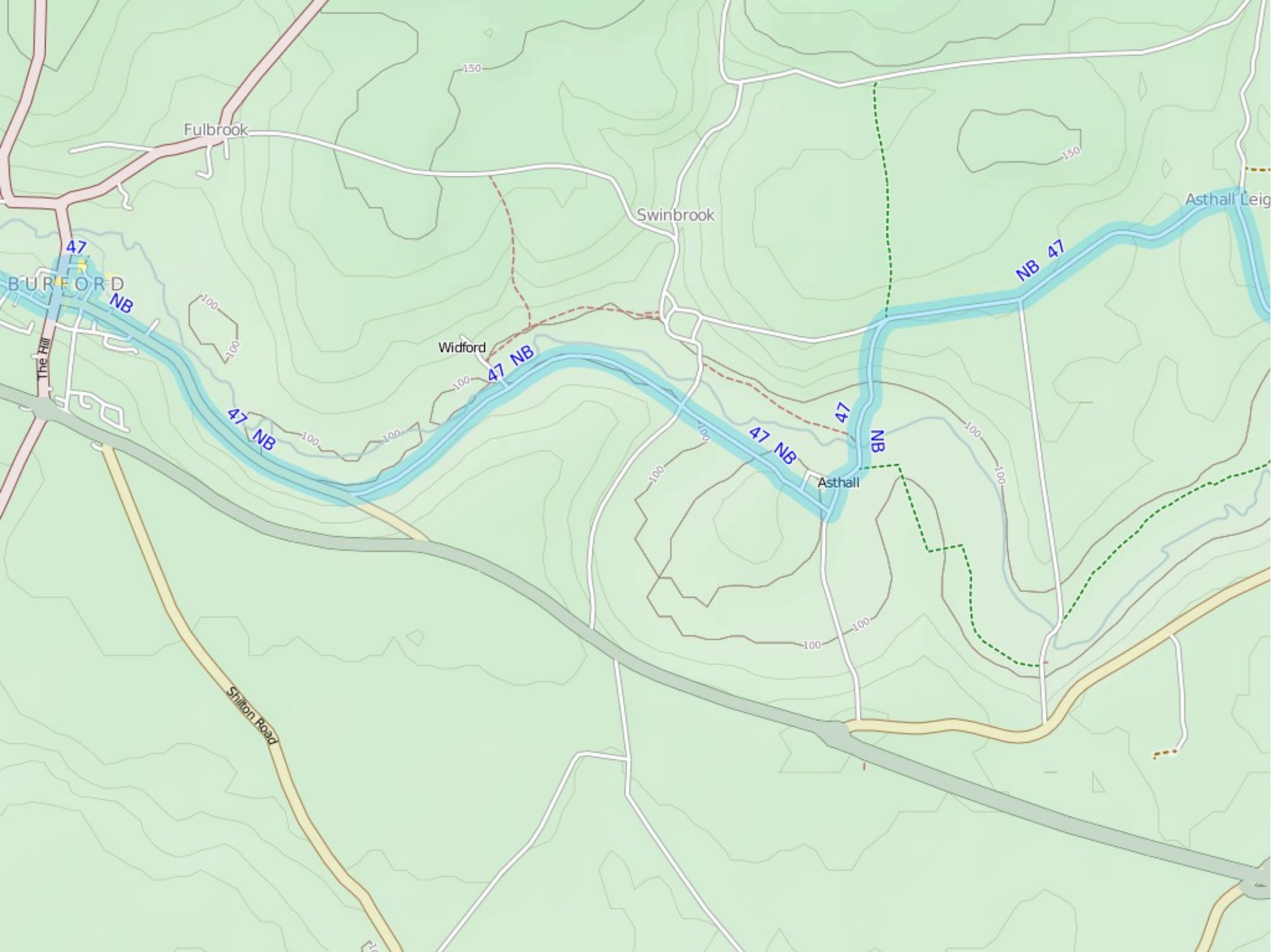
But what if, instead of a swiss army knife, we used a box of crayons? Or charcoal and newsprint? Or play-doh? What would those maps look like? What could they tell us about the world?





POI Density





Text Processing

1	2	3	4	5	6	7	8
Long1		Long4		Long6			

Long2		S3	Long5		Long7		S8
1	2	3	4	5	6	7	8
Long1		Long4		Long6			

East Birmingham Street
 East Birmingham St
 E Birmingham St
 E. B'ham St
 E. B'ham

Skeletons

Rule 2: IF a branch that leaves from a junction point does not leave the circle around its origin THEN eliminate it.

Applying these rules finally reduces the skeleton to the centrelines (see figure 8).



Figure 6. Aggregation of parcels declared as roads in the ALK

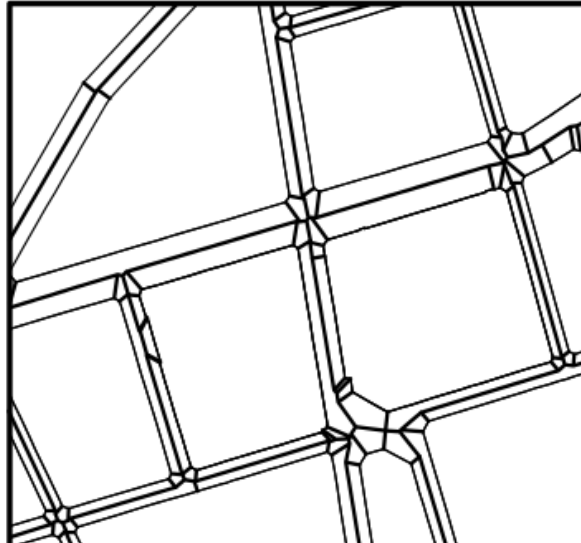


Figure 7. Straight Skeleton of aggregated road parcels

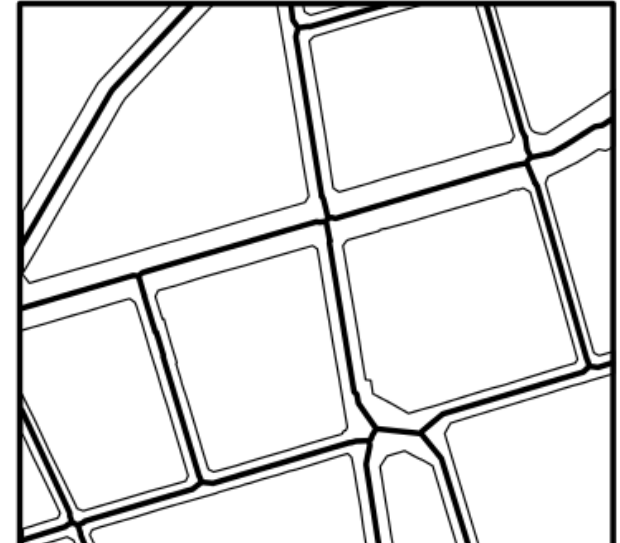


Figure 8. Centrelines of polygon

Still these centrelines differ from the optimal solution, since a junction does not appear as an intersection point of road axes. Instead of that, a centreline enters the surrounding of a junction in a rather unnatural arc, because road parcels widen in these areas. Also, at T-junctions, the straight line is bent into the direction of the incoming road. To solve this problem an algorithm has been developed which is able to change most of the junctions to the desired appearance. This algorithm mainly relies on rules which define when a junction point has to be repositioned and in which cases multiple junction points can be replaced by a single point. These rules are essentially based on a definition of junction areas. Basically, the new locations of the junction points are



👤 migurski / **Skeleton**

👁 Watch

🍴 Fork

👁 3

🍴 1

Source

Commits

Network

Pull Requests (0)

Issues (0)

Graphs

Branch: master

Switch Branches (1) ▾

Switch Tags (0)

Branch List

Computes straight skeletons of simple polygons — [Read more](#)

<http://skeleton.teczno.com:8206>

📄 Downloads

HTTP

Git Read-Only



Read-Only access

Wrote a `linemerge()` that works



Michal Migurski (author)
February 20, 2011

commit [c0e2699d9a98c5fd4ed9](#)
tree 63f20b04fd5c056fa02a
parent [3f12a432423ce2af7f81](#)

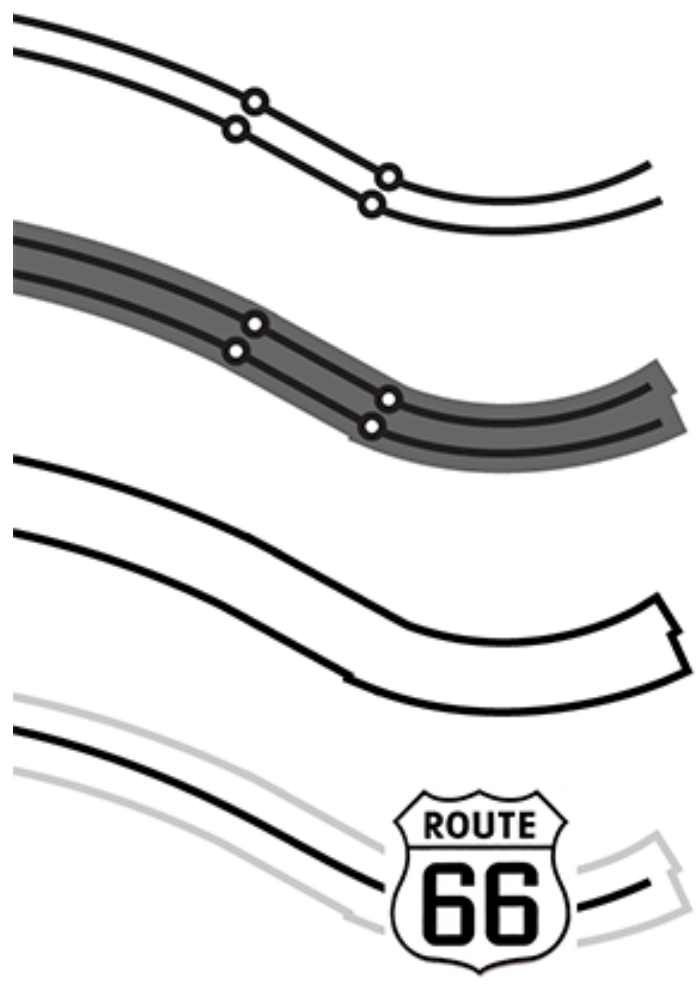
Skeleton /

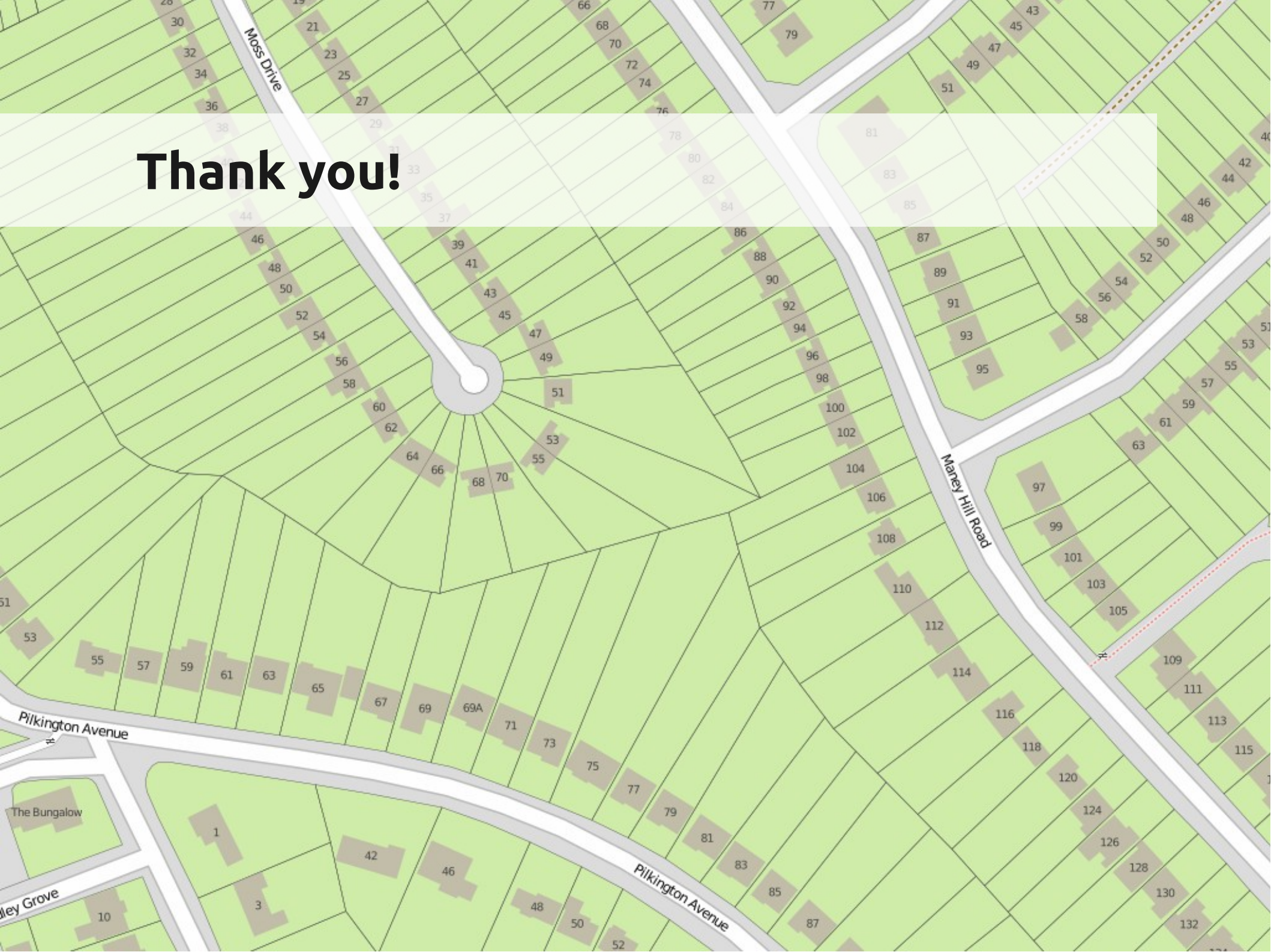
name	age	message	history
📁 CGAL/	February 20, 2011	Wrote a <code>linemerge()</code> that works [Michal Migurski]	
📁 Skeleton/	December 27, 2010	Dropped threshold for tiny angles still further to... [Michal Migurski]	
📄 .gitignore	December 19, 2010	Made old sketch-skeleton into more-proper Skeleton... [Michal Migurski]	
📄 README.txt	December 19, 2010	Tiny readme [Michal Migurski]	
📄 skeleton-test.py	December 27, 2010	... [Michal Migurski]	

README.txt

The straight skeleton:

<http://twak.blogspot.com/2009/01/that-straight-skeleton-again.html>





Thank you!