Hochschule für Technik und
Wirtschaft Dresden
University of Applied Sciences

Public Transport Stop Analysis for OSM and Welzl's Algorithm on the Sphere

Jens Flemming

jens.flemming@htw-dresden.de

https://ptsa.io.informatik.htw-dresden.de

State of the Map, OSM Science, Champs-sur-Marne (France), August 2026

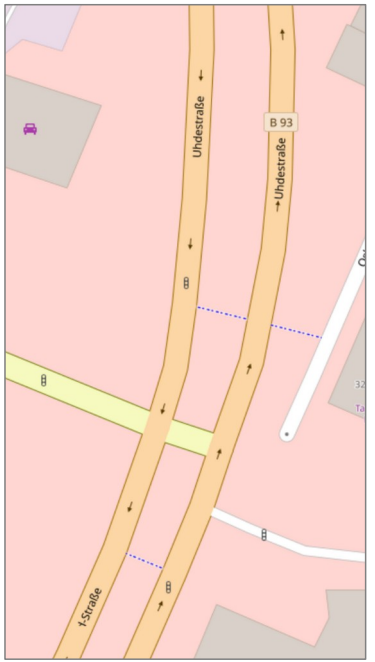
Public transport stops in OSM

2024: add a bus stop to OSM...

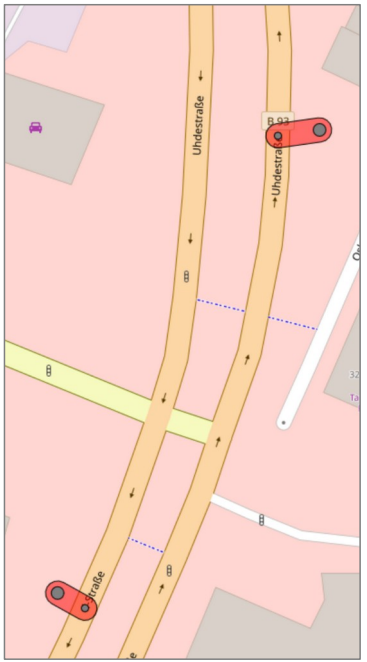
How to do that?
5 mappers, 7 solutions
(maybe the ratio is more like 5/20 for PT in OSM)

Visualize PT stop mapping styles
(and find data quality issues)
(and group OSM objects to stops)

Starting with a few cities at Germany...
How to upscale analysis to the whole planet?
About 4 million PT stops on the OSM planet!



Carto

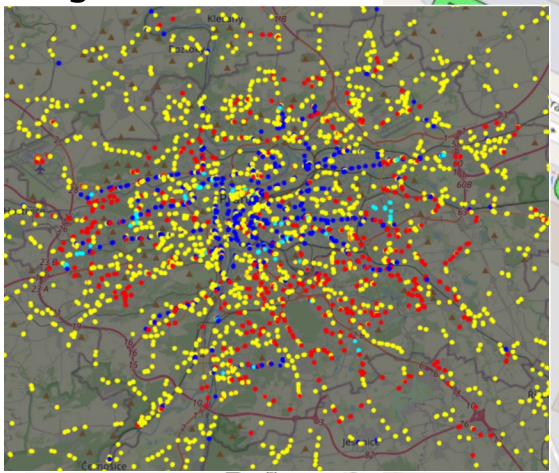


PTSA

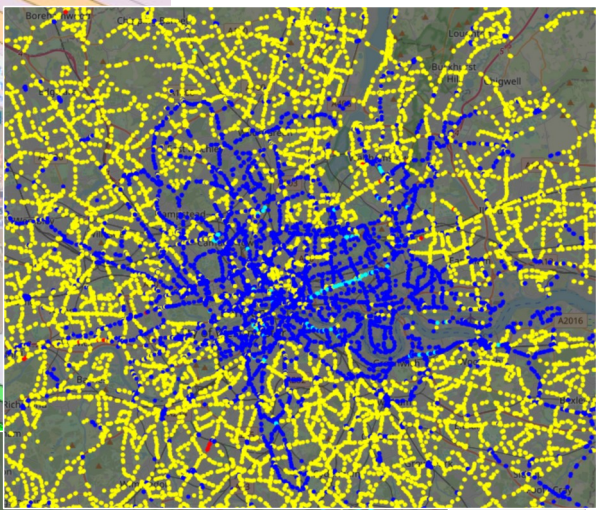
PTSA - Public Transport Stop Analysis: https://ptsa.io.informatik.htw-dresden.de

PTSA

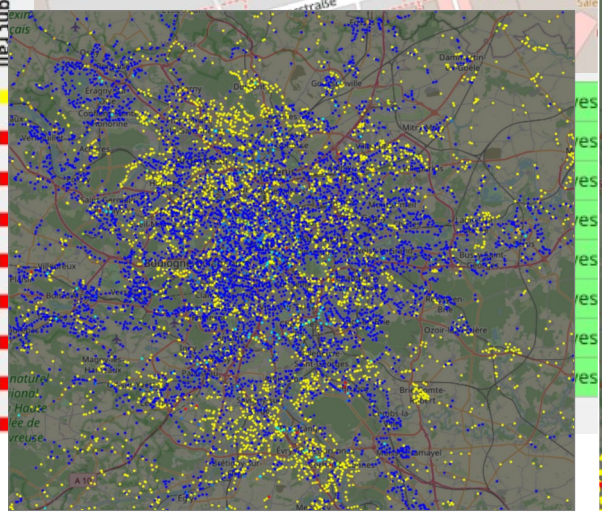
Prague

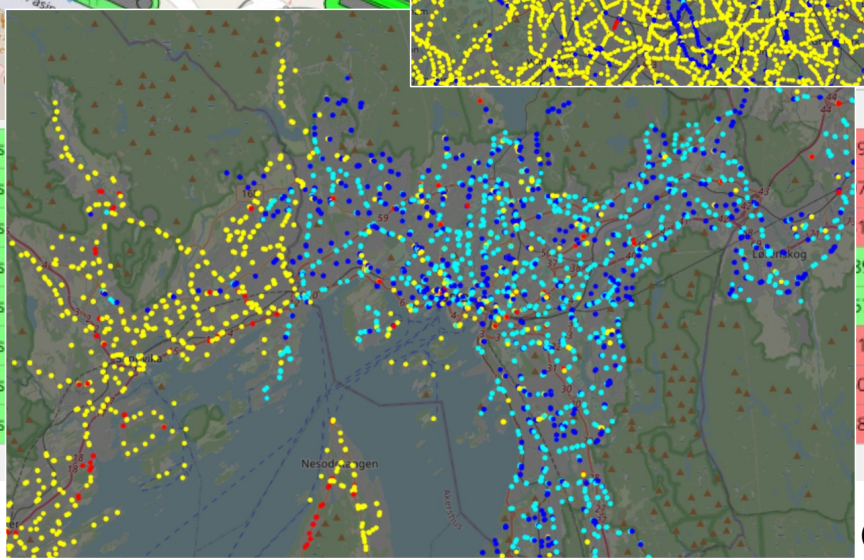


London



	bus	ley bus	are taxi	tram	gn train
platform					
stop position 1					
stop position 2					
stop position 3					
stop position 4					
stop position 5					
stop position 6					
stop position 7					
stop position 8					



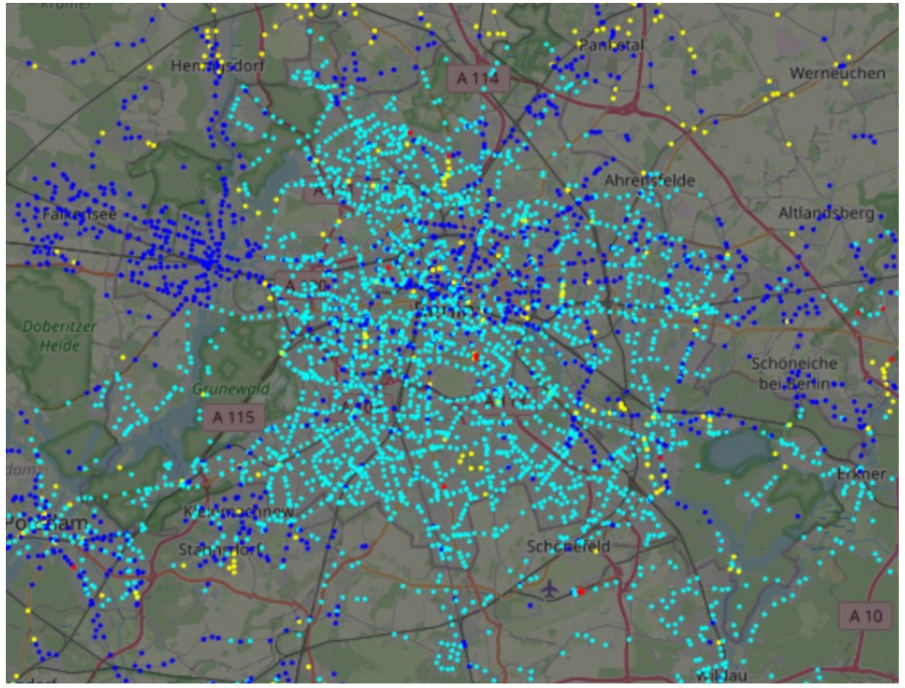


931740186949
756617313717
160668327285
39950435116
57320744631
111604292086
02045940943
812036931915

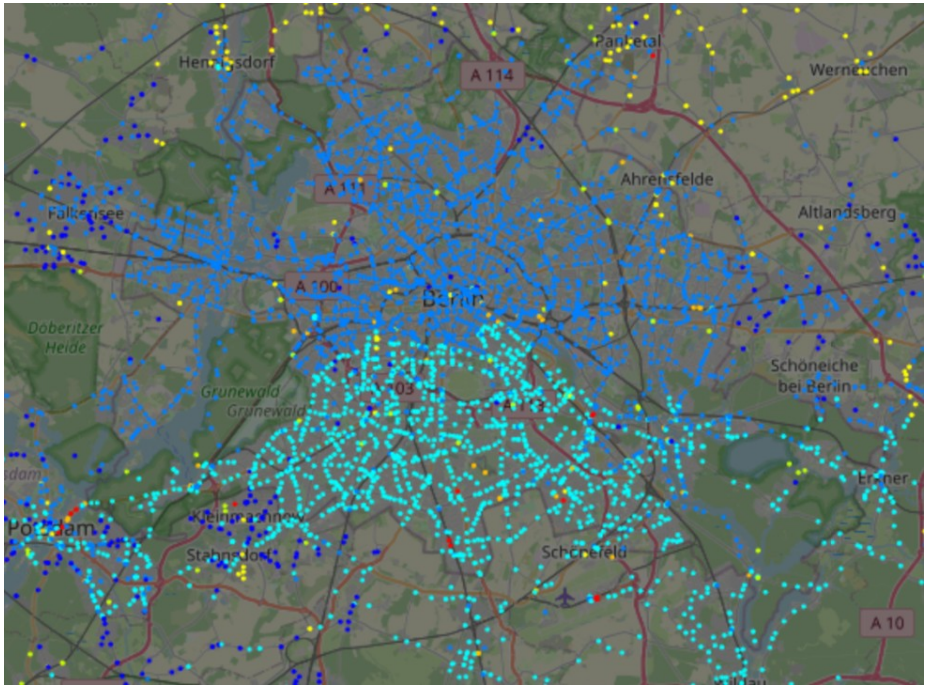
Paris

Oslo

Berlin public transport “wall”?



Berlin in early 2025



Berlin in early 2026

“wall” is still moving south…

Distances on the sphere (or on the WSG84 ellipsoid)

Have to compute **lots of short distances** for grouping OSM objects!

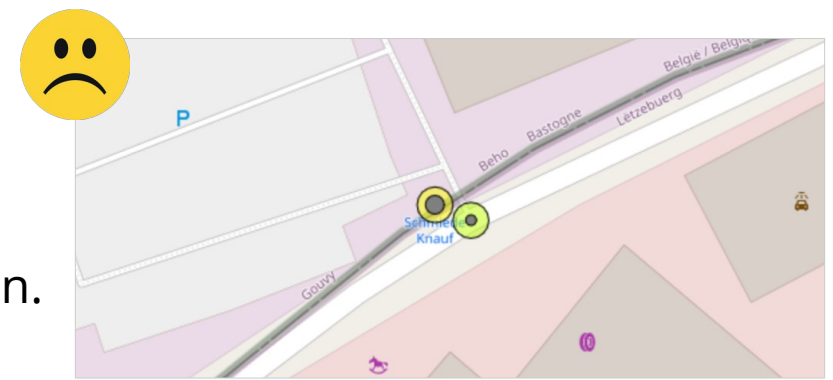
Geodetic distances about 15 times more expensive than Euclidean distances due to trigonometric function evaluations.

AND: most geodata libraries (e.g., GeoPandas) only support Euclidean distances on a flat map.

Have to **project the planet to 2d maps** without too much distortion!

Idea:

1. Cut the planet along admin boundaries (to not cut through multi-object PT stops).
2. Project each region with minimal distortion.
3. Choose region size to guarantee < 1 % mearsurement error (Euclidean vs. WSG84 geodetic).



Cut the planet

OSM has admin boundaries

but: no tool to cut along such boundaries till all regions are small enough

Build it yourself... (www2.htw-dresden.de/~fjeme691/flemming/codedata/div4aep.html)

Resulting data set “administrative units/azimuthal equidistant projections datasets” available online (www2.htw-dresden.de/~fjeme691/flemming/codedata/au_aep_datasets.html)

measurement error: 1 %
1220 regions
max. diameter: 2200 km

for 6 regions diameter is above 2200 km
(no smaller admin units in OSM)

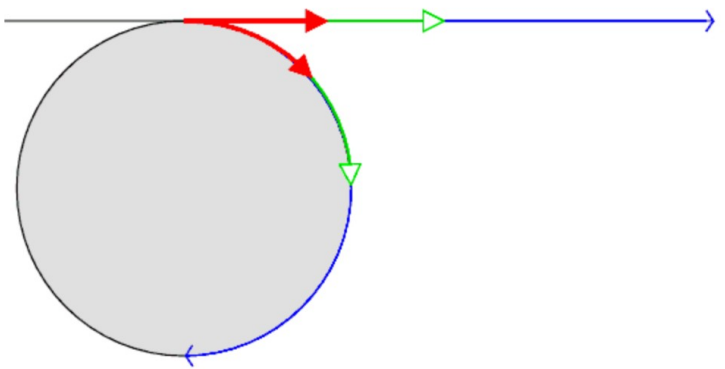
Optimal map projections

Want to use equidistant azimuthal projection

Only one parameter: **the map's center**

Measurement error grows with distance to center

Choose center such that maximum distance to region's boundary is minimal



Wikimedia Commons, Lars H. Rohwedder

Smallest enclosing circle!

Someone should have solved this already…

…but only in the plane, not on the sphere

Welzl’s algorithm

smallest enclosing circles in the plane: Welzl’s algorithm (Welzl 1991)

```
algorithm welzl is[8]
    input: Finite sets P and R of points in the plane |R| ≤ 3.
    output: Minimal disk enclosing P with R on the boundary.

    if P is empty or |R| = 3 then
        return trivial(R)
    choose p in P (randomly and uniformly)
    D := welzl(P − {p}, R)
    if p is in D then
        return D

    return welzl(P − {p}, R ∪ {p})
```

en.wikipedia.org, Smallest-circle problem, 25.05.2026

lineare runtime (in terms of points to enclose)

but: no trivial generalization to the sphere :-(

Welzl one the (hemi-)sphere

About 100 000 points to enclose (e.g., for France).

Slightly modify a 3d variant of Welzl and you get **linear runtime** on the sphere if all points are contained in a hemi-sphere.

But we do not have to know the hemi-sphere in advance!

(Flemming 2024, arXiv:2407.19840 [cs.CG])

Python package available (https://pypi.org/project/secots)

Full sphere in linear time up to now unsolved (e.g., Dinis, Mamede 2010)

The end

Thank you!

jens.flemming@htw-dresden.de

https://www2.htw-dresden.de/~fjeme691/flemming

https://ptsa.io.informatik.htw-dresden.de