

TEILAUTOMATISIERTE ERSTELLUNG VON (BAU-)BLOCKSEITEN AUF GRUNDLAGE VON ALKIS

KORIS FORUM 2024

6. Dezember 2024

Jan Lunge

Stadt Wolfsburg

Abteilung Statistik und Stadtforschung



AGENDA

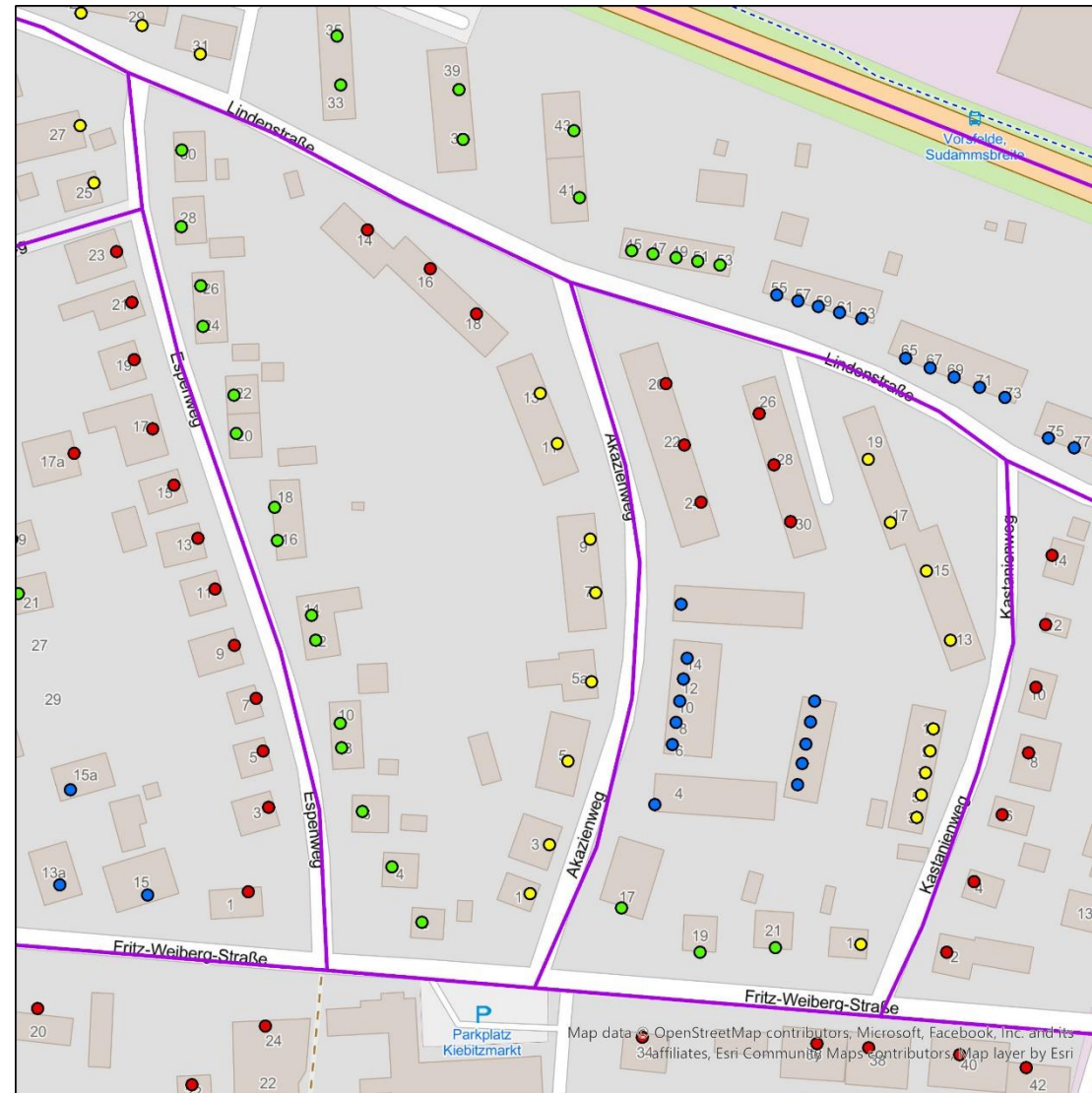
1. Ausgangslage
2. Lösungsansatz
3. Umsetzung
4. Weitere Anwendungsfälle



Ausgangslage

STATUS QUO

- Es gibt eine kleinräumige Gliederung
- Nicht für alle Gliederungsebenen liegen räumliche Repräsentationen vor
 - Ortschaftsbereiche
 - Stadt-/Ortsteile
 - Stat. Bezirke
 - Baublöcke
 - (Bau)Blockseiten

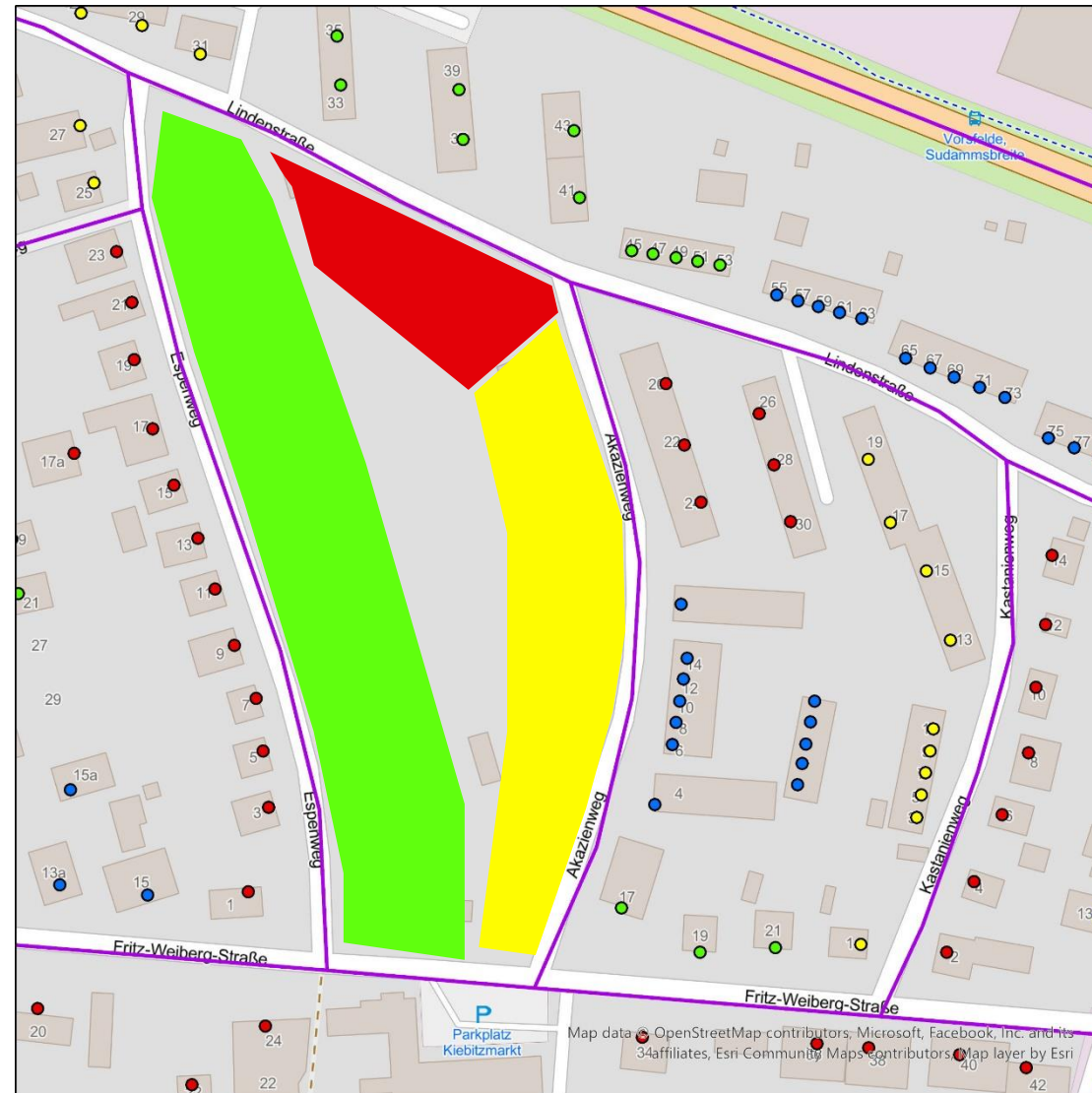


AUFGABENSTELLUNG

- Überführung der punktförmigen Blockseiteninformationen in eine flächenhafte Darstellung
- Anzahl Baublöcke in Wolfsburg: 1.996
- Anzahl Blockseiten in Wolfsburg: 5.843



(Teil-)Automatisierte Lösung





Lösungsansatz



KONZEPTIONELLE GEDANKEN

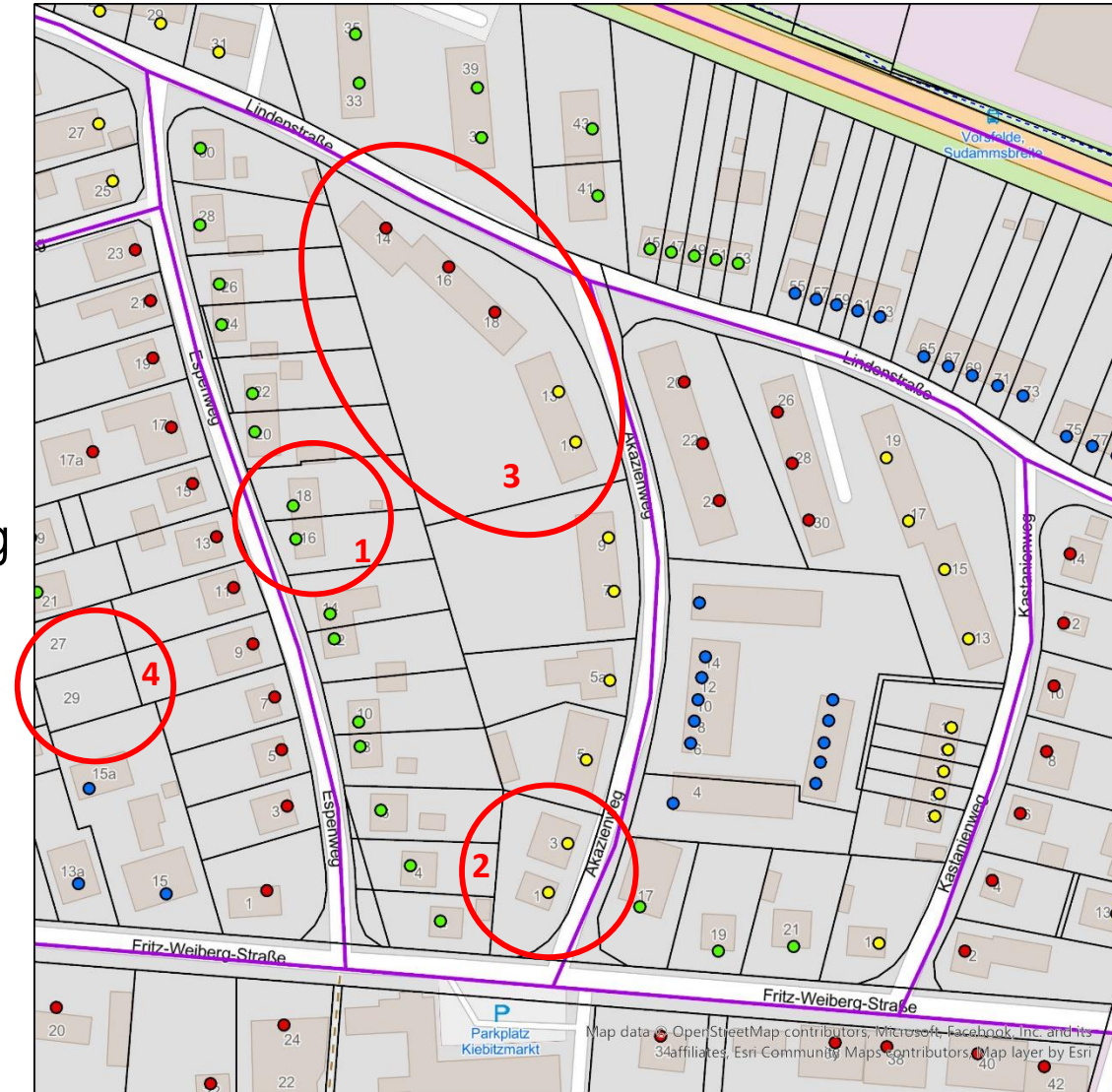
- Welchen logischen und topologischen Regeln unterliegen unsere Blockseiten?
 - Eine Blockseite kann nur eine dazugehörige Straße haben
 - Blockseiten dürfen sich nicht überlappen
 - Blockseiten müssen innerhalb eines Baublocks liegen

- Welche Datensätze stehen zur Verfügung auf denen man Blockseiten aufbauen kann?
 - Baublöcke
 - Flurstücke



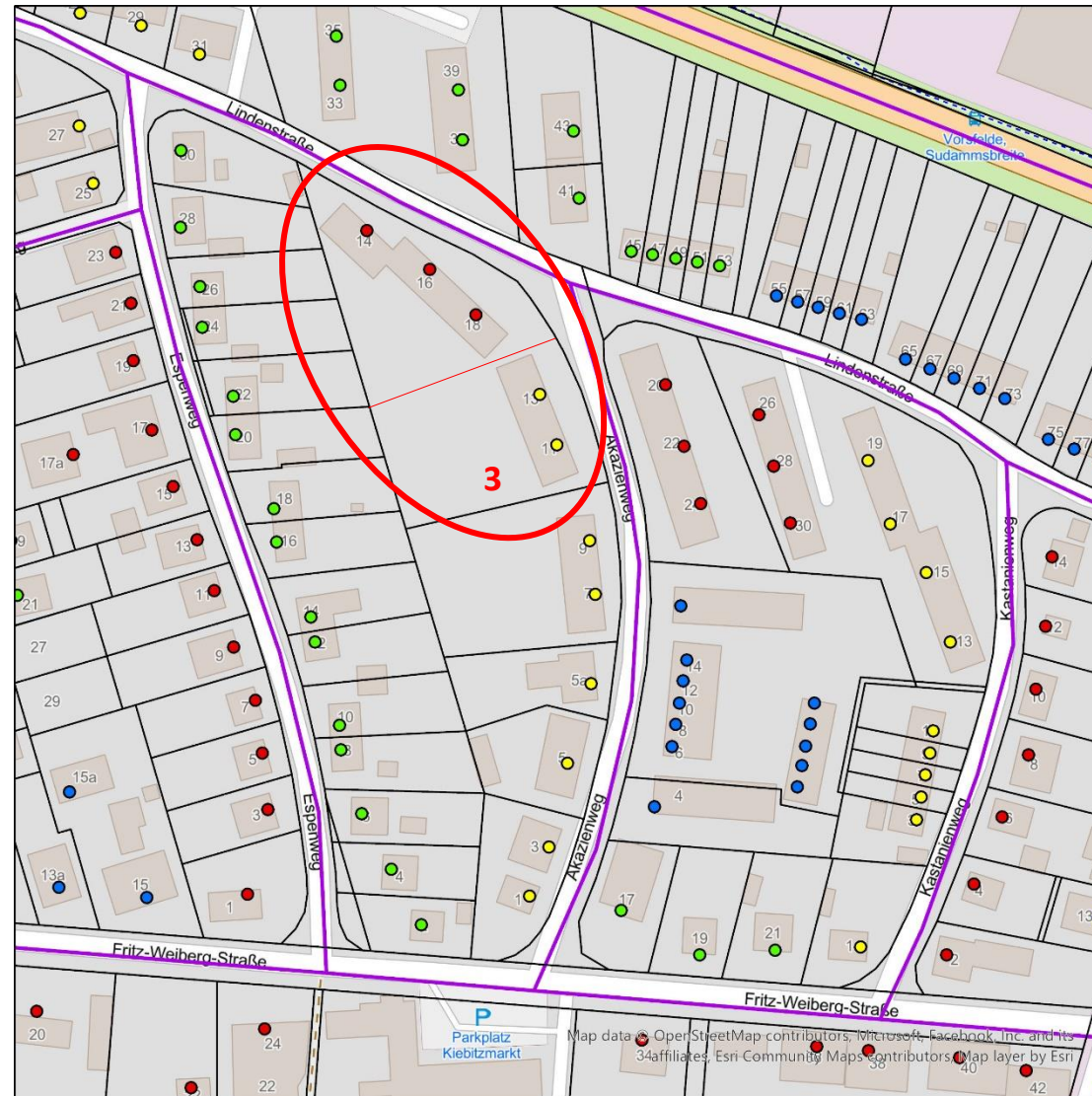
LAGEBEZIEHUNGEN

- Topologische Beziehungen zwischen den Flurstücken und Adressen mit kleinräumiger Gliederung:
 - 1) Es gibt eine 1:1 Beziehung
 - 2) Es gibt eine 1:n Beziehung mit gleicher kleinräumigen Gliederung
 - 3) Es gibt eine 1:n Beziehung mit verschiedener kleinräumigen Gliederung
 - 4) Es gibt keine Lagebeziehung



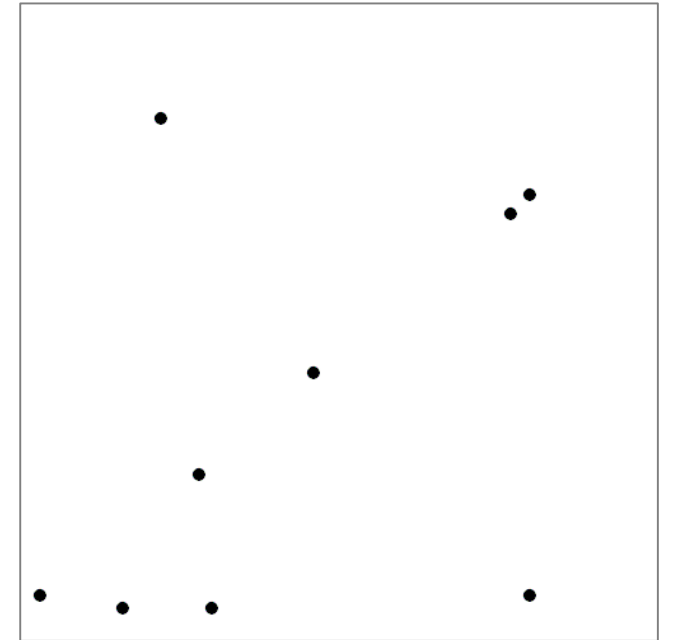
LAGEBEZIEHUNGEN

- Topologische Beziehungen zwischen den Flurstücken und Adressen mit kleinräumiger Gliederung:
 - 1) Es gibt eine 1:1 Beziehung
 - 2) Es gibt eine 1:n Beziehung mit gleicher kleinräumigen Gliederung
 - 3) Es gibt eine 1:n Beziehung mit verschiedener kleinräumigen Gliederung**
 - 4) Es gibt keine Lagebeziehung



EXKURS: VORONOI-DIAGRAMME

- auch Thiessen-Polygone
- Bereits 1644 durch Descartes beschrieben, später durch Dirichlet und Voronoi erstmals mathematisch analysiert
- Zerlegung eines Raums in „Einflussgebiete“
- Anwendung in diversen wissenschaftlichen Fachrichtungen (Mathematik, Physik, Meteorologie, Kristallografie, Anatomie, ...)

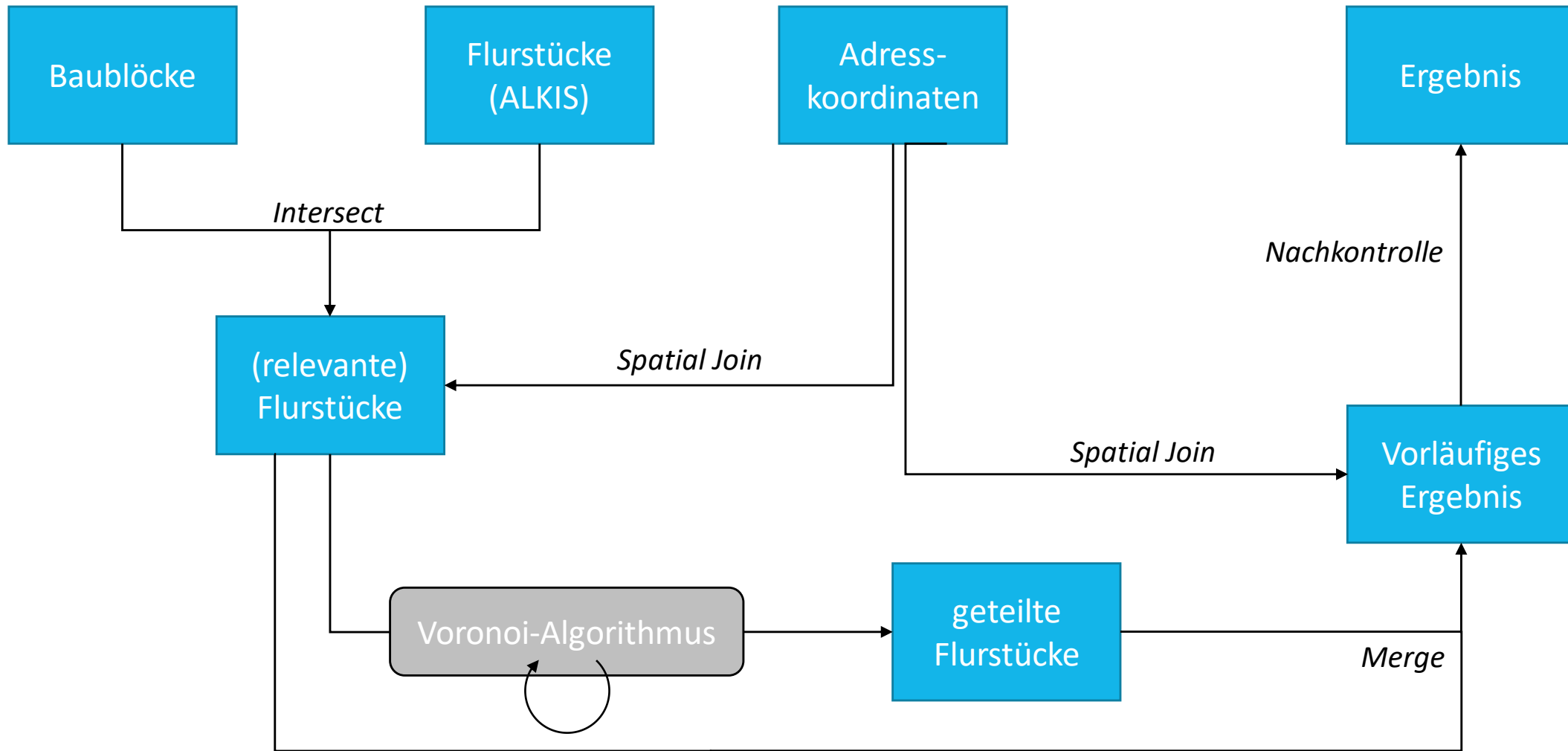


https://de.wikipedia.org/wiki/Datei:Voronoi_growth_euclidean.gif



Umsetzung

TEILUNG VON FLURSTÜCKEN



TECHNISCHE UMSETZUNG 1.0

- Zunächst Umsetzung mit Python und ArcGIS
- Problem: `arcpy.analysis.CreateThiessenPolygons()` ist nicht in der Basic-Lizenz enthalten
- Lösung: Python-Package SciPy -> `scipy.spatial.Voronoi()`



- Workflow:
 - Fläche puffern
 - Koordinaten der Stützpunkte aus Flurstück auslesen
 - Koordinaten der Adresspunkte anfügen
 - Voronoi-Regionen berechnen
 - Regionen als Polygon in neuen Datensatz einfügen
 - Flächen auf ursprüngliche Größe zuschneiden

TECHNISCHE UMSETZUNG 1.0

```
1 def Voronoi(Points, Area = ""):
2     from scipy.spatial import Voronoi as Voronoi_scipy
3     arcpy.env.overwriteOutput = True
4     arcpy.Buffer_analysis (Area, 'Voronoi_Puffer', 300)
5
6     if 'Voronoi_intermediate' in arcpy.ListFeatureClasses(): arcpy.Delete_management('Voronoi_intermediate')
7     arcpy.CreateFeatureclass_management(out_path= arcpy.env.workspace, out_name='Voronoi_intermediate', geomet
8
9     point_coords = []
10    with arcpy.da.SearchCursor(Points, ["SHAPE@XY"]) as cursor:
11        for row in cursor:
12            point_coords.append([row[0][0],row[0][1]])
13    del cursor
14
15    with arcpy.da.SearchCursor('Voronoi_Puffer', ["SHAPE@"]) as cursor:
16        vertices = cursor.next()[0]
17        for P in vertices.getPart(0):
18            #print P.X, P.Y
19            point_coords.append([P.X, P.Y])
20            #print "appended"
21    del cursor
22    arcpy.Delete_management('Voronoi_Puffer')
23    #print(point_coords)
24    #sys.exit()
25
26    V_scipy = Voronoi_scipy(point_coords) #Berechnung der Voronoi-Polygone mit zentriertem Koordinatensystem
```



TECHNISCHE UMSETZUNG 1.0

- Vorteile:
 - Viel Funktionalität ist bereits vorhanden und muss nur verkettet werden
 - Integration in GIS
 - Aufbau auf vorhandenem Know-how
- Probleme:
 - ArcGIS ist eine proprietäre Software
 - Workflow kann nicht vollständig mit ArcGIS abgebildet werden
 - Workflow ist inperformant

TECHNISCHE UMSETZUNG 2.0

- Streichen von ArcGIS Pro aus der Verarbeitung
- Was muss ersetzt werden und was kommt als Ersatz in Frage?
 - Spatial Join
 - Auslesen von Koordinaten aus Geometrien
 - Schreiben von Geometrien
 - (Geodatenformat)
- Vollständige Umstellung auf open-source Lösungen:
 - Python (z.B. geopandas, shapely, ...)
 - PostGIS
- Refactoring des Workflow




```

29 def get_coords(geom):
30     """
31     ...Funktion zum Auslesen der Koordinaten der übergebenen Geometrie. Bei Polygonen wird eine
32     ...eine Liste der Koordinaten aller Stützpunkte übergeben.
33
34     ...Parameters
35     -----
36     ...geom: shapely.geometry-Objekt
37     ...Geometrieobjekt (Point, Polygon oder Multipolygon) dessen Koordinaten ausgelesen werden sollen
38
39     ...Returns
40     -----
41     ...coords: list
42     ...Liste mit den Koordinaten der übergebenen Geometrie.
43     """
44
45     if geom.geom_type == "Point":
46         coords = [geom.x, geom.y]
47
48     elif geom.geom_type == "Polygon":
49         # Auslesen der Koordinaten aller Stützpunkte und abspeichern als Tuple
50         xx, yy = geom.exterior.coords.xy
51         xx = xx.tolist()
52         yy = yy.tolist()
53         coords = list(zip(xx, yy))
54
55     elif geom.geom_type == "MultiPolygon":
56         if len(geom.geoms) == 1:
57             # Auslesen der Koordinaten aller Stützpunkte der ersten Geometrie
58             # und abspeichern als Tuple
59             xx, yy = geom.geoms[0].exterior.coords.xy
60             xx = xx.tolist()
61             yy = yy.tolist()
62             coords = list(zip(xx, yy))
63
64         elif len(geom.geoms) > 1:
65             print("Achtung: Gebäude MultiPolygon besteht aus mehr als einem Polygon.")
66         elif len(geom.geoms) < 1:
67             print("Achtung: Gebäude MultiPolygon ist leer")
68     else:

```



T

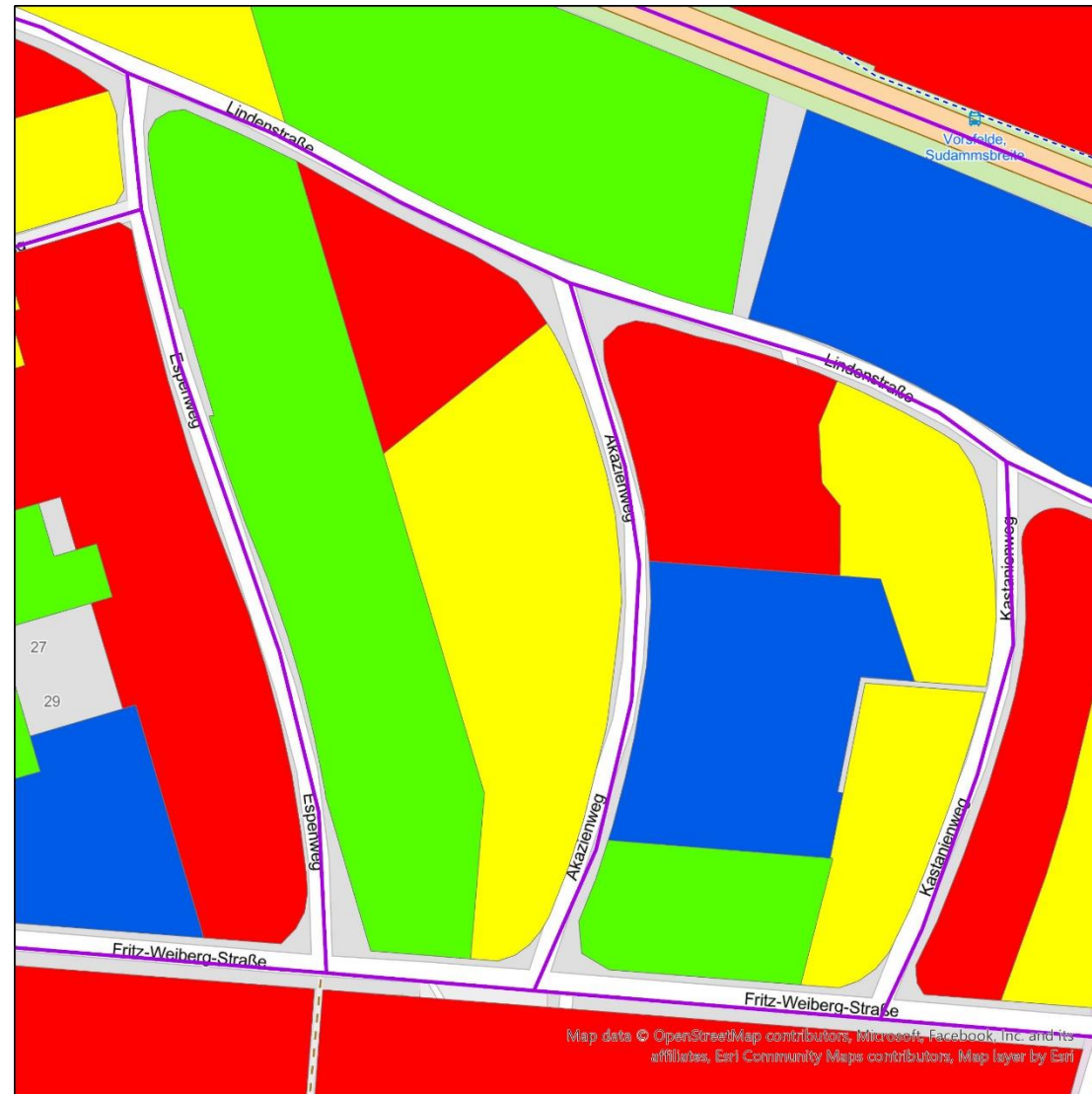
```
74 def voronoi(Points, Area, buffer_distance = 50):
75     """
76     ...Funktion zur Aufteilung von Polygonen mit mehreren Adresspunkten nach geometrischer Nähe (Voronoi-
77
78     ...Parameters
79     -----
80     ...Points: geopandas.GeoSeries oder list of shapely.geometry.Points-Objekten
81     ...Die Punkte innerhalb der Fläche anhand derer die Fläche aufgeteilt werden soll (Punkte).
82     ...Area: shapely.geometry-Objekt
83     ...Die mittels Voronoi zu zerteilende Fläche (Polygon oder MultiPolygon).
84     ...buffer_distance: int, optional
85     ...Default ist 50. Die Pufferdistanz in Metern für das Puffern der Flächengeometrien. Der Wert
86     ...hängt von der Größe der aufzuteilenden Flächen ab. Bei Gebäuden ist 50 okay, bei Flurstücken
87     ...kann man auf 100 gehen.
88
89     ...Returns
90     -----
91     ...v_geoms_clipped: list
92     ...Liste mit allen Voronoi-Polygonen als shapely-geometry-Polygon-Objekten.
93     """
94     # Puffern des zu zerschneidenden Polygons, damit alle Randbereiche korrekt mitgenommen werden
95     # 100 Meter ggf. schon zu großzügig für Gebäude, vielleicht 70 oder 50 Meter ausreichend.
96     Area_buffered = Area.buffer(buffer_distance)
97
98     # Auslesen der Koordinaten der Adresspunkte, die im Polygon liegen
99     point_coords = []
00     for point in Points:
01         point_coords.append(get_coords(point))
02
03     # Auslesen der Koordinaten aller Polygon-Stützpunkte und anfügen an die Koordinatenliste
04     for vertex in get_coords(Area_buffered):
05         point_coords.append([vertex[0], vertex[1]])
06
07     # Berechnung der Voronoi-Polygone mit zentriertem Koordinatensystem
08     V_scipy = _Voronoi_scipy(point_coords)
09
```

6.



ERGEBNIS

- Alle relevanten (Teil-)Flurstücke werden zusammengeführt
- Infos zur kleinräumigen Gliederung werden per Spatial Join hinzugefügt
- Manuelle Nachkontrolle und ggf. Anpassungen nötig

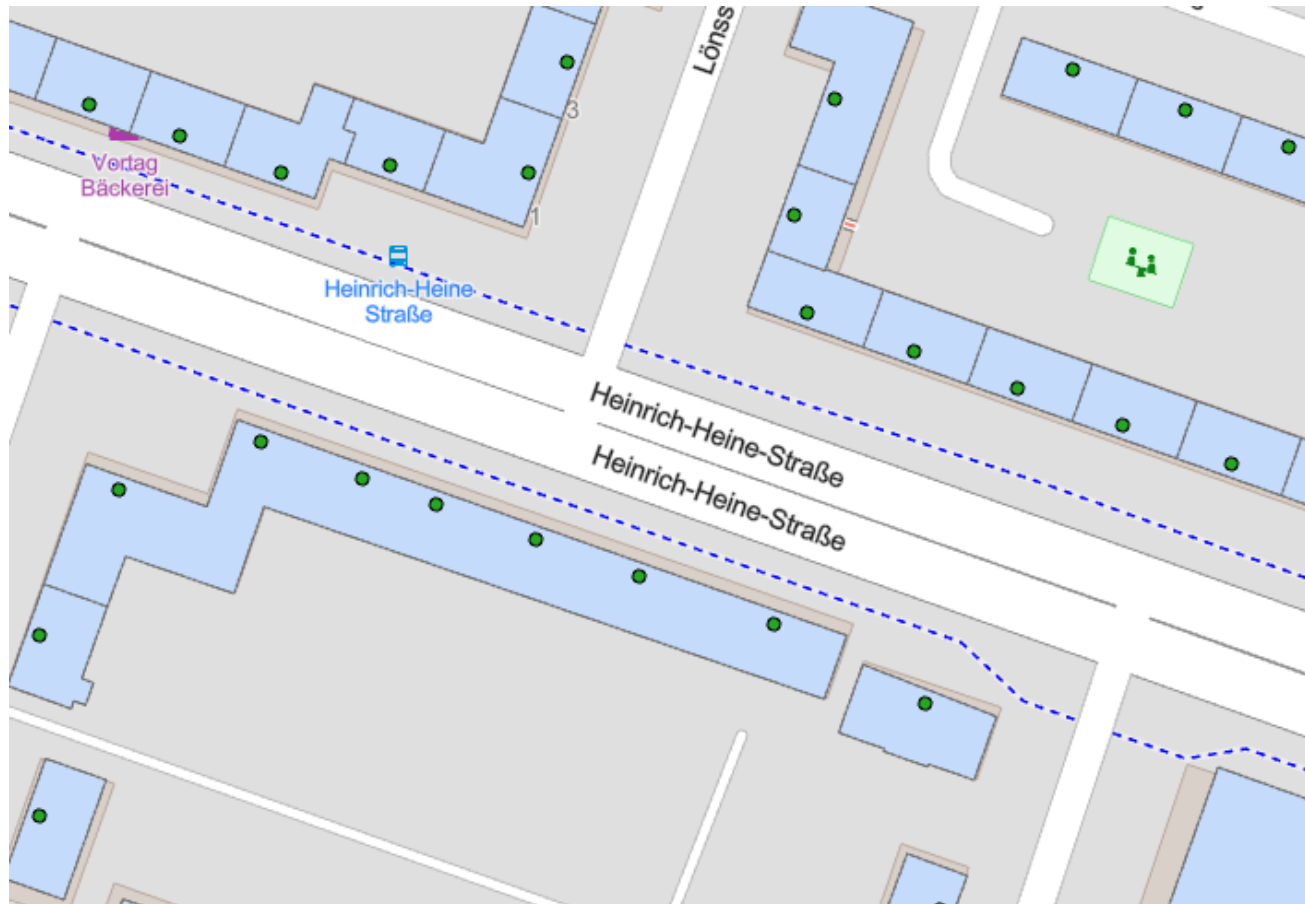




Weitere Anwendungsfälle

WEITERE ANWENDUNGSFÄLLE

- Herstellung einer Gebäudebasis für die Kommunale Wärmeplanung



VIELEN DANK FÜR IHRE AUFMERKSAMKEIT!

Jan Lunge
Stadt Wolfsburg
Wissenschaftlicher Mitarbeiter
jan.lunge@stadt.wolfsburg.de
+49 5361 28-1499

