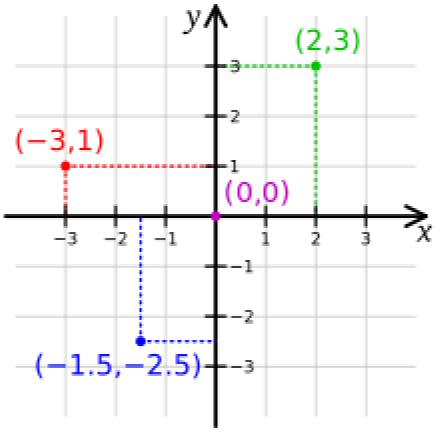
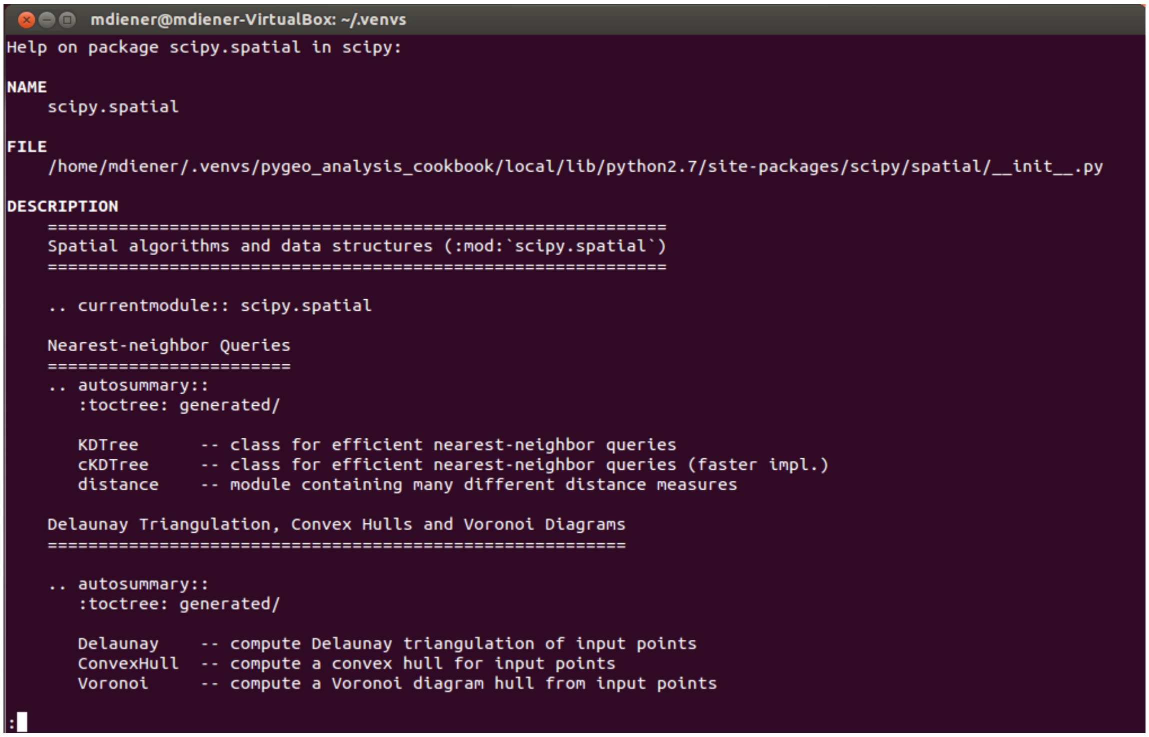
1
Setting Up Your Geospatial Python Environment

Installing shapely, matplotlib, and descartes





```
Help on package scipy.spatial in scipy:

NAME
    scipy.spatial

FILE
    /home/mdiener/.venvs/pygeo_analysis_cookbook/local/lib/python2.7/site-packages/scipy/spatial/__init__.py

DESCRIPTION
    =============================================================
    Spatial algorithms and data structures (:mod:`scipy.spatial`)
    =============================================================

    .. currentmodule:: scipy.spatial

    Nearest-neighbor Queries
    ========================
    .. autosummary::
       :toctree: generated/

       KDTree      -- class for efficient nearest-neighbor queries
       cKDTree     -- class for efficient nearest-neighbor queries (faster impl.)
       distance    -- module containing many different distance measures

    Delaunay Triangulation, Convex Hulls and Voronoi Diagrams
    =========================================================

    .. autosummary::
       :toctree: generated/

       Delaunay    -- compute Delaunay triangulation of input points
       ConvexHull  -- compute a convex hull for input points
       Voronoi     -- compute a Voronoi diagram hull from input points

:
```

2

Working with Projections

Introduction

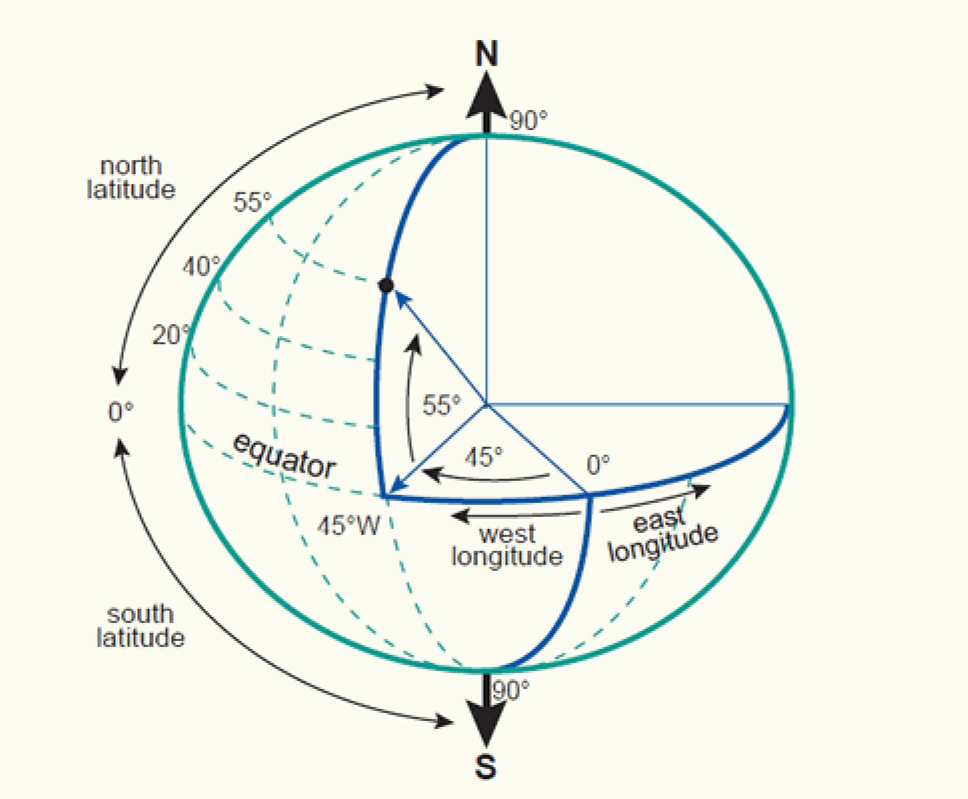


Illustration 1: Geographic Coordinate System (http://kartoweb.itc.nl/geometrics/coordinate%20systems/coordsys.html)

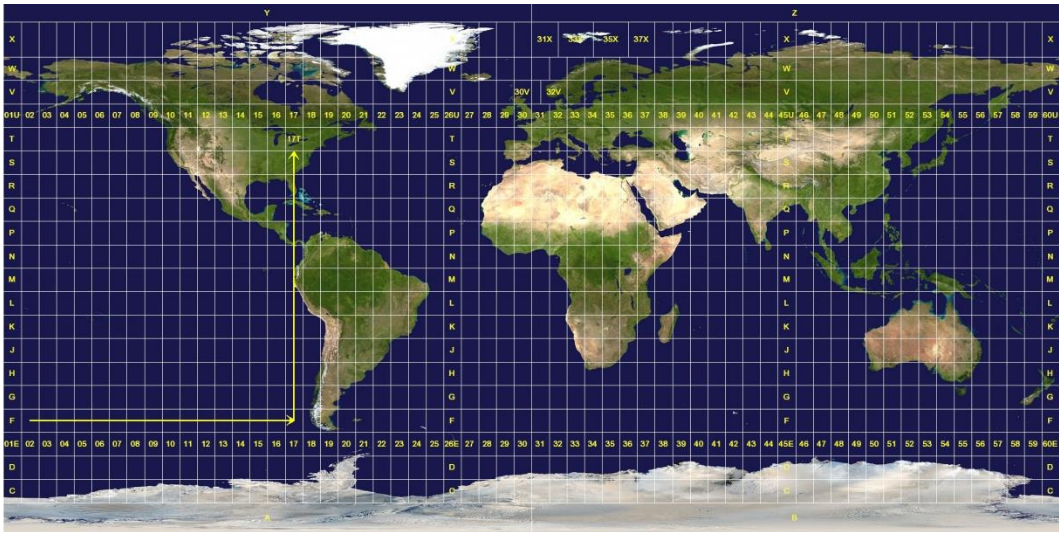
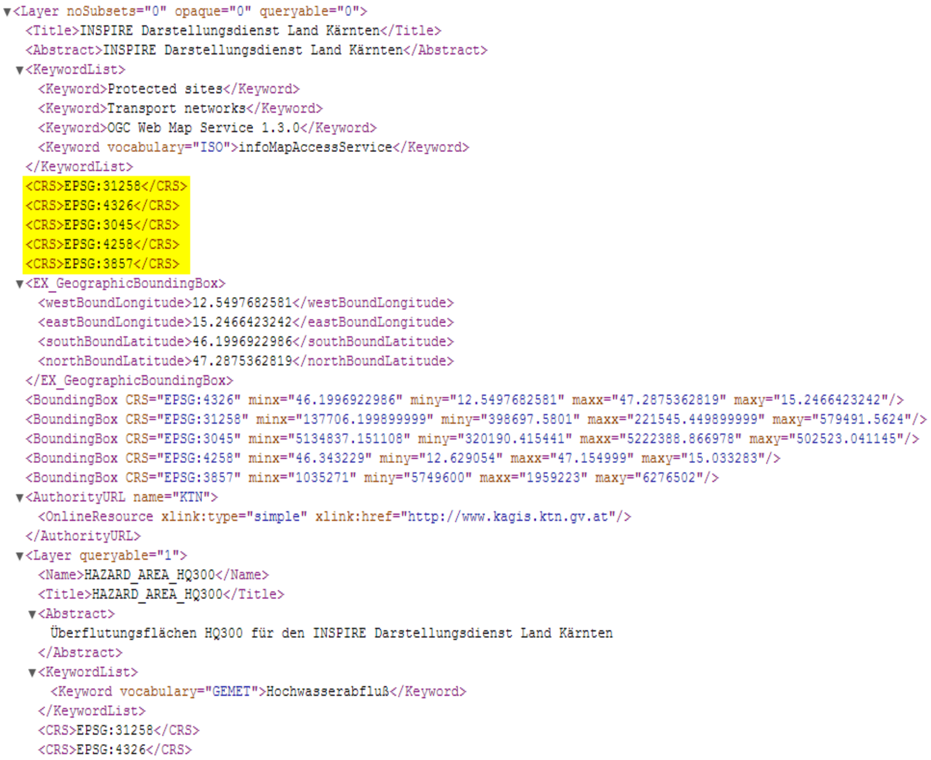

Y
Z
X
W
V
31X
33X
35X
37X
30V
32V
01U 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26U 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45U 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60U
17T
T
S
R
Q
P
N
M
L
K
J
H
G
F
01E 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26E 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45E 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60E
D
C
A
B

Illustration 2: Projected Coordinate System UTM
(http://en.wikipedia.org/wiki/Universal_Transverse_Mercator_coordinate_system#mediaviewer/File:Utm-zones.jpg)

Listing the projection(s) from a WMS server

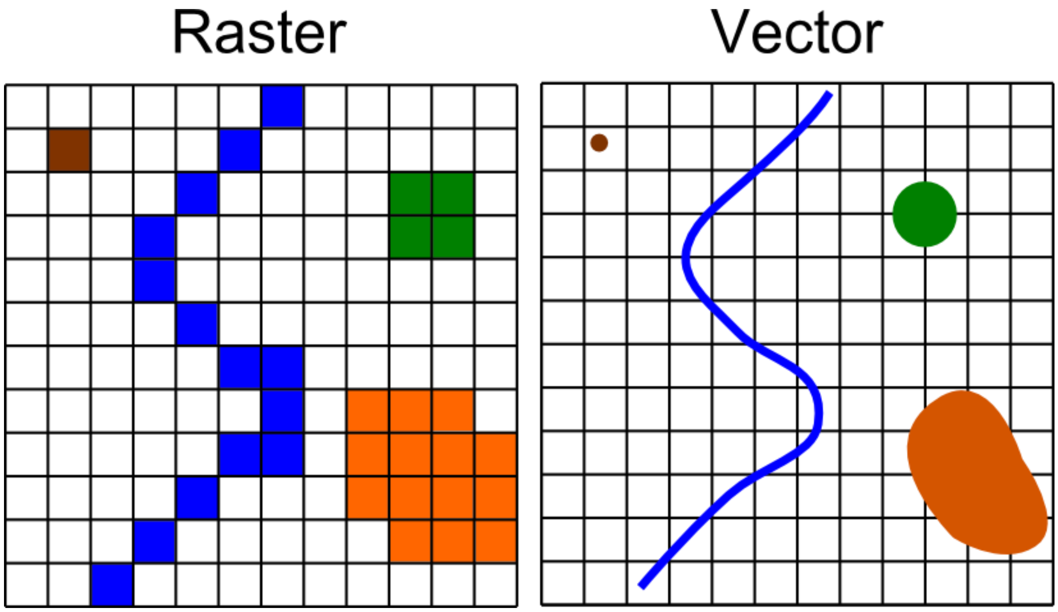


```
▼<Layer noSubsets="0" opaque="0" queryable="0">
   <Title>INSPIRE Darstellungsdienst Land Kärnten</Title>
   <Abstract>INSPIRE Darstellungsdienst Land Kärnten</Abstract>
 ▼<KeywordList>
     <Keyword>Protected sites</Keyword>
     <Keyword>Transport networks</Keyword>
     <Keyword>OGC Web Map Service 1.3.0</Keyword>
     <Keyword vocabulary="ISO">infoMapAccessService</Keyword>
   </KeywordList>
   <CRS>EPSG:31258</CRS>
   <CRS>EPSG:4326</CRS>
   <CRS>EPSG:3045</CRS>
   <CRS>EPSG:4258</CRS>
   <CRS>EPSG:3857</CRS>
 ▼<EX_GeographicBoundingBox>
     <westBoundLongitude>12.5497682581</westBoundLongitude>
     <eastBoundLongitude>15.2466423242</eastBoundLongitude>
     <southBoundLatitude>46.1996922986</southBoundLatitude>
     <northBoundLatitude>47.2875362819</northBoundLatitude>
   </EX_GeographicBoundingBox>
   <BoundingBox CRS="EPSG:4326" minx="46.1996922986" miny="12.5497682581" maxx="47.2875362819" maxy="15.2466423242"/>
   <BoundingBox CRS="EPSG:31258" minx="137706.199899999" miny="398697.5801" maxx="221545.449899999" maxy="579491.5624"/>
   <BoundingBox CRS="EPSG:3045" minx="5134837.151108" miny="320190.415441" maxx="5222388.866978" maxy="502523.041145"/>
   <BoundingBox CRS="EPSG:4258" minx="46.343229" miny="12.629054" maxx="47.154999" maxy="15.033283"/>
   <BoundingBox CRS="EPSG:3857" minx="1035271" miny="5749600" maxx="1959223" maxy="6276502"/>
 ▼<AuthorityURL name="KTN">
     <OnlineResource xlink:type="simple" xlink:href="http://www.kagis.ktn.gv.at"/>
   </AuthorityURL>
 ▼<Layer queryable="1">
     <Name>HAZARD_AREA_HQ300</Name>
     <Title>HAZARD_AREA_HQ300</Title>
   ▼<Abstract>
       Überflutungsflächen HQ300 für den INSPIRE Darstellungsdienst Land Kärnten
     </Abstract>
   ▼<KeywordList>
       <Keyword vocabulary="GEMET">Hochwasserabfluß</Keyword>
     </KeywordList>
     <CRS>EPSG:31258</CRS>
     <CRS>EPSG:4326</CRS>
```

3

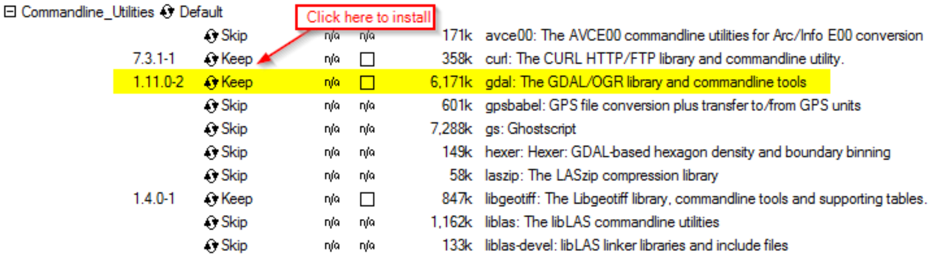
Moving Spatial Data from One Format to Another

Introduction



A Michael Diener drawing

Converting a Shapefile to a PostGIS table using ogr2ogr

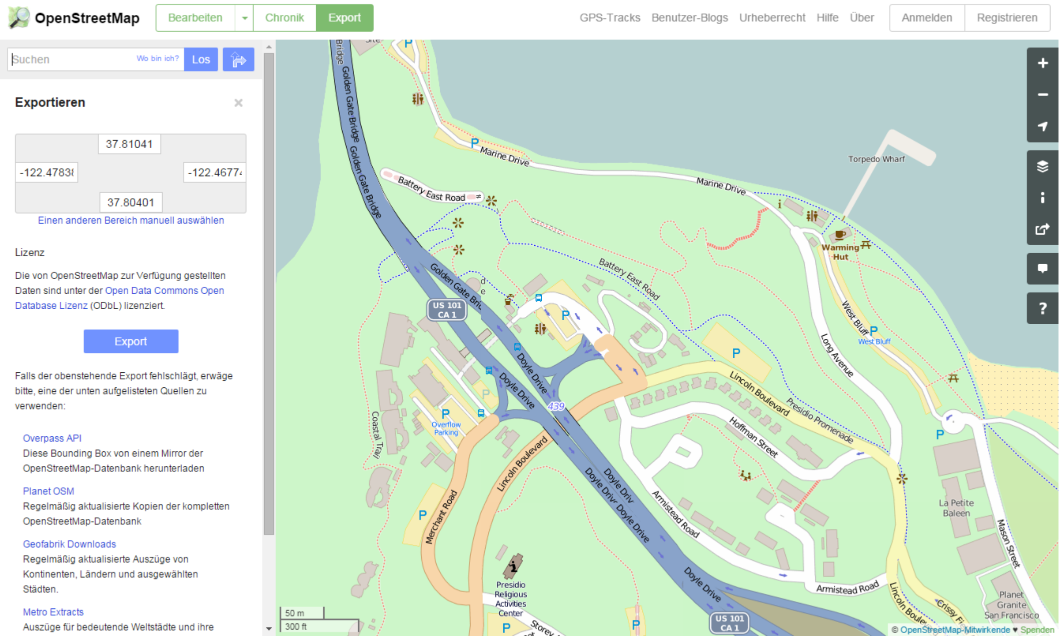


⊟ Commandline_Utilities ↻ Default

Click here to install

	↻ Skip	n/a	n/a	171k	avce00: The AVCE00 commandline utilities for Arc/Info E00 conversion
7.3.1-1	↻ Keep	n/a	□	358k	curl: The CURL HTTP/FTP library and commandline utility.
1.11.0-2	↻ Keep	n/a	□	6,171k	gdal: The GDAL/OGR library and commandline tools
	↻ Skip	n/a	n/a	601k	gpsbabel: GPS file conversion plus transfer to/from GPS units
	↻ Skip	n/a	n/a	7,288k	gs: Ghostscript
	↻ Skip	n/a	n/a	149k	hexer: Hexer: GDAL-based hexagon density and boundary binning
	↻ Skip	n/a	n/a	58k	laszip: The LASzip compression library
1.4.0-1	↻ Keep	n/a	□	847k	libgeotiff: The Libgeotiff library, commandline tools and supporting tables.
	↻ Skip	n/a	n/a	1,162k	liblas: The libLAS commandline utilities
	↻ Skip	n/a	n/a	133k	liblas-devel: libLAS linker libraries and include files

Converting an OpenStreetMap (OSM) XML to a Shapefile



Exportieren

37.81041

-122.4783 -122.4677

37.80401

Einen anderen Bereich manuell auswählen

Lizenz

Die von OpenStreetMap zur Verfügung gestellten Daten sind unter der Open Data Commons Open Database Lizenz (ODbL) lizenziert.

Export

Falls der obenstehende Export fehlschlägt, erwäge bitte, eine der unten aufgelisteten Quellen zu verwenden:

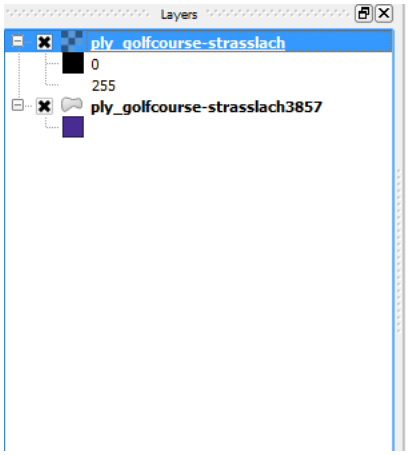
Overpass API
Diese Bounding Box von einem Mirror der OpenStreetMap-Datenbank herunterladen

Planet OSM
Regelmäßig aktualisierte Kopien der kompletten OpenStreetMap-Datenbank

Geofabrik Downloads
Regelmäßig aktualisierte Auszüge von Kontinenten, Ländern und ausgewählten Städten.

Metro Extracts
Auszüge für bedeutende Weltstädte und ihre

Converting a Shapefile (vector) to a GeoTiff (raster)



Layers

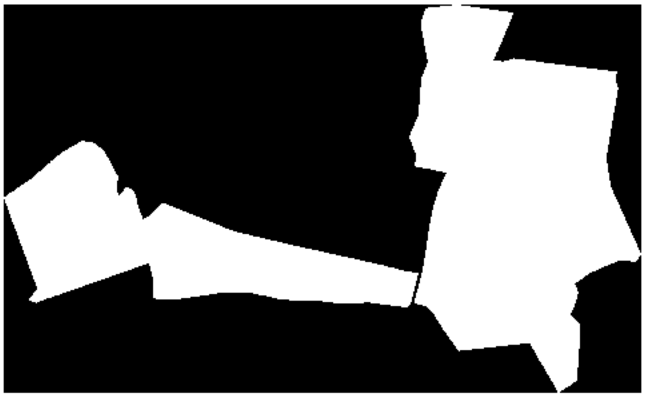
☒ ply_golfcourse-strasslach

■ 0

□ 255

☒ ply_golfcourse-strasslach3857

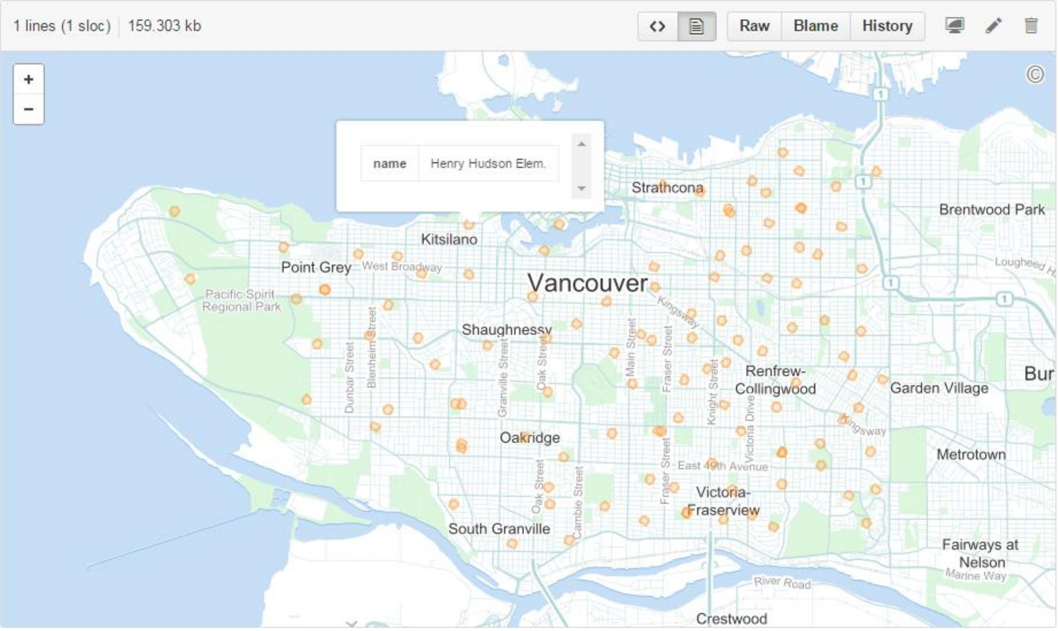
■



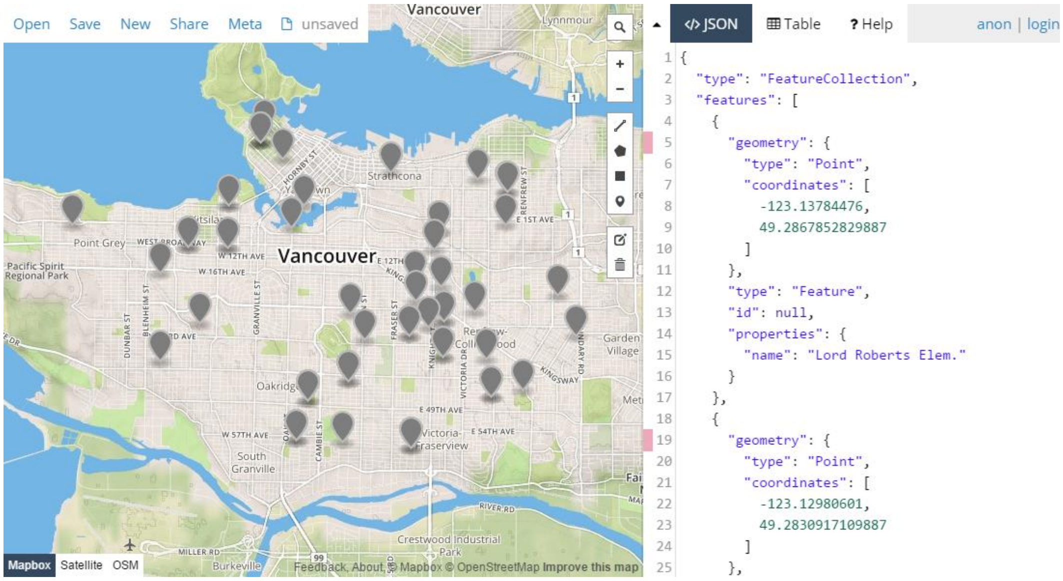
4

Working with PostGIS

Executing a PostGIS ST_Buffer analysis query and exporting it to GeoJSON

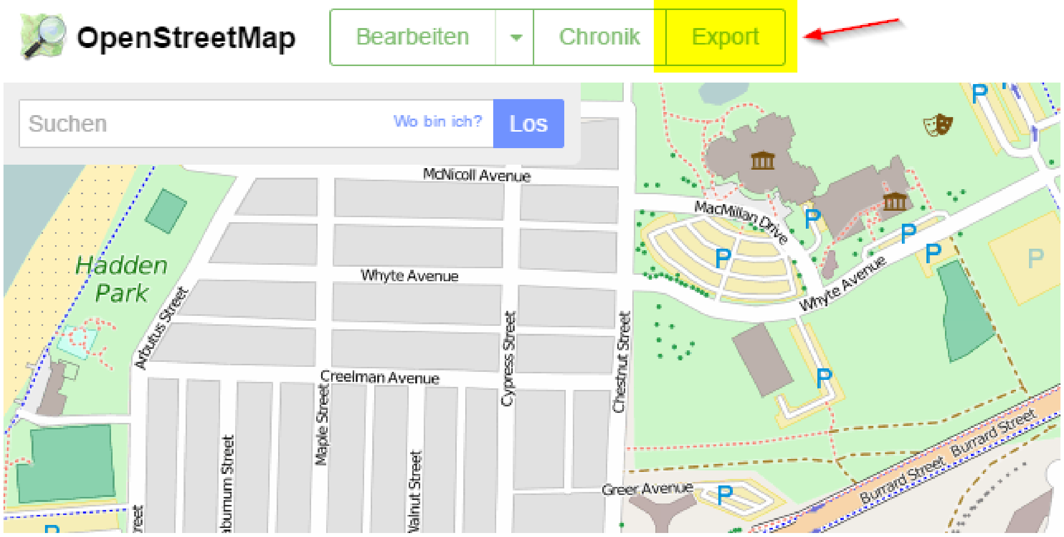


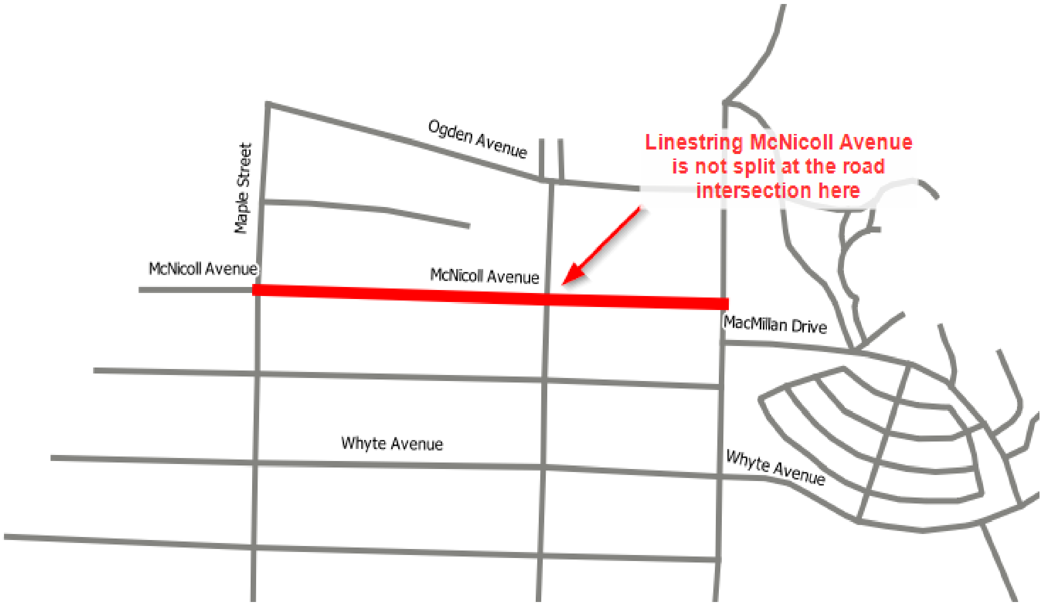
Finding out whether a point is inside a polygon.



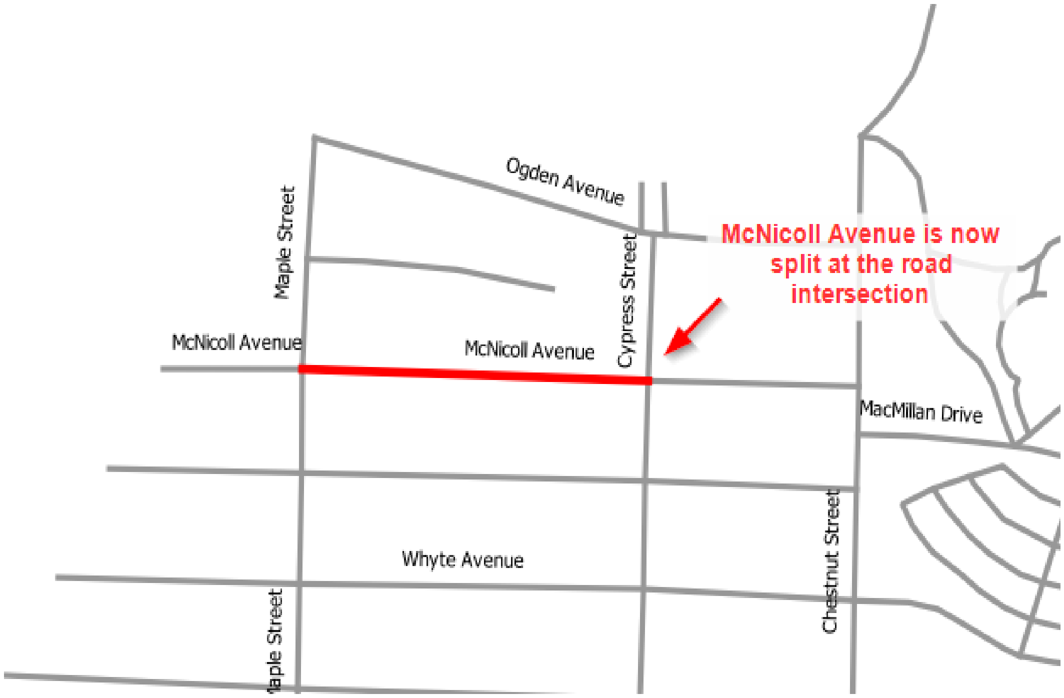
```
{
  "type": "FeatureCollection",
  "features": [
    {
      "geometry": {
        "type": "Point",
        "coordinates": [
          -123.13784476,
          49.2867852829887
        ]
      },
      "type": "Feature",
      "id": null,
      "properties": {
        "name": "Lord Roberts Elem."
      }
    },
    {
      "geometry": {
        "type": "Point",
        "coordinates": [
          -123.12980601,
          49.2830917109887
        ]
      },
```

Splitting LineStrings at intersections using ST_Node



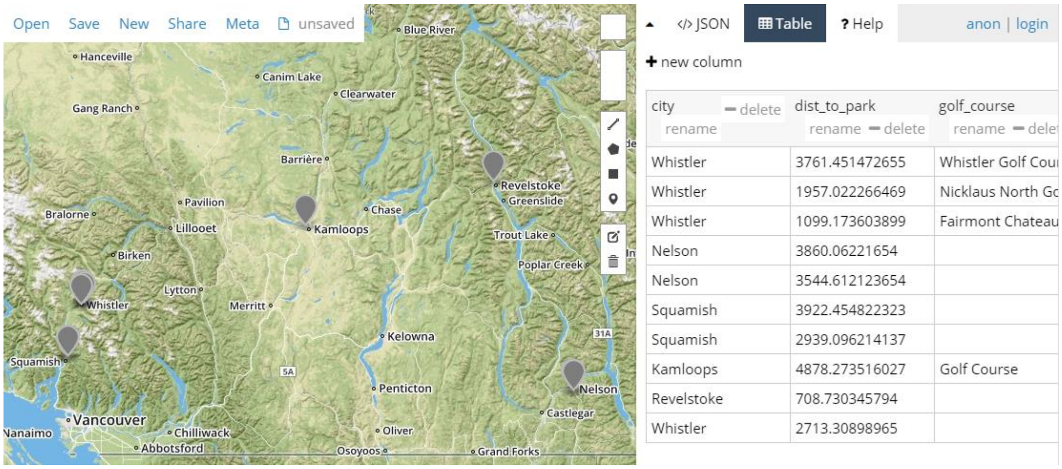


Linestring McNicoll Avenue is not split at the road intersection here



McNicoll Avenue is now split at the road intersection

Conducting a complex spatial analysis query using ST_Distance()

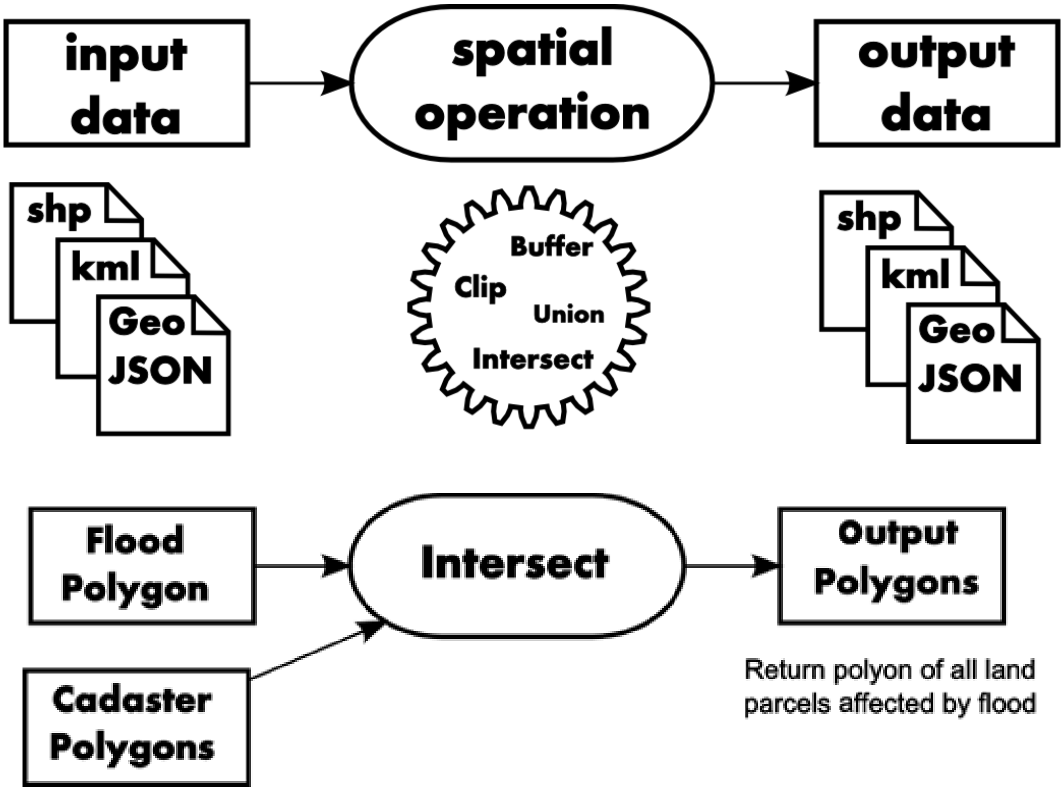


city	dist_to_park	golf_course
Whistler	3761.451472655	Whistler Golf Cou
Whistler	1957.022266469	Nicklaus North Gc
Whistler	1099.173603899	Fairmont Chateau
Nelson	3860.06221654	
Nelson	3544.612123654	
Squamish	3922.454822323	
Squamish	2939.096214137	
Kamloops	4878.273516027	Golf Course
Revelstoke	708.730345794	
Whistler	2713.30898965	

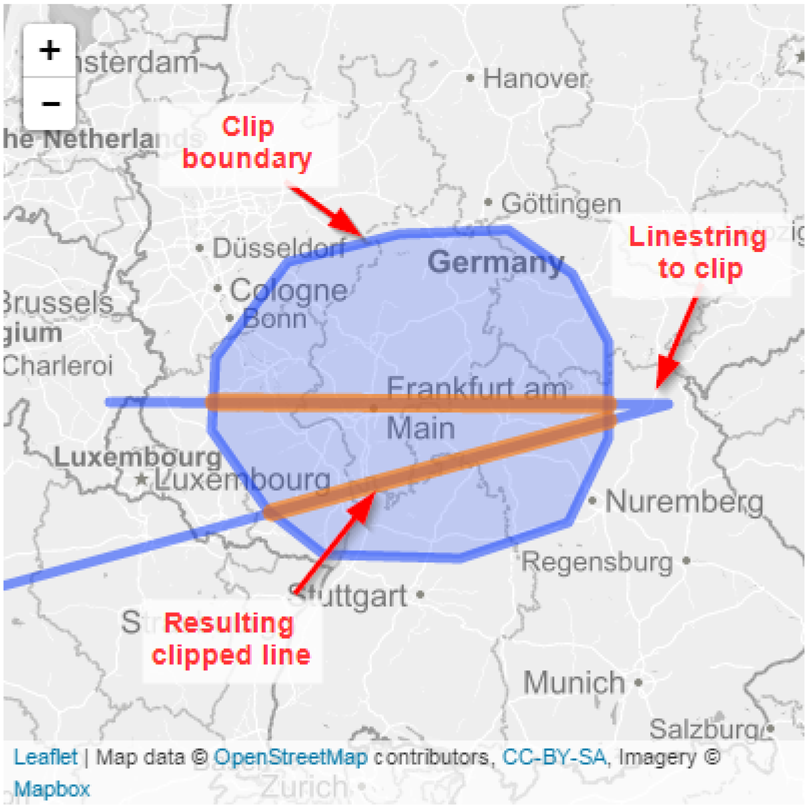
5

Vector Analysis

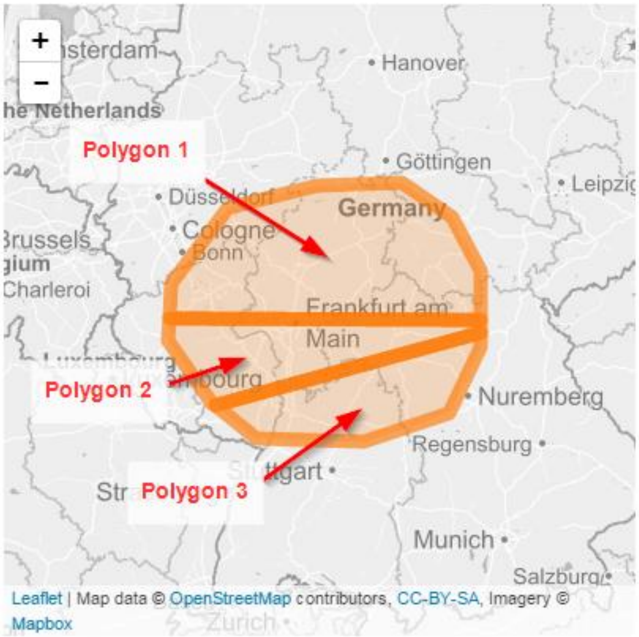
Introduction



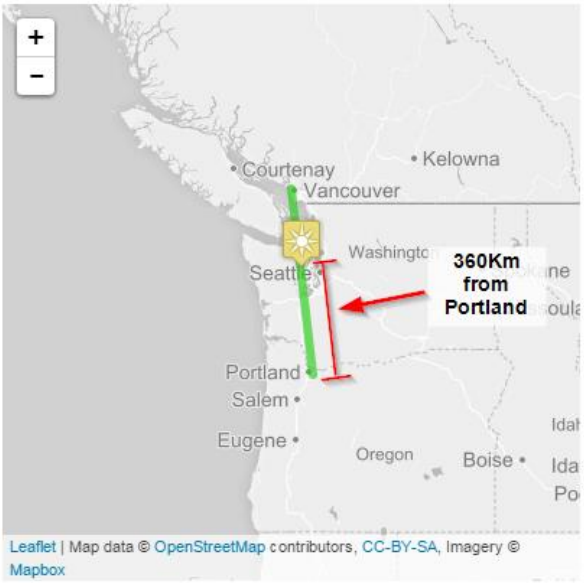
Clipping LineStrings to an area of interest



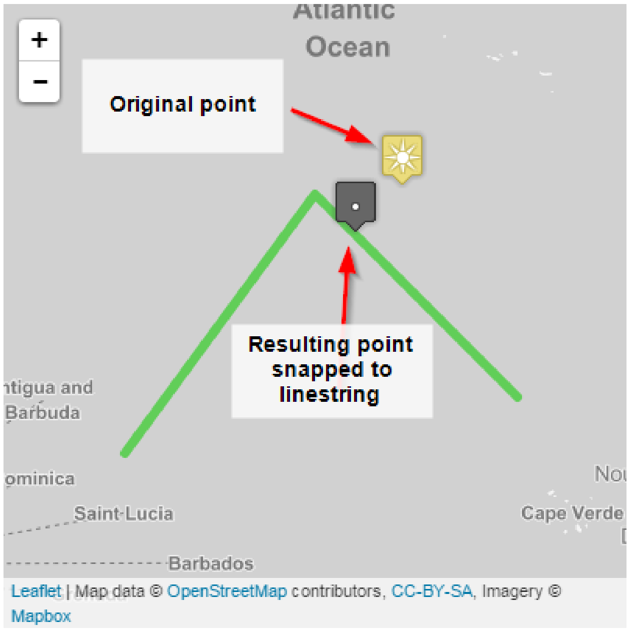
Splitting polygons with lines



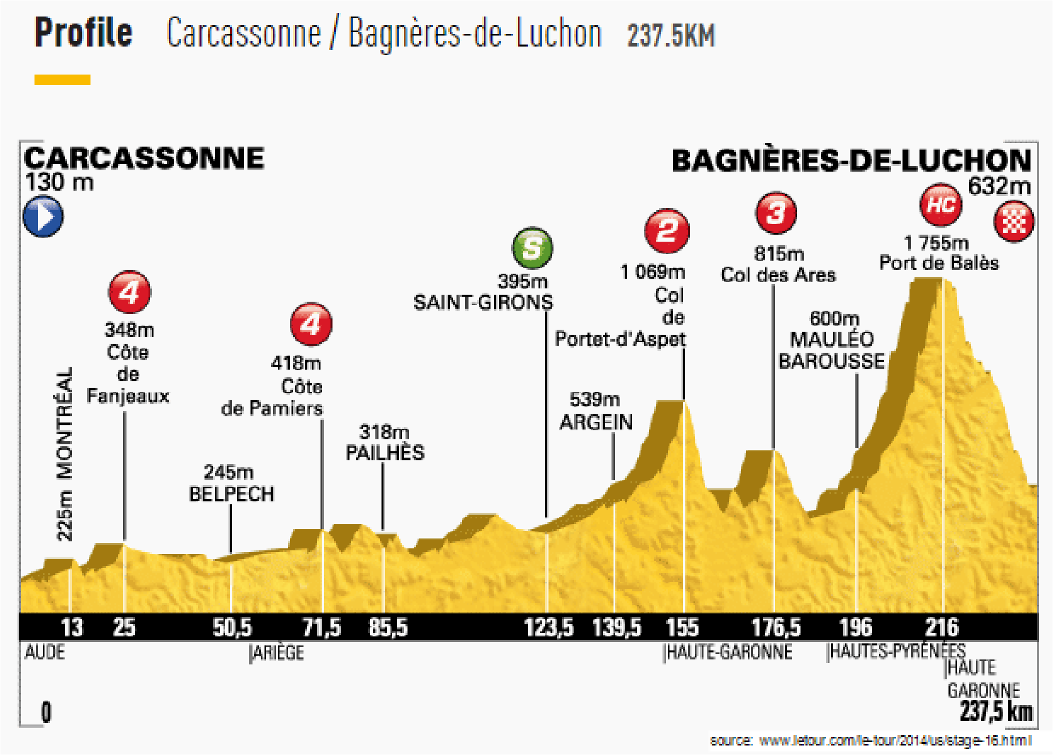
Finding the location of a point on a line using linear referencing



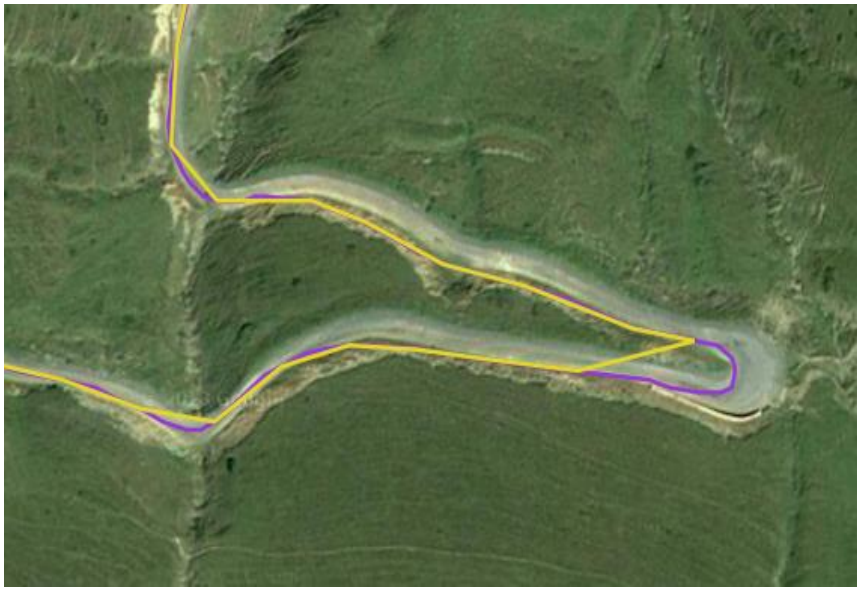
Snapping a point to the nearest line


+
−
Atlantic
Ocean
Original point
Resulting point snapped to linestring
ntigua and
Barbuda
ominica
Saint Lucia
Barbados
Cape Verde
Nou
Leaflet | Map data © OpenStreetMap contributors, CC-BY-SA, Imagery © Mapbox

Calculating 3D ground distance and total elevation gain



source: www.letour.com/le-tour/2014/us/stage-16.html

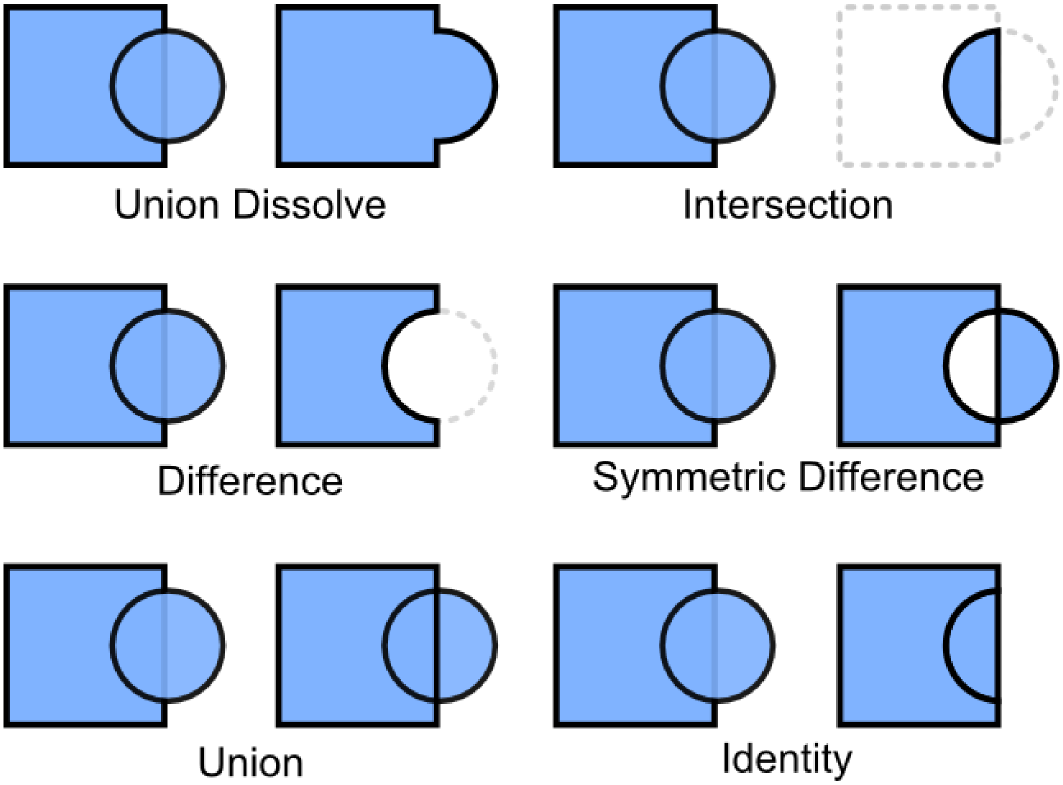


Data source: http://www.mapcycle.com.au/LeTour2014/#

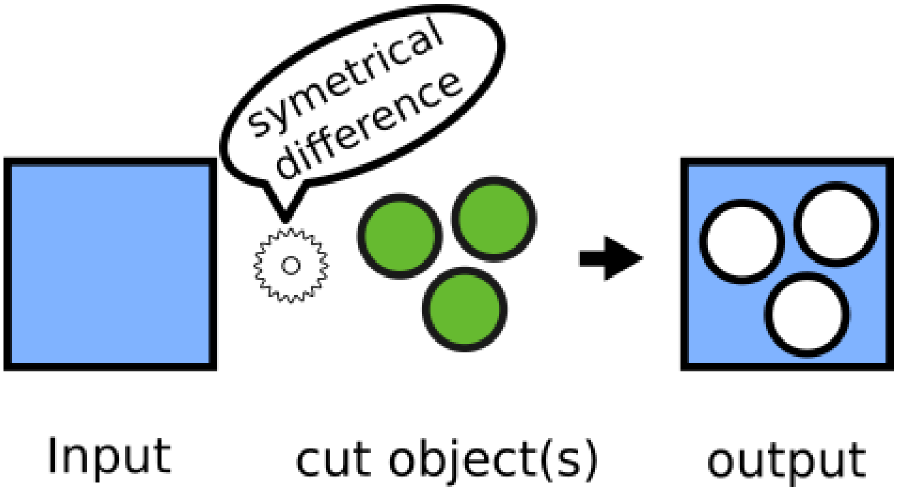
6

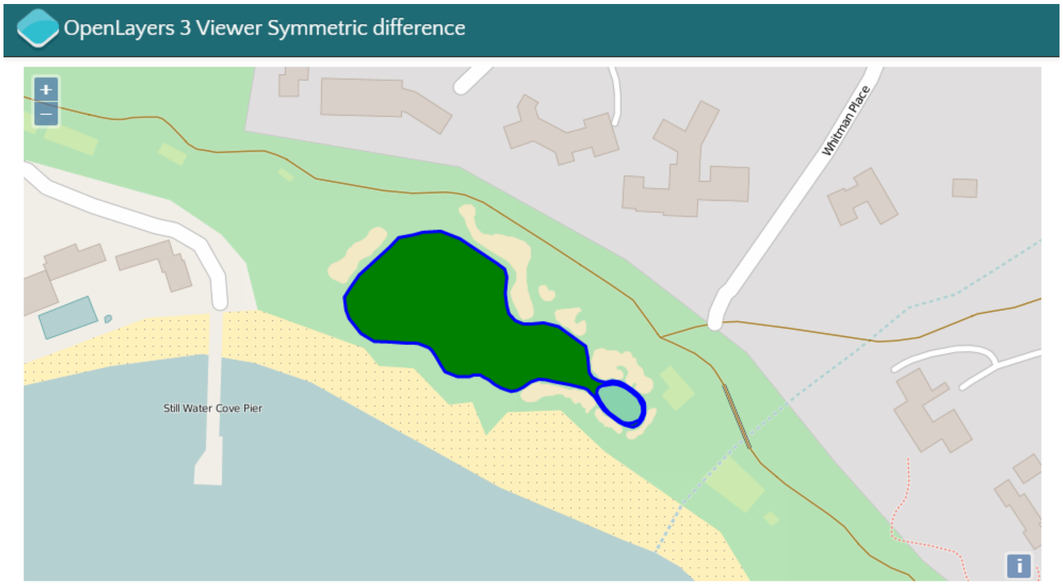
Overlay Analysis

Introduction

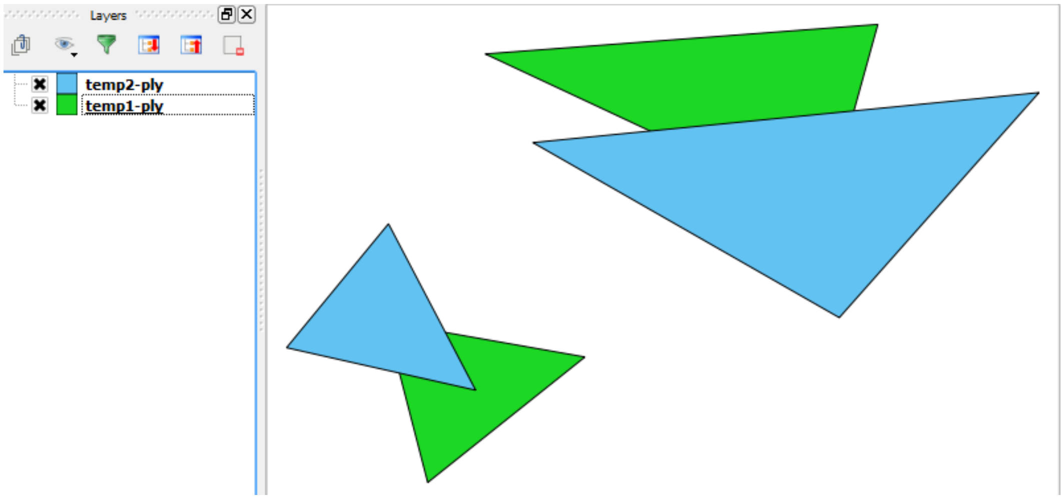


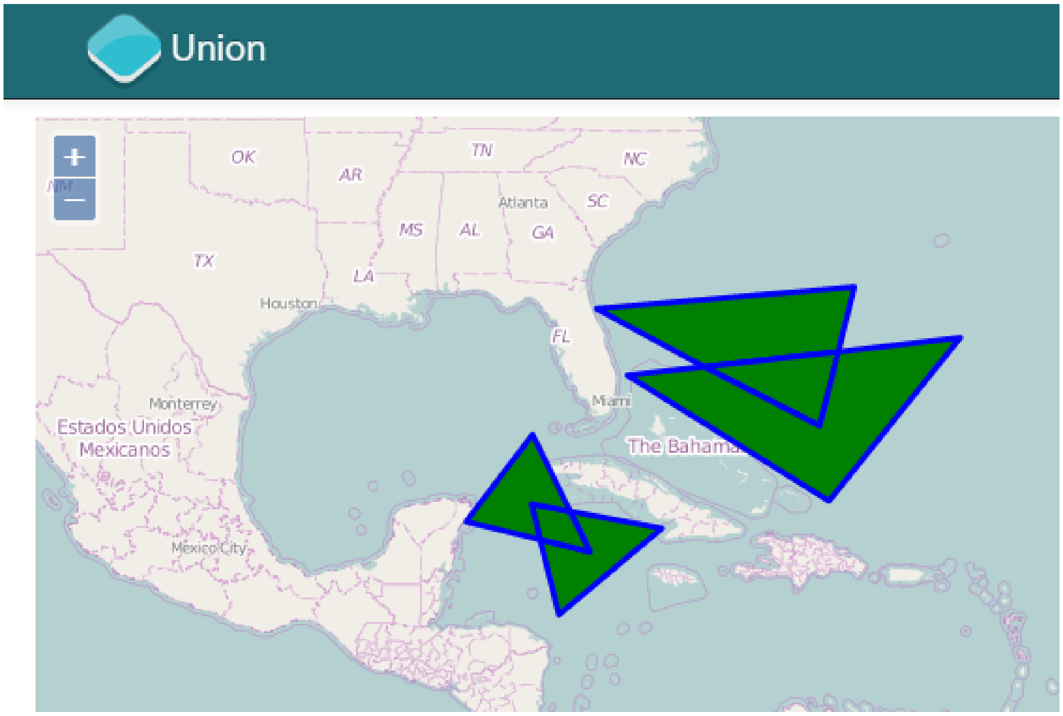
Punching holes in polygons with a symmetric difference operation



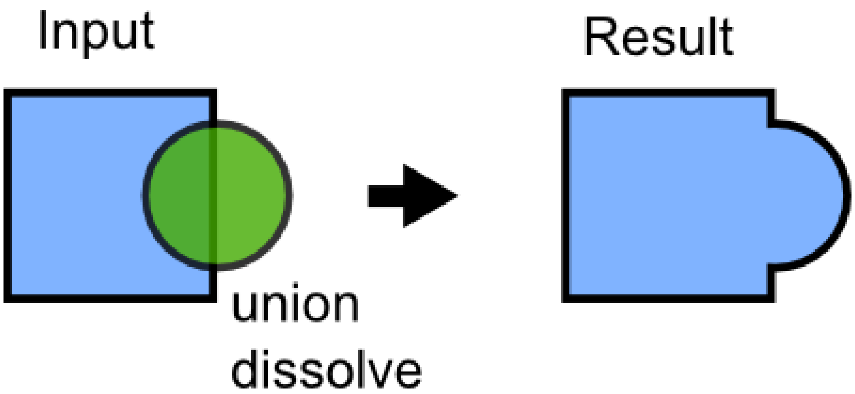

OpenLayers 3 Viewer Symmetric difference

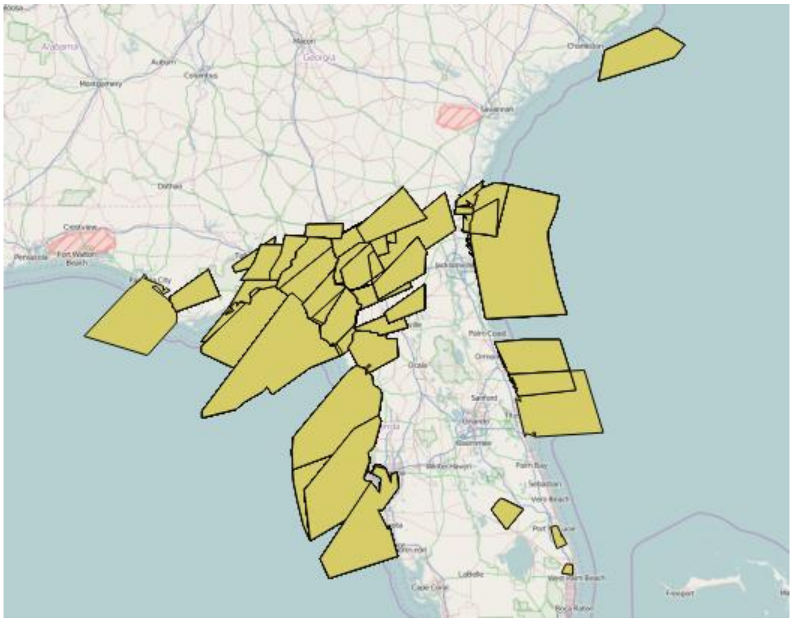
Union polygons without merging

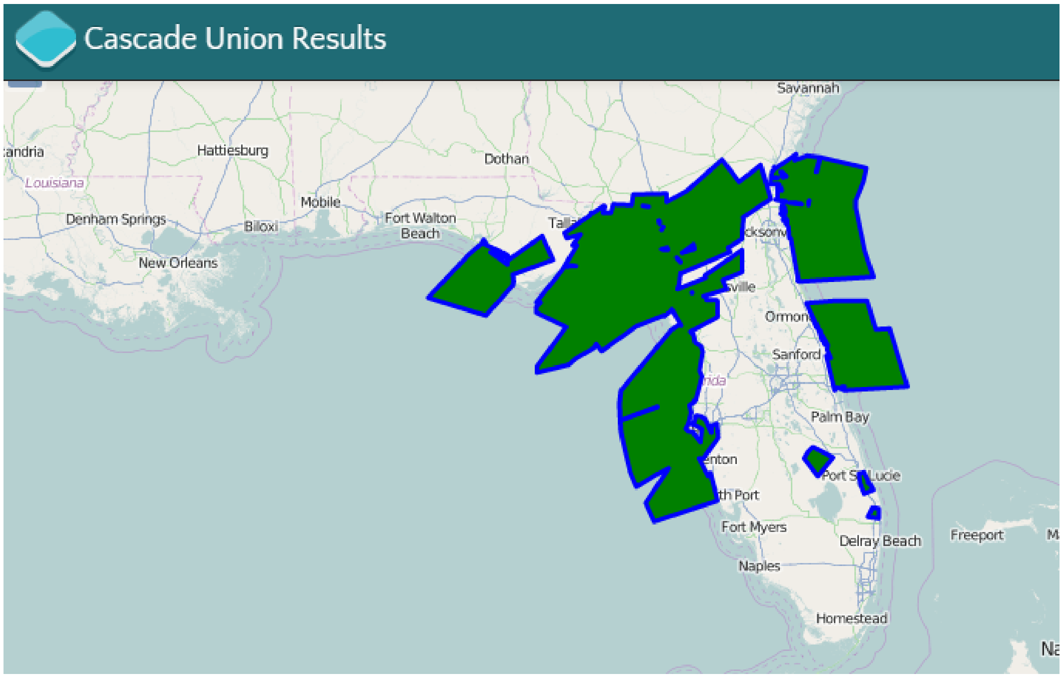




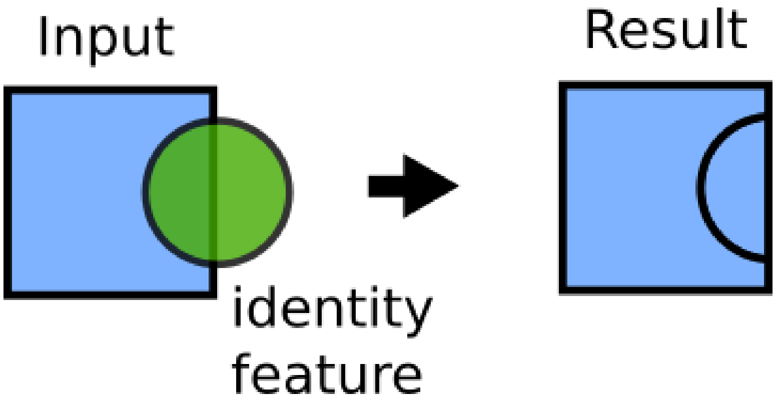
Union polygons with merging (dissolving)

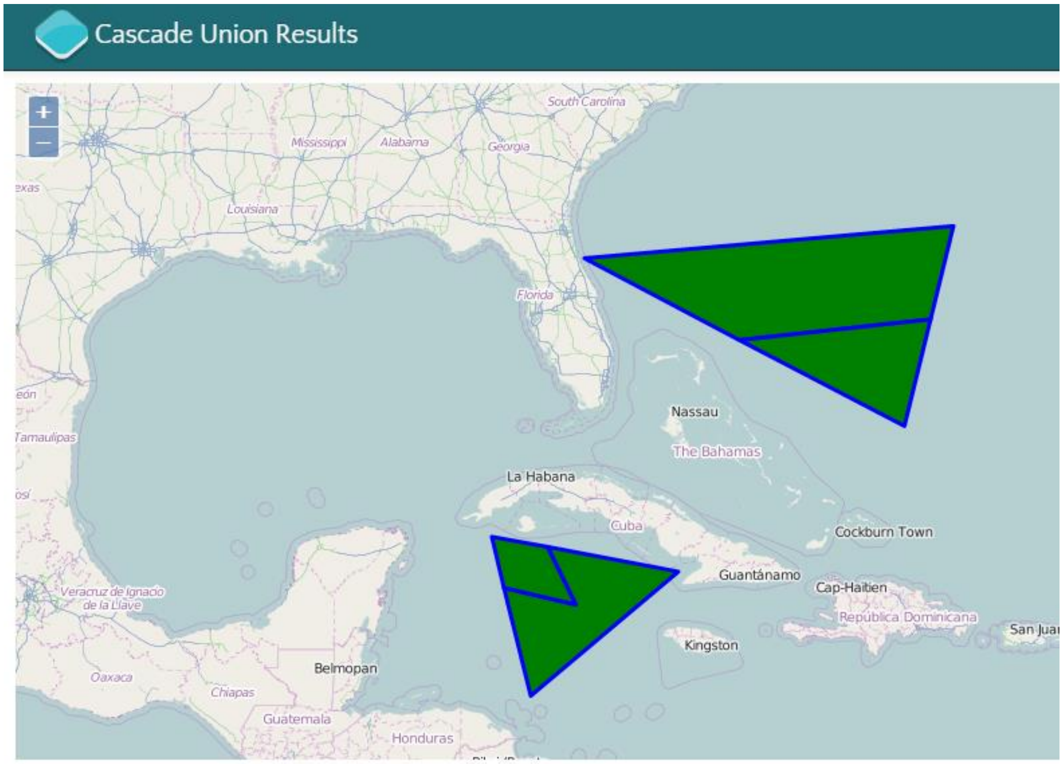






Performing an identity function (difference + intersection)

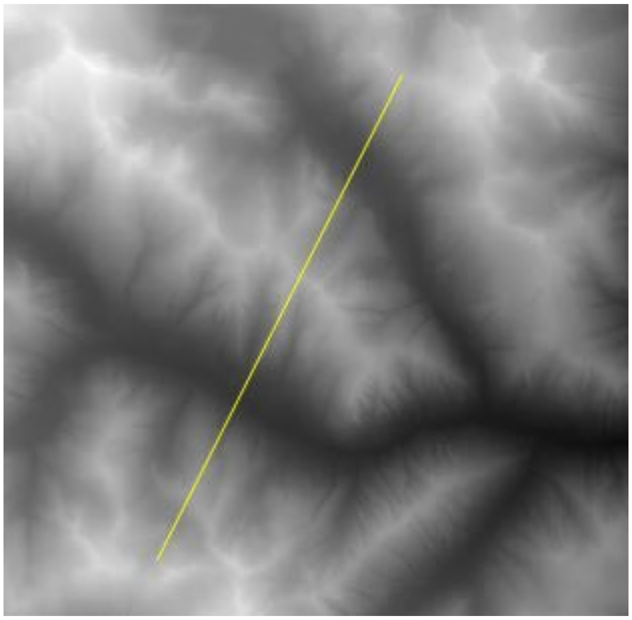


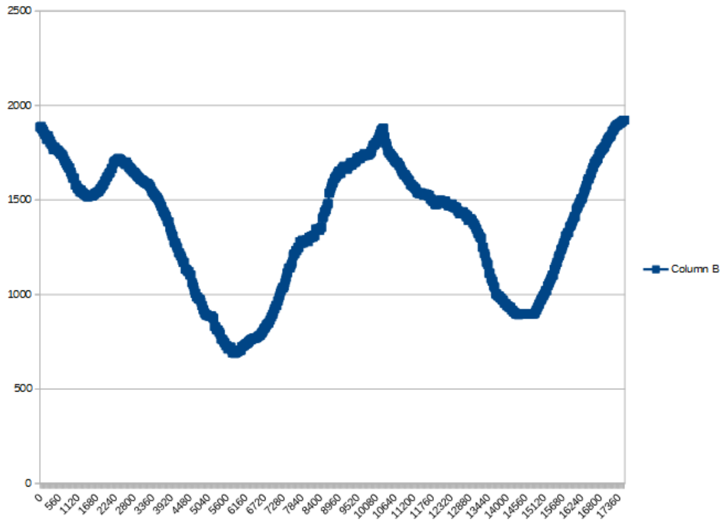


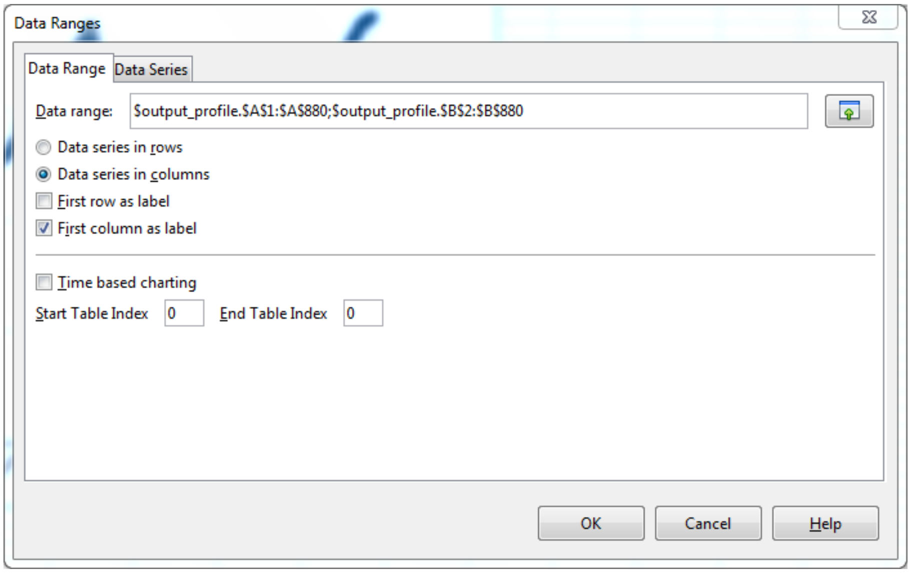
Cascade Union Results

7
Raster Analysis

Loading a DEM USGS ACSII CDED into PostGIS







Data range: $output_profile.$A$1:$A$880;$output_profile.$B$2:$B$880

○ Data series in rows

● Data series in columns

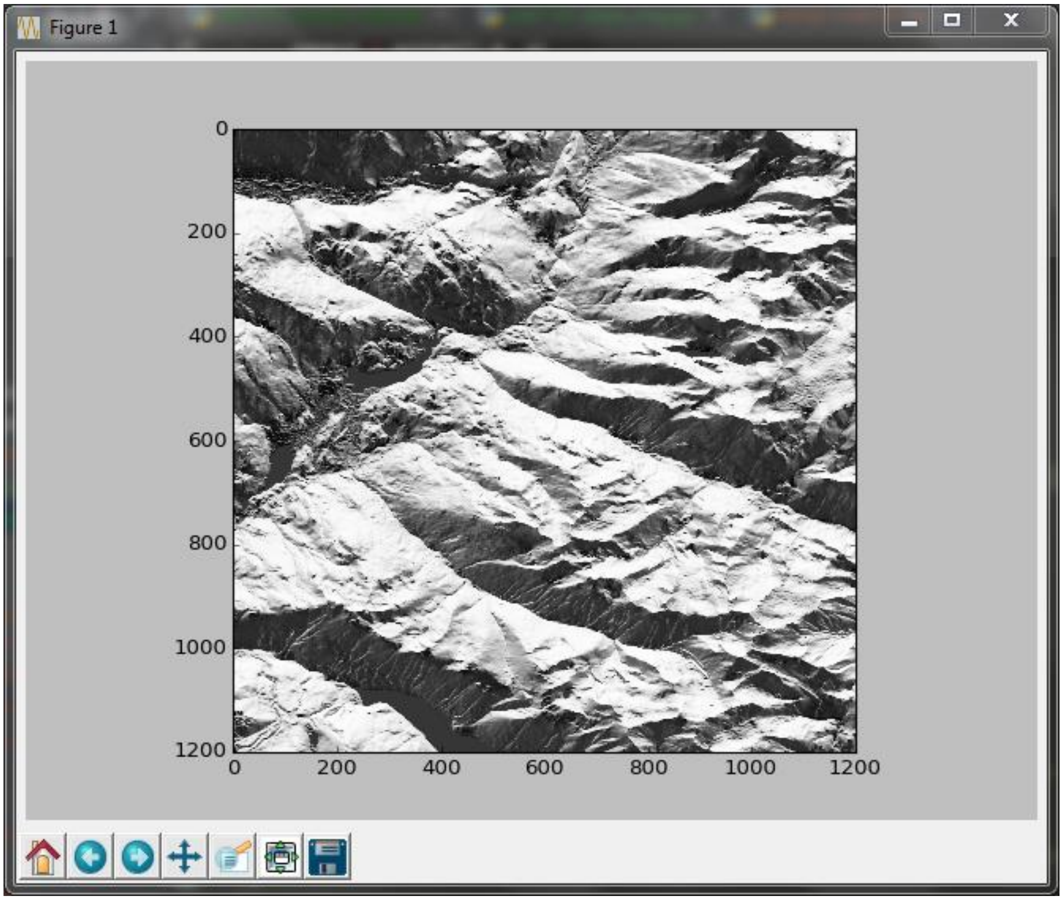
□ First row as label

☑ First column as label

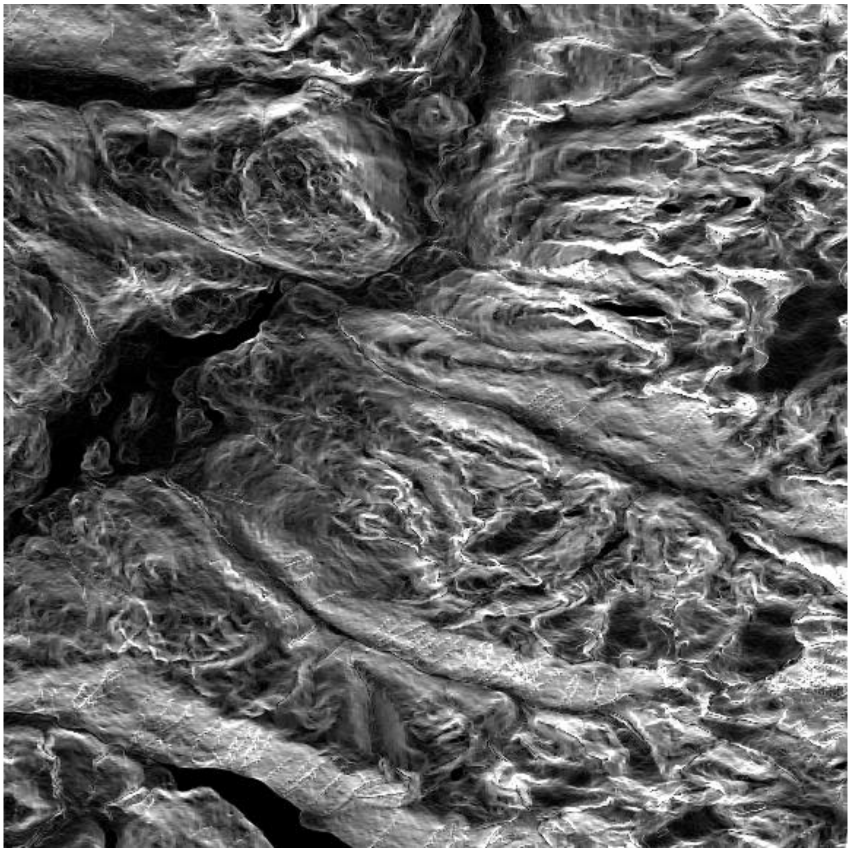
□ Time based charting

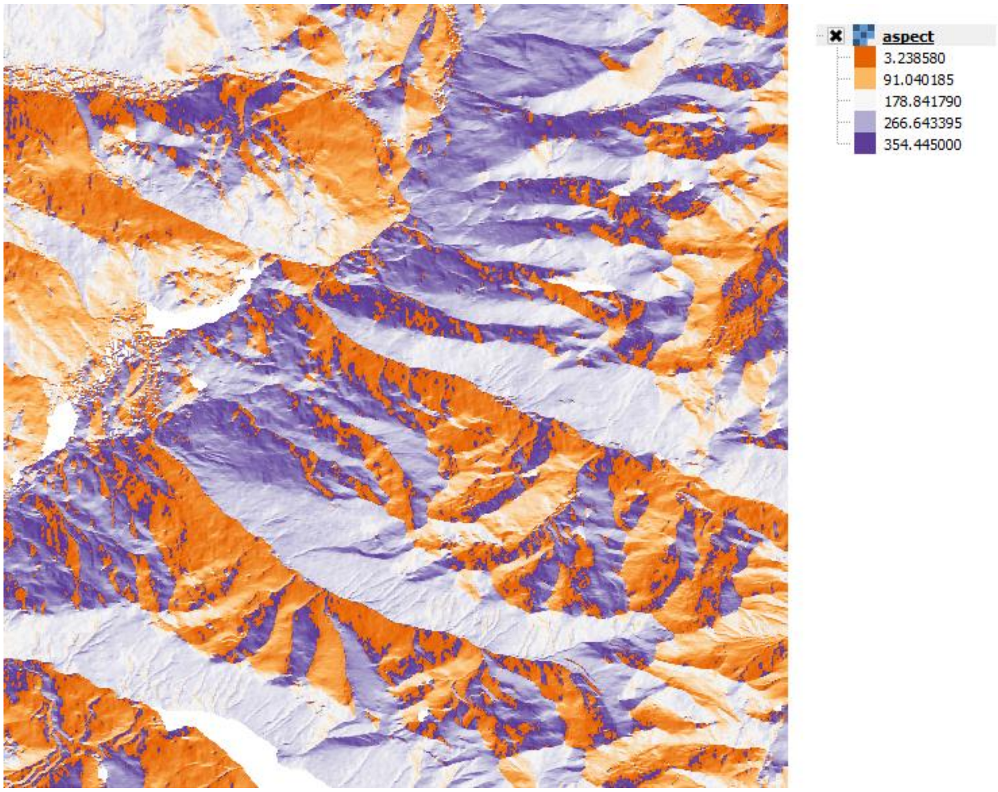
Start Table Index 0 End Table Index 0

Creating a hillshade raster from your DEM with ogr

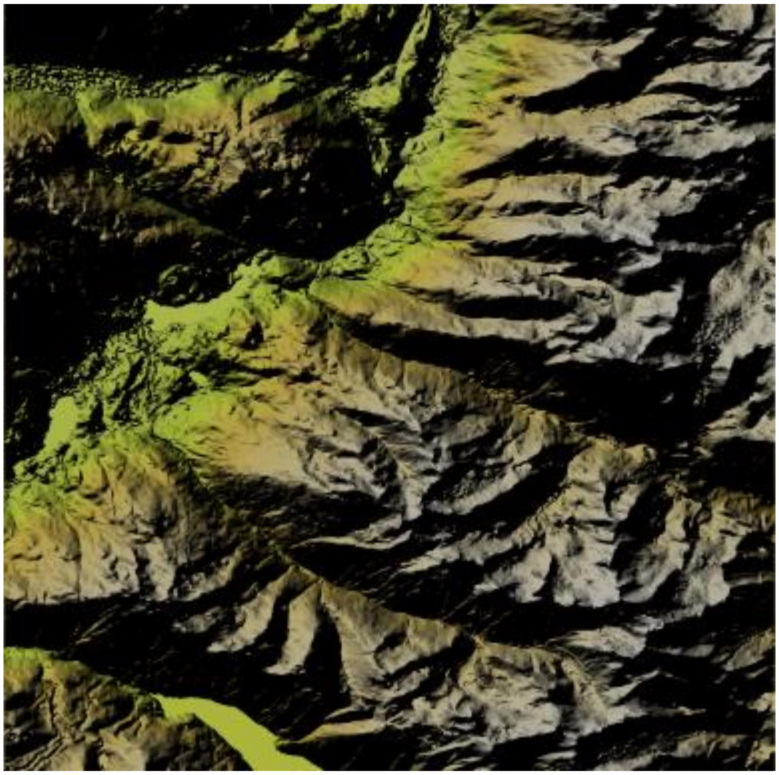


Generating slope and aspect images from your DEM




aspect
3.238580
91.040185
178.841790
266.643395
354.445000

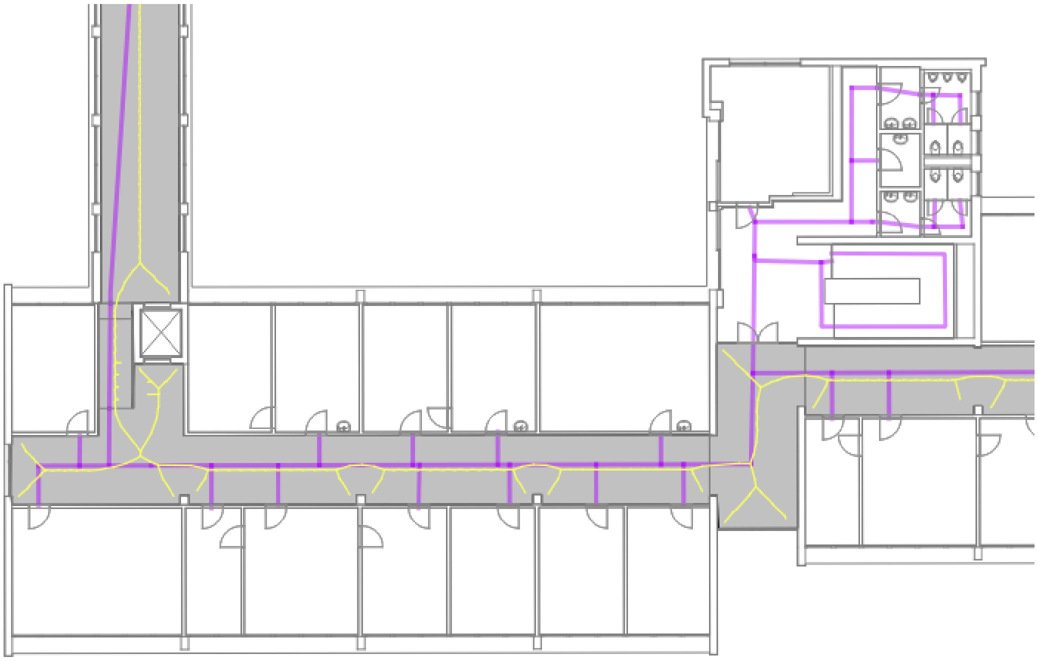
Merging rasters to generate a color relief map



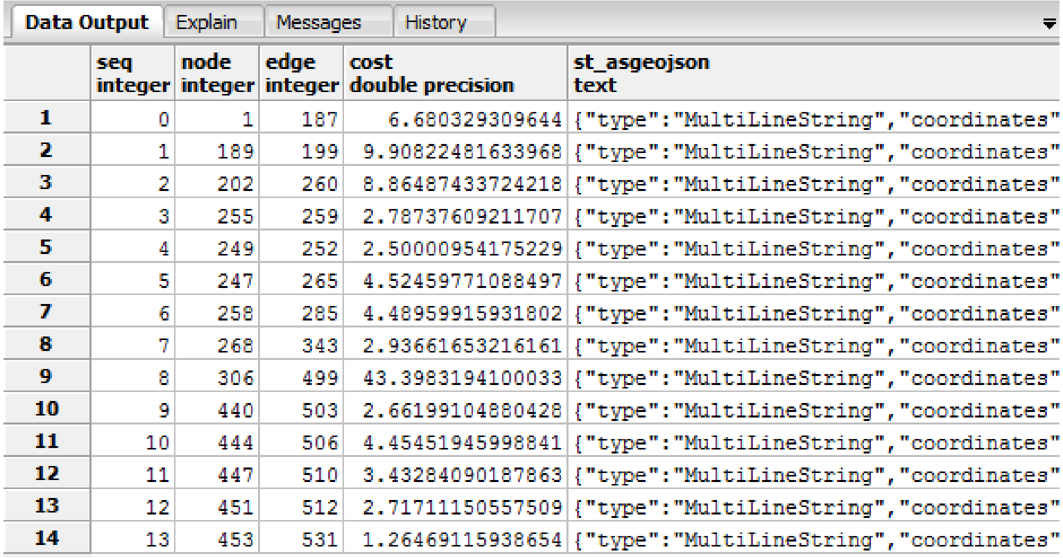
8

Network Routing Analysis

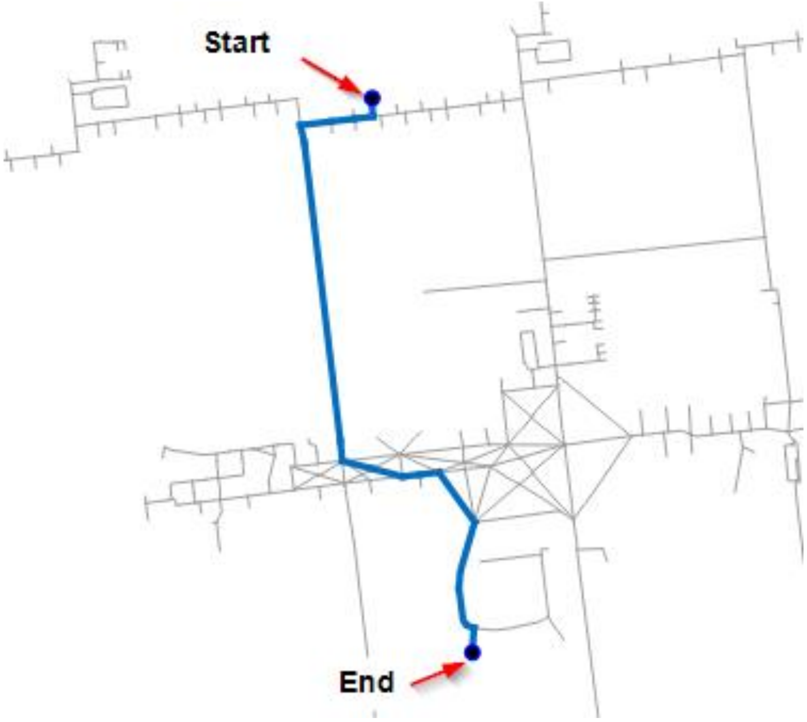
Introduction



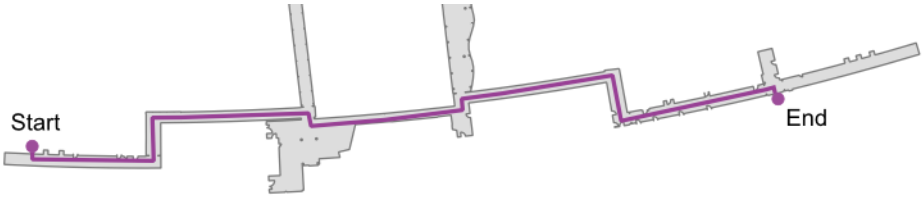
Finding the Dijkstra shortest path with pgRouting



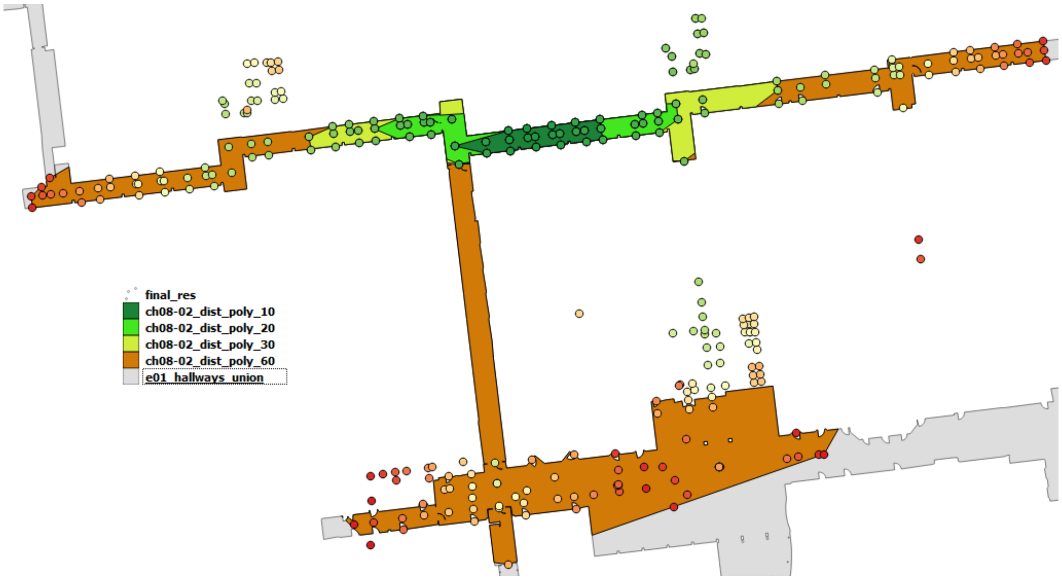
	seq	node	edge	cost	st_asgeojson
	integer	integer	integer	double precision	text
1	0	1	187	6.680329309644	{"type":"MultiLineString","coordinates"
2	1	189	199	9.90822481633968	{"type":"MultiLineString","coordinates"
3	2	202	260	8.86487433724218	{"type":"MultiLineString","coordinates"
4	3	255	259	2.78737609211707	{"type":"MultiLineString","coordinates"
5	4	249	252	2.50000954175229	{"type":"MultiLineString","coordinates"
6	5	247	265	4.52459771088497	{"type":"MultiLineString","coordinates"
7	6	258	285	4.48959915931802	{"type":"MultiLineString","coordinates"
8	7	268	343	2.93661653216161	{"type":"MultiLineString","coordinates"
9	8	306	499	43.3983194100033	{"type":"MultiLineString","coordinates"
10	9	440	503	2.66199104880428	{"type":"MultiLineString","coordinates"
11	10	444	506	4.45451945998841	{"type":"MultiLineString","coordinates"
12	11	447	510	3.43284090187863	{"type":"MultiLineString","coordinates"
13	12	451	512	2.71711150557509	{"type":"MultiLineString","coordinates"
14	13	453	531	1.26469115938654	{"type":"MultiLineString","coordinates"



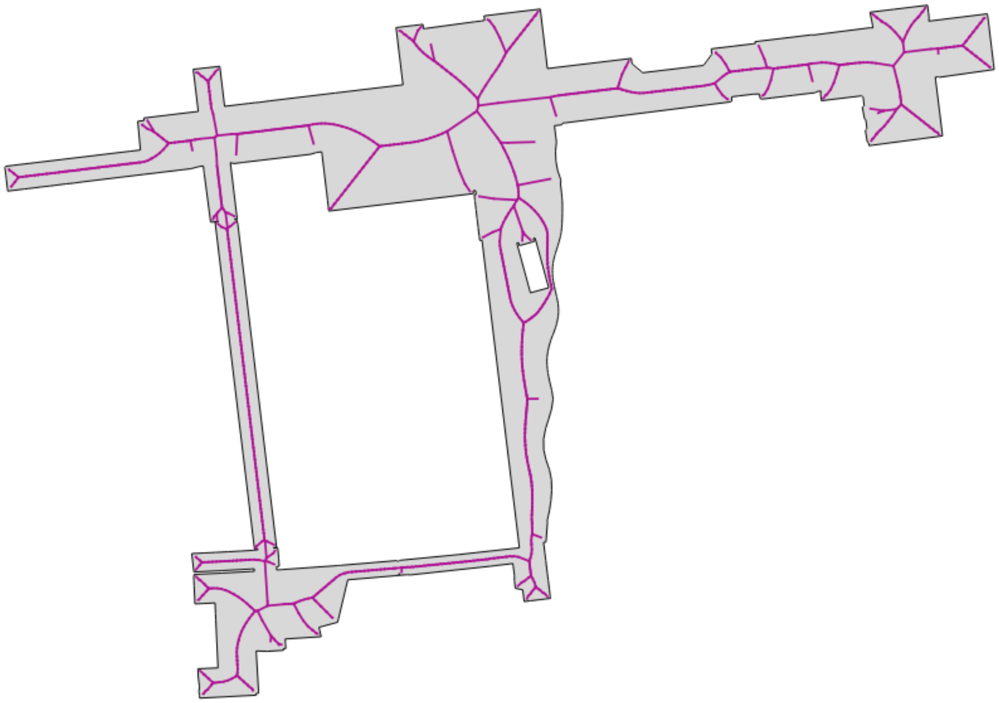
Finding the Dijkstra shortest path with NetworkX in pure Python



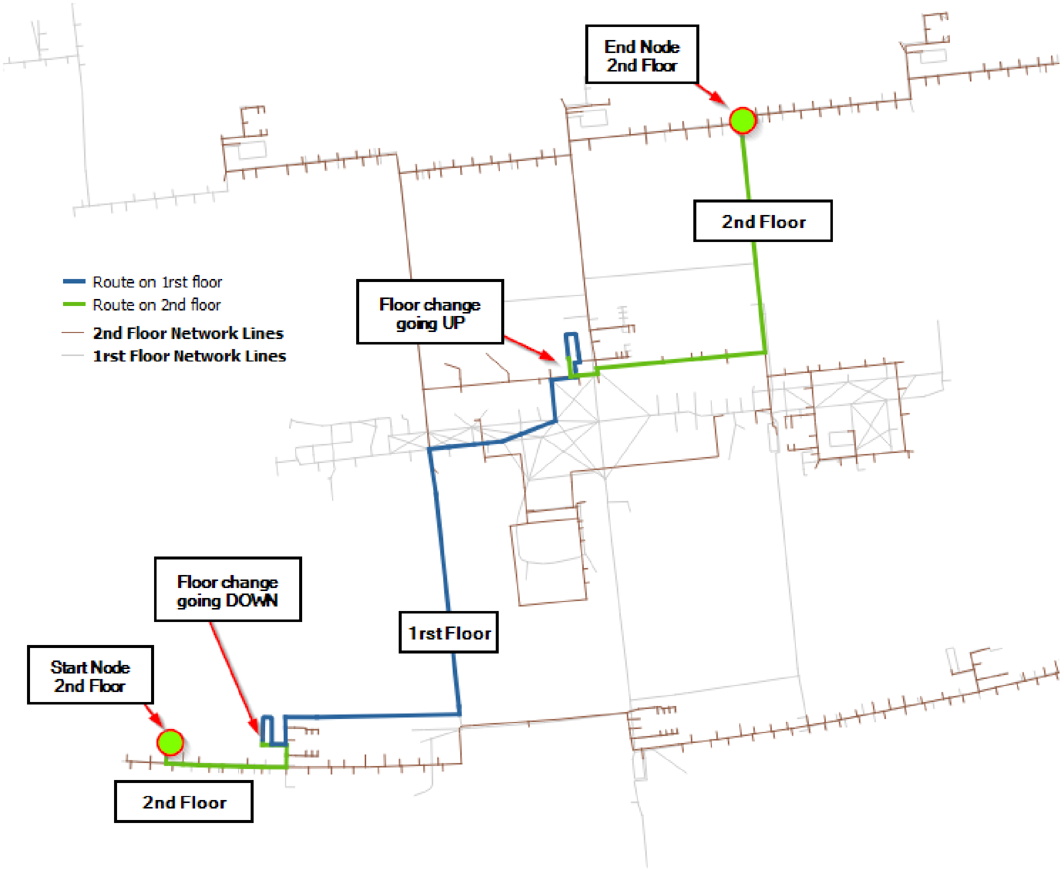
Generating evacuation polygons based on an indoor shortest path



Creating centerlines from polygons



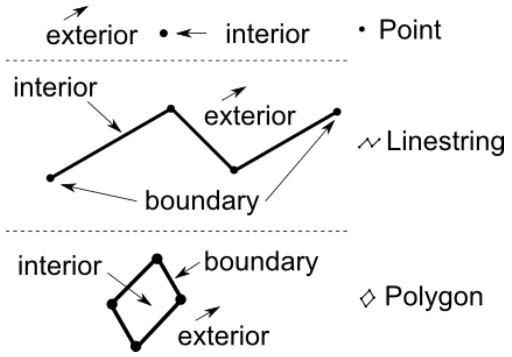
Building an indoor routing system in 3D



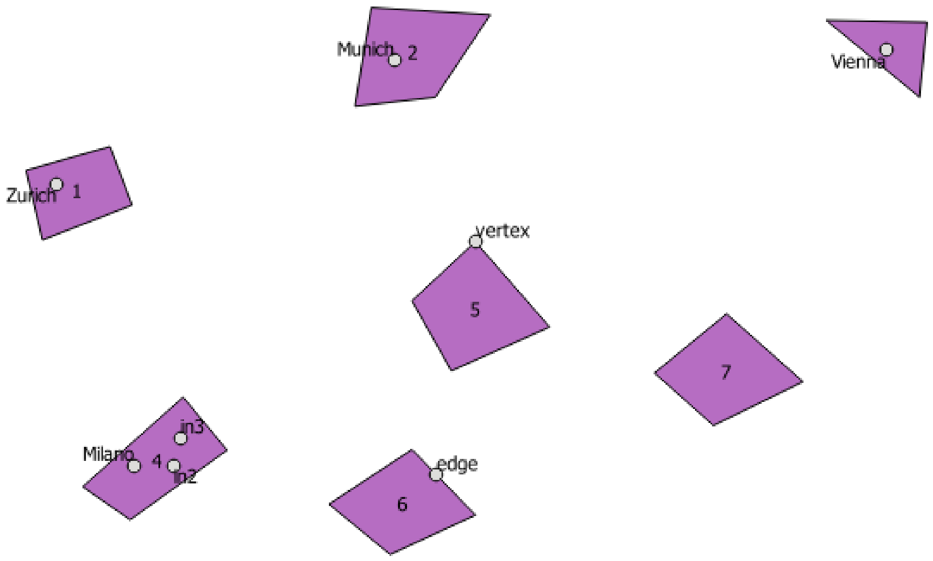
9

Topology Checking and Data Validation

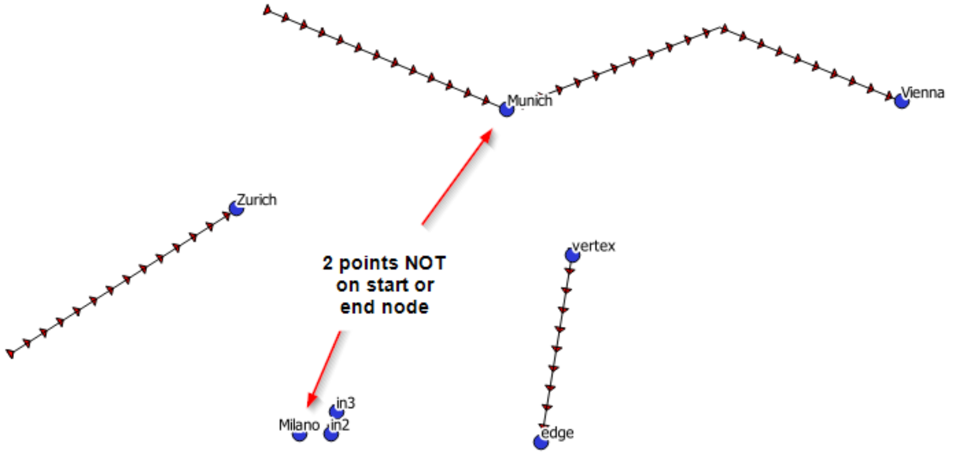
Introduction



Creating a rule – only one point inside a polygon

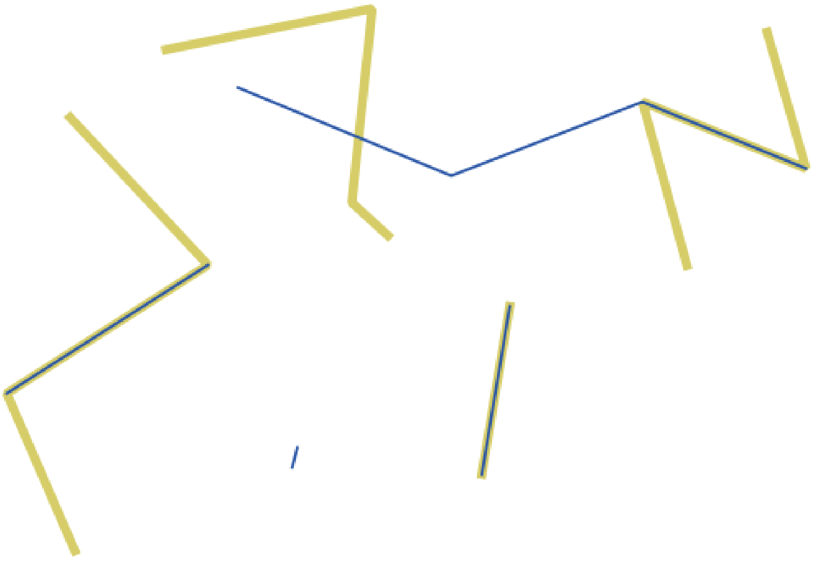


A point must be on the starting and ending nodes of a line only

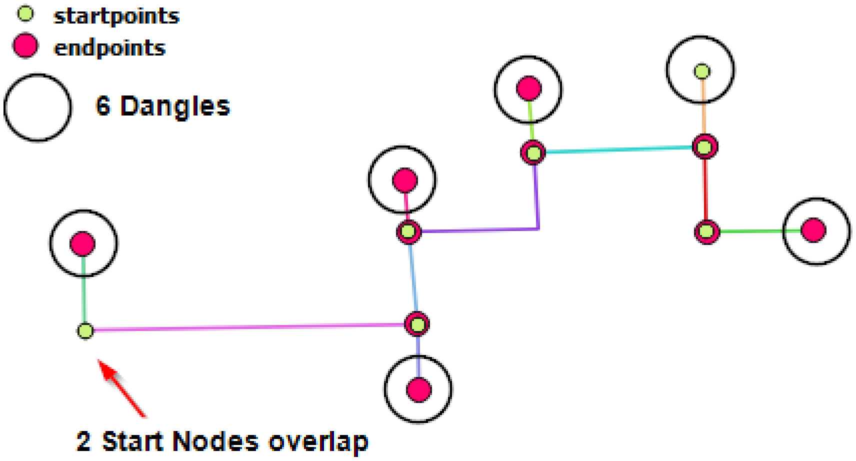


2 points NOT on start or end node

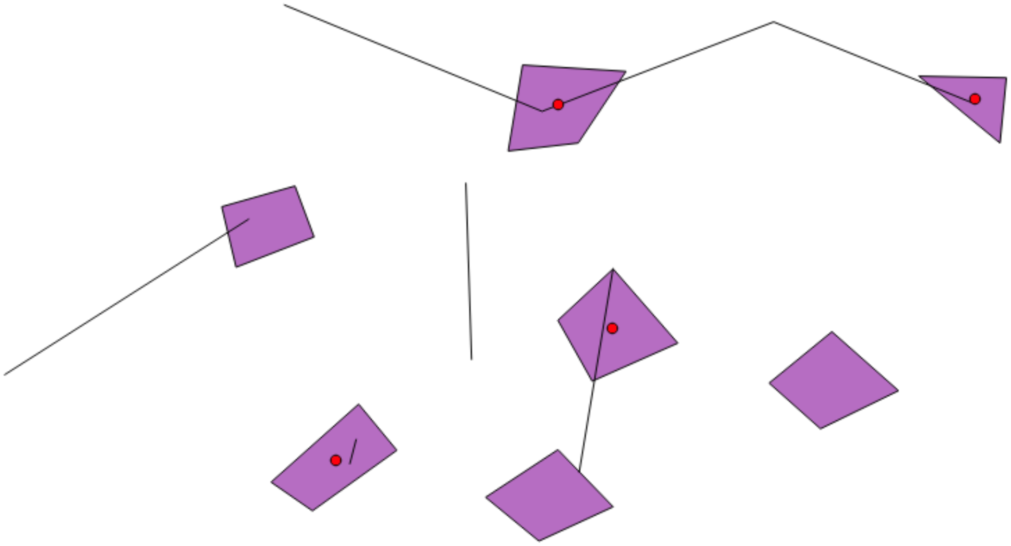
LineStrings must not overlap



A LineString must not have Dangles



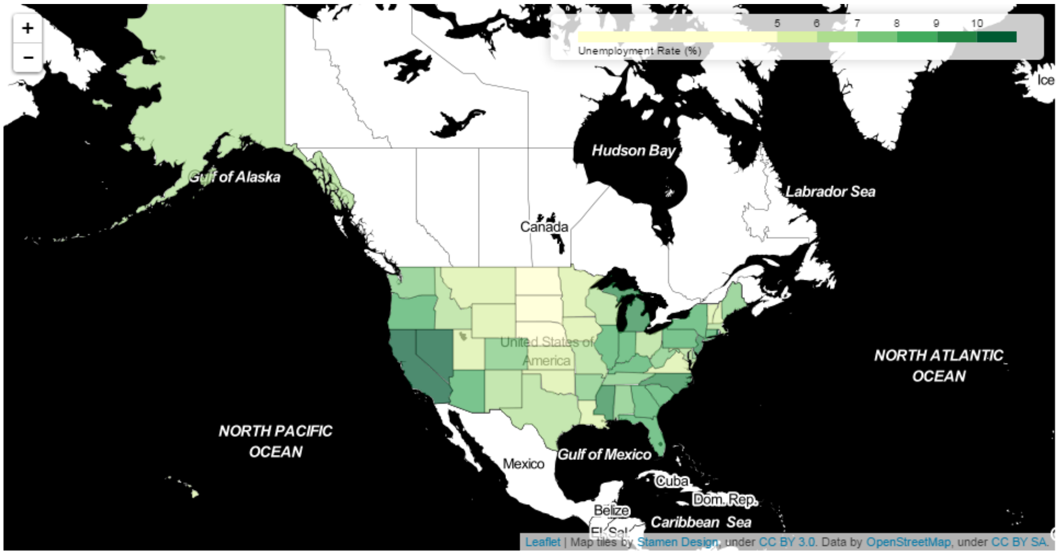
A polygon centroid must be within a specific distance of a line



10

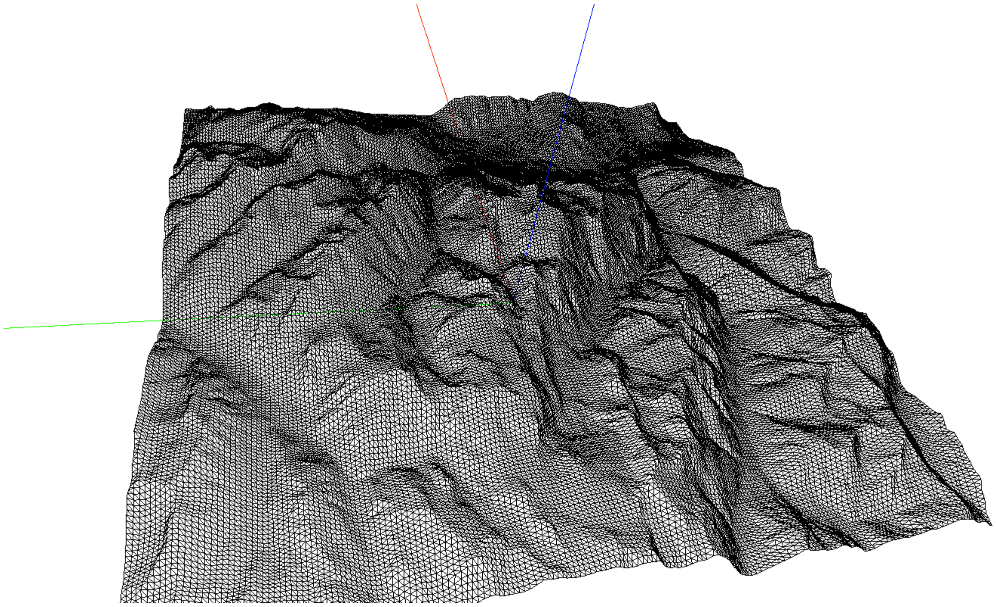
Visualizing Your Analysis

Generating a leaflet web map with Folium

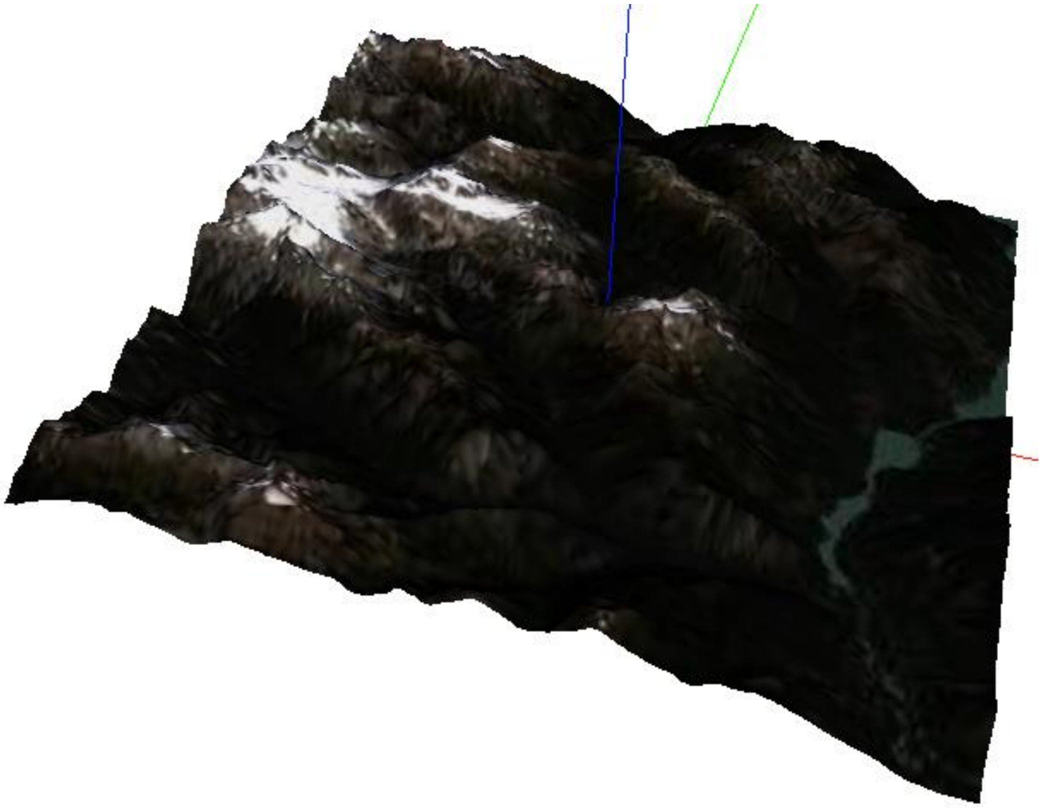


Setting up TileStache to serve tiles

Visualizing DEM data with Three.js



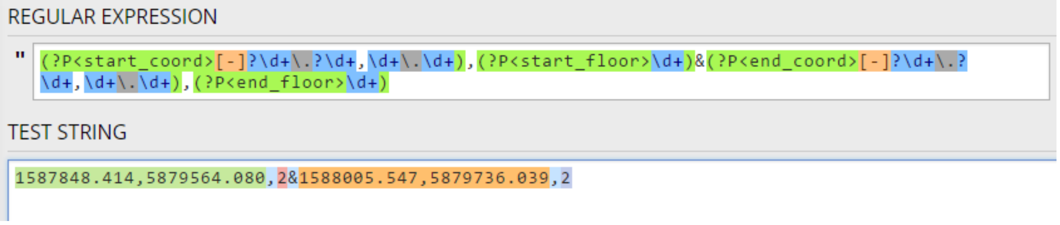
Draping an orthophoto over a DEM



11

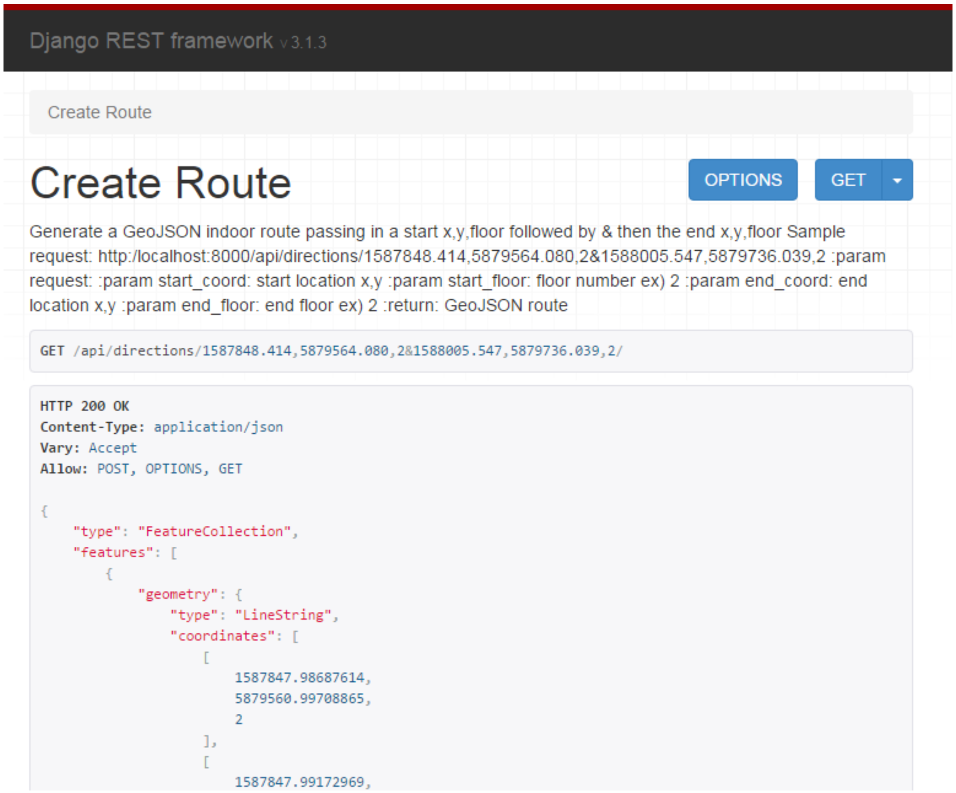
Web Analysis with GeoDjango

Creating an indoor web routing service



```
" (?P<start_coord>[-]?\d+\.?\d+,\d+\.\d+),(?P<start_floor>\d+)&(?P<end_coord>[-]?\d+\.?
\d+,\d+\.\d+),(?P<end_floor>\d+)
```

```
1587848.414,5879564.080,2&1588005.547,5879736.039,2
```



Create Route

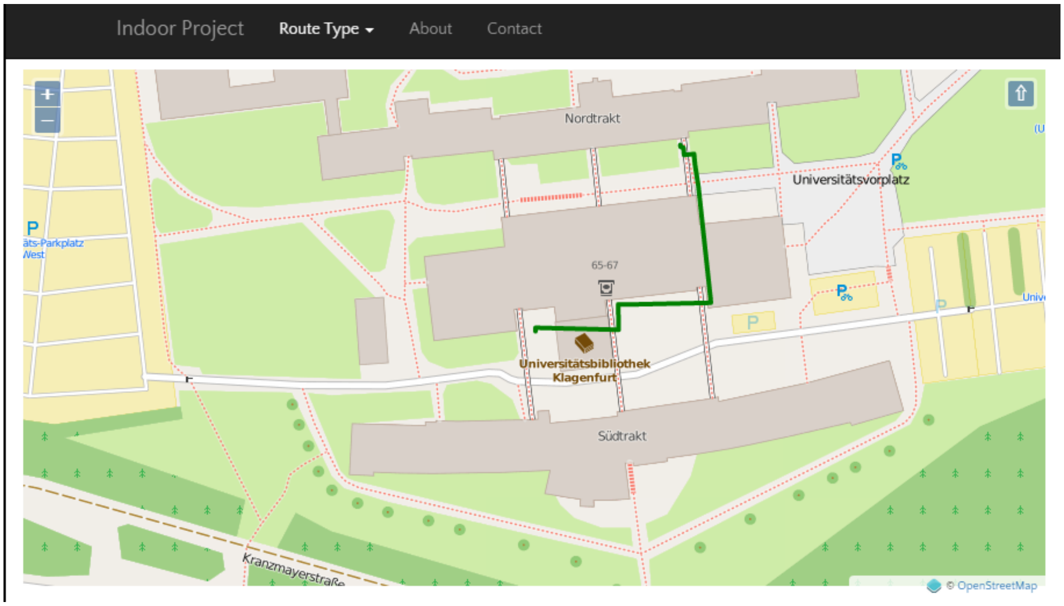
Generate a GeoJSON indoor route passing in a start x,y,floor followed by & then the end x,y,floor Sample request: http:/localhost:8000/api/directions/1587848.414,5879564.080,2&1588005.547,5879736.039,2 :param request: :param start_coord: start location x,y :param start_floor: floor number ex) 2 :param end_coord: end location x,y :param end_floor: end floor ex) 2 :return: GeoJSON route

```
GET /api/directions/1587848.414,5879564.080,2&1588005.547,5879736.039,2/
```

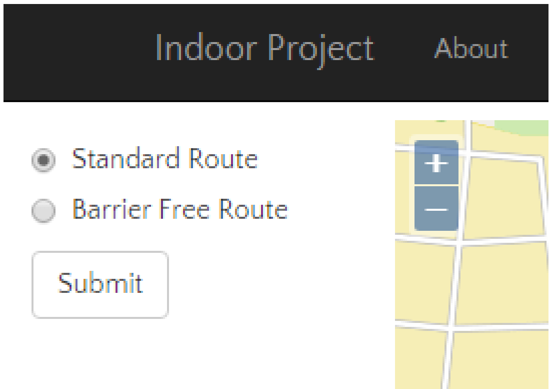
```
HTTP 200 OK
Content-Type: application/json
Vary: Accept
Allow: POST, OPTIONS, GET

{
    "type": "FeatureCollection",
    "features": [
        {
            "geometry": {
                "type": "LineString",
                "coordinates": [
                    [
                        1587847.98687614,
                        5879560.99708865,
                        2
                    ],
                    [
                        1587847.99172969,
```

Visualizing an indoor routing service

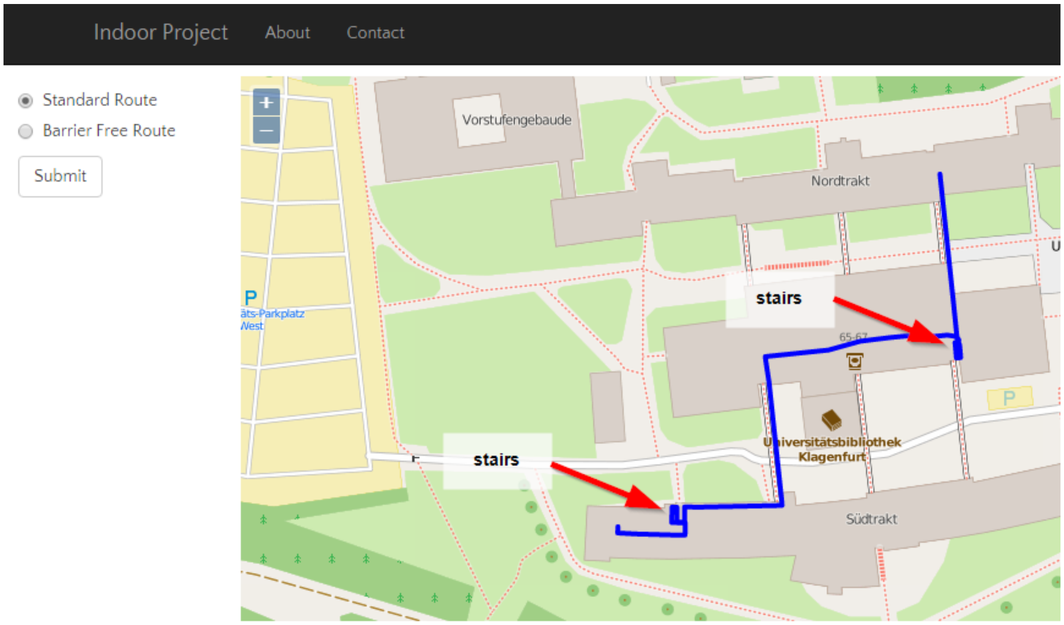


Creating an indoor route-type service



● Standard Route

○ Barrier Free Route

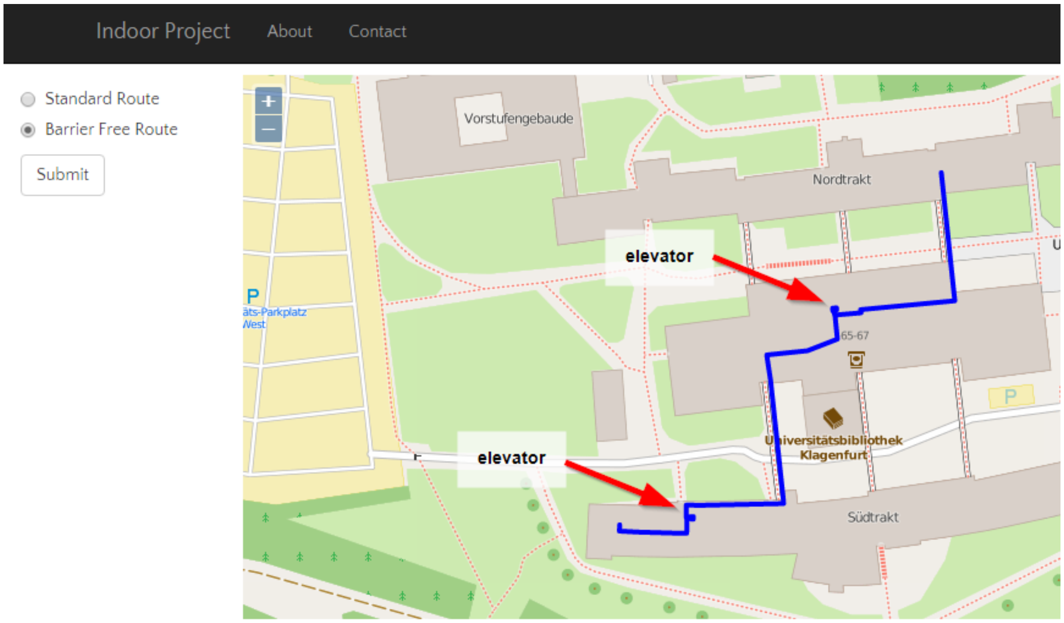


Indoor Project About Contact

● Standard Route

○ Barrier Free Route

Submit



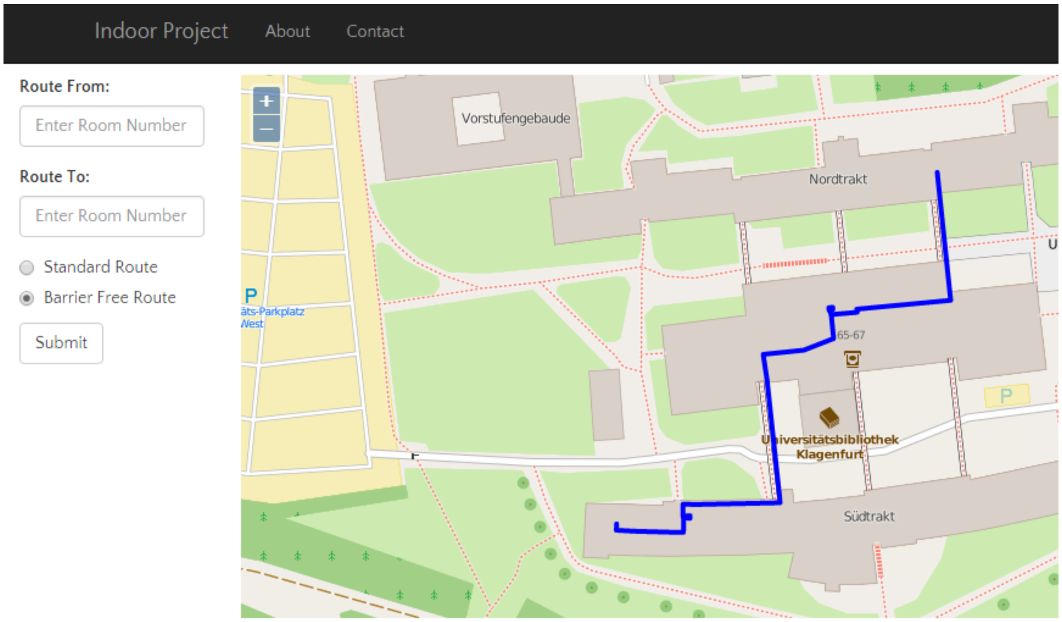
Indoor Project About Contact

○ Standard Route

● Barrier Free Route

Submit

Creating an indoor route from room to room

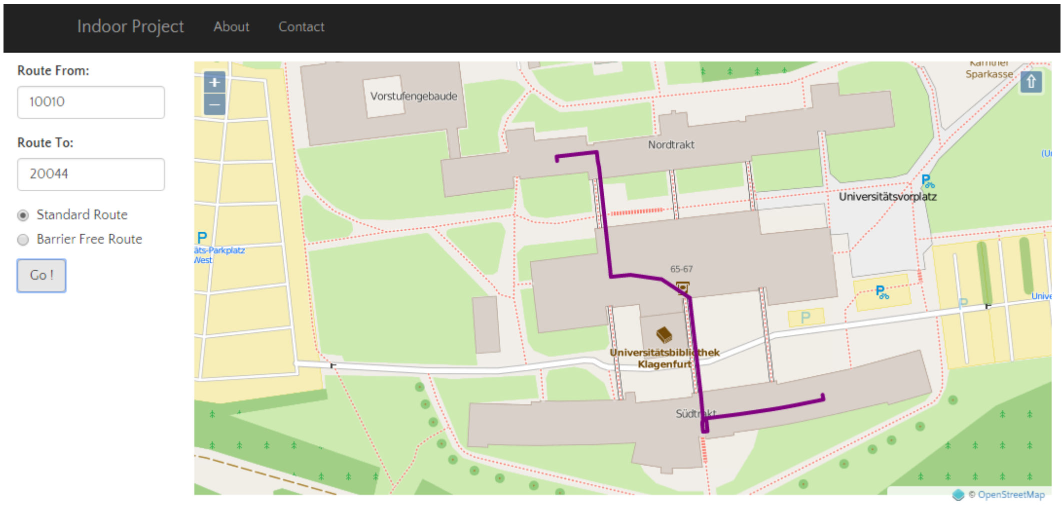


Route From: Enter Room Number

Route To: Enter Room Number

○ Standard Route

● Barrier Free Route



Route From: 10010

Route To: 20044

● Standard Route

○ Barrier Free Route