Appendix 2c – Data preparation: Green space entry detection

Why green space entry detection?

* To measure the distance from a building entry to the nearest green space, we need a fixed point that demarcates the nearest entry point to that green space.
* Unfortunately, the data that we have acquired does not provide reliable information on those entry points.
* Consequently, we have to estimate where the entry points of each park in a city might be located on a network.

Green space entry detection

* To detect green space entry points as reliable as possible, we chose to intersect the urban atlas (UA) polygons of the classes urban green (14100) and forest (31000) with the OpenStreetMap (OSM) network data.
* We used the outline of the UA polygons so the intersection algorithm only yields points that are on the edge between green space and city.
* To refine this approach, we tested the outlines of green space polygons with buffers of different sizes applied to them.
* We selected three Berlin green spaces (Tempelhofer Feld, Viktoriapark and Treptower Park) to validate the accuracy of the different buffer sizes.
* During the process of green space entry detection, we apply the different buffer sizes according to their accuracy until each green space in a city has at least one entry point.
* We developed the *greenSpacePrep* function in R to take on this task.

Functionality

Data used:

* OSM street network
* UA green space polygons (classes 14100, 31000)

Method:

* Intersect OSM street network with UA polygons of classes 14100, 31000

1. Extract all parks / forests from all UA cities to avoid missing parks at shared edges of polynuclear cities
   1. Since people might travel across city boundaries to reach green spaces, we included respective UA land-cover polygons from neighboring cities.
   2. (Add graphic?)
2. Cast green space from MULTIPOLYGONS / POLYGONs to LINESTRINGs and intersect with OSM street network
   1. Assumption: intersection = green space entry

* To ensure all parks have an entry point:
  + Start at negative buffer of 5 (see sensitivity analysis for explanation)
* For green spaces where outline was not intersected with network:
  + Increase buffer by 5 m, repeat procedure
  + Higher increase with larger buffer size to reduce computation time:
    - < 50 m: 5 m increase
    - 50 – 100 m: 10 m increase
    - 100 – 200 m: 25 m increase
    - > 200 m: 50m increase

Validation of method:

* Validation data:
  + Three Parks in Berlin:
    - Viktoriapark
      * Park entries well defined by surrounding walls / greenery
    - Treptower Park
      * Rather open / lots of meadows at parks edges
      * Delimited by the river Spree on one side
    - Tempelhofer Feld
      * in original UA data set classified as sports and leisure facilities
      * fence around entire area
      * entry only possible through gates
  + Marked GPS coordinates of each entry point as validation points
    - Where are people expected to enter the Parks?
      * Paths leading into the parks
      * Gates
      * Other (bridges, station exits…)
    - For sensitivity analysis:
      * Google Earth Pro aerial photographs to mark the paths serving as entrances
* Sensitivity analysis of buffer sizes
  + Entry point layers with buffer sizes -10, -5, 0, 5, 10 using above mentioned method
  + Checked for each validation point whether or not it was detected with the utilized buffer size
  + Accuracy assessment for classes
    - Accuracy = Correctly detecting green space entries (validation data) without producing pseudo entries (at intersections with street network that are no entry points)
    - Most accurate buffer size: negative 5 m
      * (Only validated in Berlin!)
      * Probably because of park polygons aligning with next biggest streets
    - Second most accurate: 0 m
    - Decreasing accuracy with higher buffer size