

MapCSS

ein neuer Standard für Kartographie

Stephan Bösch-Plepelits

<http://plepe.at> - <http://twitter.com/plepe>
plepe@openstreetmap.at

Die Folien zum Vortrag: <http://plepe.at/185>

Ablauf

- Mini-Einführung in das OpenStreetMap Datenmodell
- Was ist MapCSS? Warum MapCSS?
- Überblick über verschiedene Libraries:
 - Overpass Turbo
 - JOSM
 - Mapnik + pgmapcss
- Beispiele
- Zeit für Fragen

OpenStreetMap Datenmodell

- **Node (Punkt)**
hat Koordinaten und optional Tags.
- **Way (Weg, Fläche)**
besteht aus Nodes, hat Tags.
- **Relation (Beziehung)**
besteht aus Nodes, Ways und Relations, hat Tags.
- **Tags (Attribute)**
Beliebige Anzahl an Tags.
Jeder „Tag“ hat einen Schlüssel und einen Wert.

OpenStreetMap Datenmodell

The image displays a map of Vienna with several data tables overlaid, illustrating the OpenStreetMap data model. The tables are:

- Table 1 (Top Left):** A table with two columns: 'Schlüssel' (Key) and 'Wert' (Value). It lists attributes for a bus route, such as 'from: Dr. Karl Renner Ring', 'name: Bus 48A: Dr. Karl Renner Ring', 'operator: Wiener Linien', and 'type: route'.
- Table 2 (Top Center):** A table with two columns: 'Schlüssel' (Key) and 'Wert' (Value). It lists attributes for a Billa supermarket, such as 'name: Billa', 'shop: supermarket', and 'wheelchair: limited'.
- Table 3 (Middle Right):** A table with two columns: 'Schlüssel' (Key) and 'Wert' (Value). It lists attributes for a highway, such as 'highway: traffic_signals'.
- Table 4 (Middle Left):** A table with two columns: 'Schlüssel' (Key) and 'Wert' (Value). It lists attributes for a street, such as 'name: Neustiftgasse', 'maxspeed: 30', and 'source:maxspeed: AT:zone:30'.
- Table 5 (Bottom Center):** A table with two columns: 'Schlüssel' (Key) and 'Wert' (Value). It lists attributes for a building, such as 'building: yes' and 'type: multipolygon'.
- Table 6 (Bottom Right):** A table with two columns: 'Schlüssel' (Key) and 'Wert' (Value). It lists attributes for an address, such as 'addr:city: Wien', 'addr:country: AT', and 'addr:street: Schottenfeldgasse'.
- Table 7 (Bottom Right):** A table with two columns: 'Schlüssel' (Key) and 'Wert' (Value). It lists attributes for a tree, such as 'natural: tree'.

The map background shows a street grid with various features like buildings, trees, and bus routes. A scale bar at the top left indicates a distance of 211 meters. The map is overlaid with a grid of yellow squares, likely representing the OSM data tiles.

MapCSS

- Sprache unabhängig von Datenbank
- Verschiedene Libraries, beginnend mit „Halcyon“ für den Editor Potlatch2 (aus dem Jahr 2009)
- Syntax angelehnt an CSS für HTML; Selektoren beziehen sich aber auf geographische Objekte
- Info & Spezifikation:
<http://wiki.openstreetmap.org/wiki/MapCSS>

MapCSS - Example

```
canvas {
  fill-color: grey;
}
line|z12-[highway=primary],
line|z13-[highway=secondary],
line|z14-[highway=tertiary] {
  color: #000000;
  width: 6;
  casing-width: 1;
  casing-color: #a0a0a0;
  text: eval(tag("ref") . " - " .
    tag("name"));
}
area|z14-[leisure=park] {
  fill-color: green;
}
node|z10-13[place=city][population>150000] {
  text: name;
  font-size: 10
}
node|z14-16[place=city][population>150000] {
  text: name;
  font-size: 12
}
```



<http://pgmapcss.openstreetbrowser.org/?style=0ceb7&zoom=14&lat=48.2078&lon=16.3700>

MapCSS - Selektoren

- Objekttyp:
node, point, way, line, area, relation, canvas
- Zoom-Level: |z15, |z15-, |z-15, |z13-15
- Tag-Vergleiche (mehrfach):
[highway] // hat tag 'highway'
[highway=residential] // tag 'highway' hat wert 'residential'
[highway!=footway] // hat keinen tag 'highway' oder tag 'highway' hat nicht wert 'footway'
- Pseudo Klassen: :active, :hover, ...
- „Layer“ bzw. „Pseudo Element“ - mehrere Symbolizer für gleiches Objekt:
line[highway]::layer1
line[railway]::layer2

MapCSS – some Features

- alle Wege die Mitglied einer Relation sind:

```
relation[type=route] way { ... }  
relation[type=route] > way { ... }
```
- setze tag „foo“ auf den angegebenen Wert:

```
{ set tag = value; }
```
- Rechne metrische Angabe in Pixel um:

```
{ width: eval(metric("30m")); }
```


CSS Alternativen

- Cascadenik
- CartoCSS

beide spezifisch für Mapnik; in einem eigenen File müssen Layer und Datenbankqueries definiert werden, damit nicht Plattform-übergreifend.

Overpass Turbo

- Web-Applikation mit internem MapCSS Rendering
- Datenquelle: Overpass API (via HTTP)
- URL: <http://overpass-turbo.eu/>

Overpass Turbo

```
1 <osm-script output="json" timeout="25">
2   <!-- query part for: "natural=tree" -->
3   <query type="node">
4     <has-kv k="natural" v="tree"/>
5     <bbox-query {{bbox}}/>
6   </query>
7   <print mode="body"/>
8   <recurse type="down"/>
9   <print mode="skeleton" order="quadtile"/>
10 </osm-script>
11 {{style:
12
13  /* Hier: MapCSS */
14  node[natural=tree][type=conifer] {
15    color: #ff0000;
16  }
17  node[natural=tree][type=broad_leaved] {
18    color: #7f0000;
19  }
20 }
21 }}
```

Node 2044185028

Tags:

- circumference=0.32
- diameter_crown=2
- height=5
- natural=tree
- source=OGD Vienna
- species=Tilia cordata
- species:de=Stadtlinde
- start_date=2006
- taxon=Tilia cordata Greenspire
- taxon:cultivar=Greenspire
- tree.ref=3031
- type=broad_leaved

Coordinates:

48.2034542 / 16.3650395 (lat/lon)

geladen - Nodes: 248, Ways: 0, Relations: 0
angezeigt - POIs: 248, Linien: 0, Polygone: 0

<http://overpass-turbo.eu/s/3f8>

Overpass Turbo

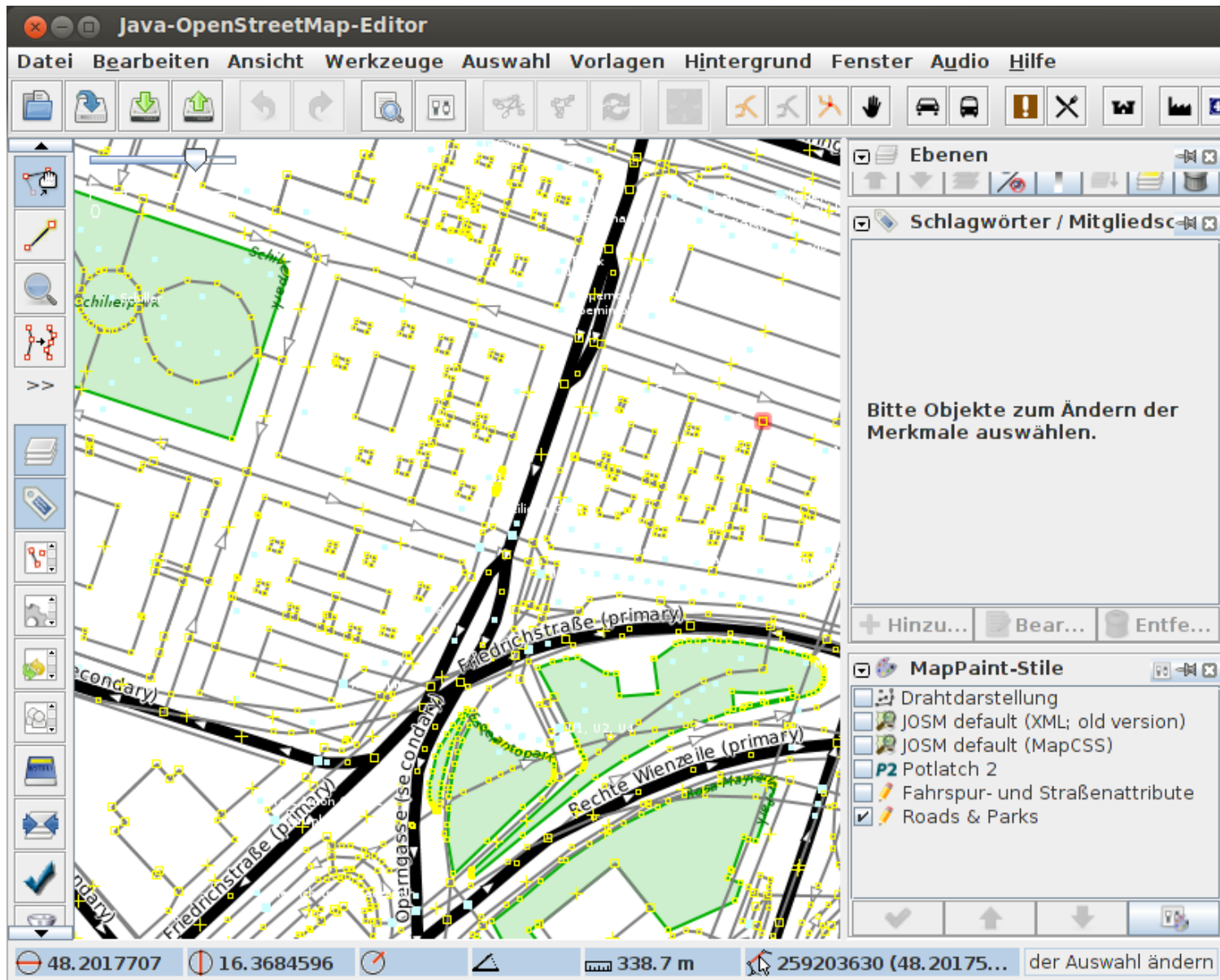
Fazit:

- Overpass Turbo praktisch um schnell Daten zu erfragen und in andere Formate umzuwandeln
- MapCSS Support stark eingeschränkt (keine Texte, kein Casing)
- Eigenheiten, zB. Nodes werden als kreisrundes Polygon behandelt
- Man braucht Query + Stylesheet

JOSM

- Editor für OpenStreetMap; Desktop Applikation in Java
- Datenquelle: OSM API oder (offline) OSM file
- URL: <http://josm.openstreetmap.de/>
- Verwaltung unter Ansicht → MapPaint-Stile

JOSM



JOSM

Fazit:

- MapCSS Support nicht standardkonform, dafür aber sehr mächtig
- Große Anzahl bereits vorhandener Stile, einfaches Hinzufügen eigener Stile
- Wireframe muss explizit ausgeschaltet werden:

```
canvas {  
  default-points: false;  
  default-lines: false;  
}
```

Mapnik + pgmapcss

- Mapnik ist ein in C++ geschriebener Render. Nicht interaktiv; Mapnik rendert Images (JPG, PNG, SVG, PDF) entweder von Command Line oder über andere Programme (zB. Apache + mod_tile, node_mapnik, Mapnik Viewer, ...)
- pgmapcss übersetzt MapCSS in ein Mapnik Stylesheet und eine Datenbankfunktion
- Datenquelle: PostgreSQL/Postgis mit osm2pgsql importiert
- URL: <https://github.com/plepe/pgmapcss>
Demo: <http://pgmapcss.openstreetbrowser.org/>

Mapnik + pgmapcss

pgmapcss demo page - Mozilla Firefox

pgmapcss demo page

pgmapcss.openstreetbrowser.org/?style=86f2f&zoom=15&lat=48.20

pgmapcss Edit Load Export

```
/* draw a green polygon for all parks */
area[leisure=park] {
  fill-color: #00af00;
}

/* print a label on all parks */
text: eval(tag(name));
font-family: "DejaVu Sans";
font-style: italic;
font-size: 9;
text-color: #005f00;
}

/* all major roads will be rendered with a 2px
black line */
line|z12-[highway=primary],
line|z12-[highway=secondary],
line|z12-[highway=tertiary] {
  color: #000000;
  width: 2;
}

/* and a black label next to the line
with a 50% transparent white halo */
text: eval(tag(name));
text-offset: 9;
text-color: #000000;
text-halo-color: #ffffff7f;
text-halo-radius: 1;
text-position: line;
}
```

z15 Lat: 48.2021 Lon: 16.3708 Style: 86f2f

pgmapcss-demo | © OpenStreetMap contributors

[Update map](#) Docu: [MapCSS](#), [Properties](#)

<http://pgmapcss.openstreetbrowser.org/?style=86f2f&base=none&zoom=15&lat=48.2021&lon=16.3708>

Mapnik + pgmapcss

Fazit:

- Nicht interaktiv
- MapCSS ist leichter zu schreiben als Mapnik XML
- Große Zahl an MapCSS Funktionen unterstützt + zusätzliche Funktionen (zB geometrische Funktionen, kombinieren von Objekten, ...)
- Problem: Wenn pgmapcss mit Mapnik 2.x verwendet wird, müssen viele Werte zur „Compile Time“ vorhersagbar sein
- Der Entwickler steht vor Ihnen :-)

Beispiele

- Stile, die möglichst standardkonform sind.
- <https://github.com/plepe/mapcss-examples>
Use it! Fork it!

Beispiel: Schwarzplan

```
canvas {  
  fill-color: white; /* MapCSS 0.2, pgmapcss */  
  background-color: white; /* JOSM */  
}  
area[building] {  
  fill-color: black;  
  fill-opacity: 1; /*JOSM,Overp.*  
  width: 0; /* Overpass */  
}
```

Funktioniert mit:
JOSM, pgmapcss,
Overpass Turbo (teilweise)

<http://overpass-turbo.eu/s/3iZ>

<http://pgmapcss.openstreetbrowser.org/?style=92c25&zoom=15&lat=48.2034&lon=16.3612>

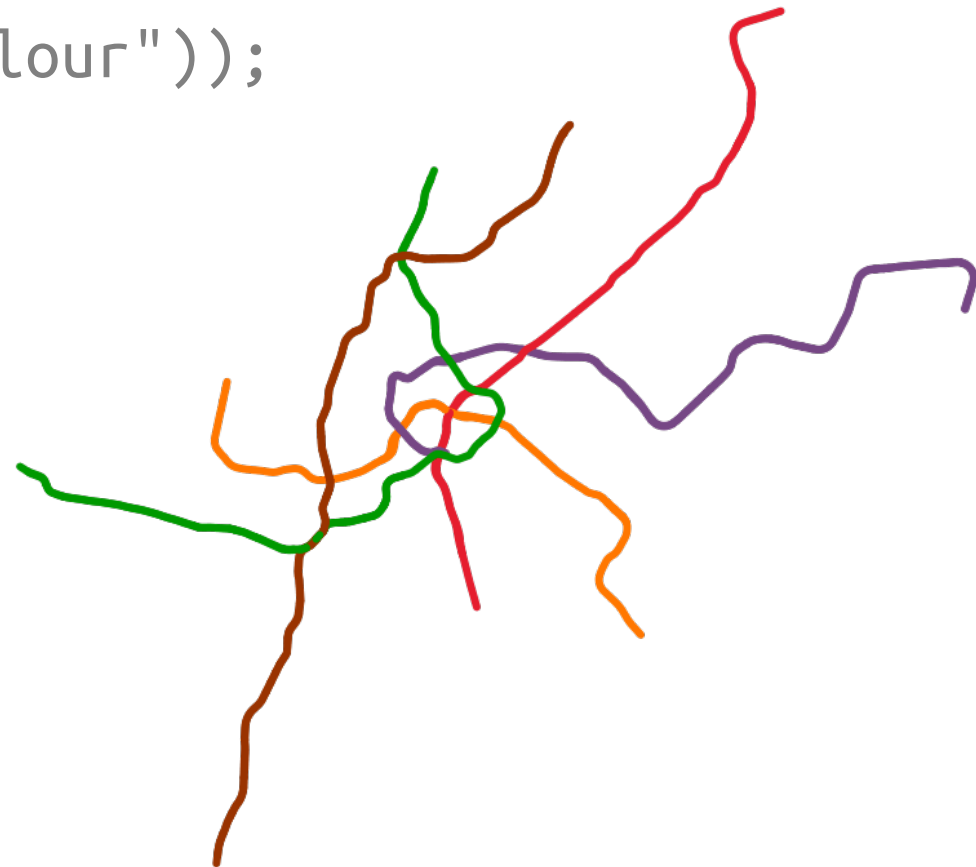


Beispiel: Wiener U-Bahn

```
/* Color all members of the subway relations in the  
tag "colour" of the relation */
```

```
relation[route=subway] > way[railway=subway] {  
  color: eval(parent_tag("colour"));  
  width: 6;  
  linecap: round;  
}
```

Funktioniert mit:
JOSM, pgmapcss (teilweise)

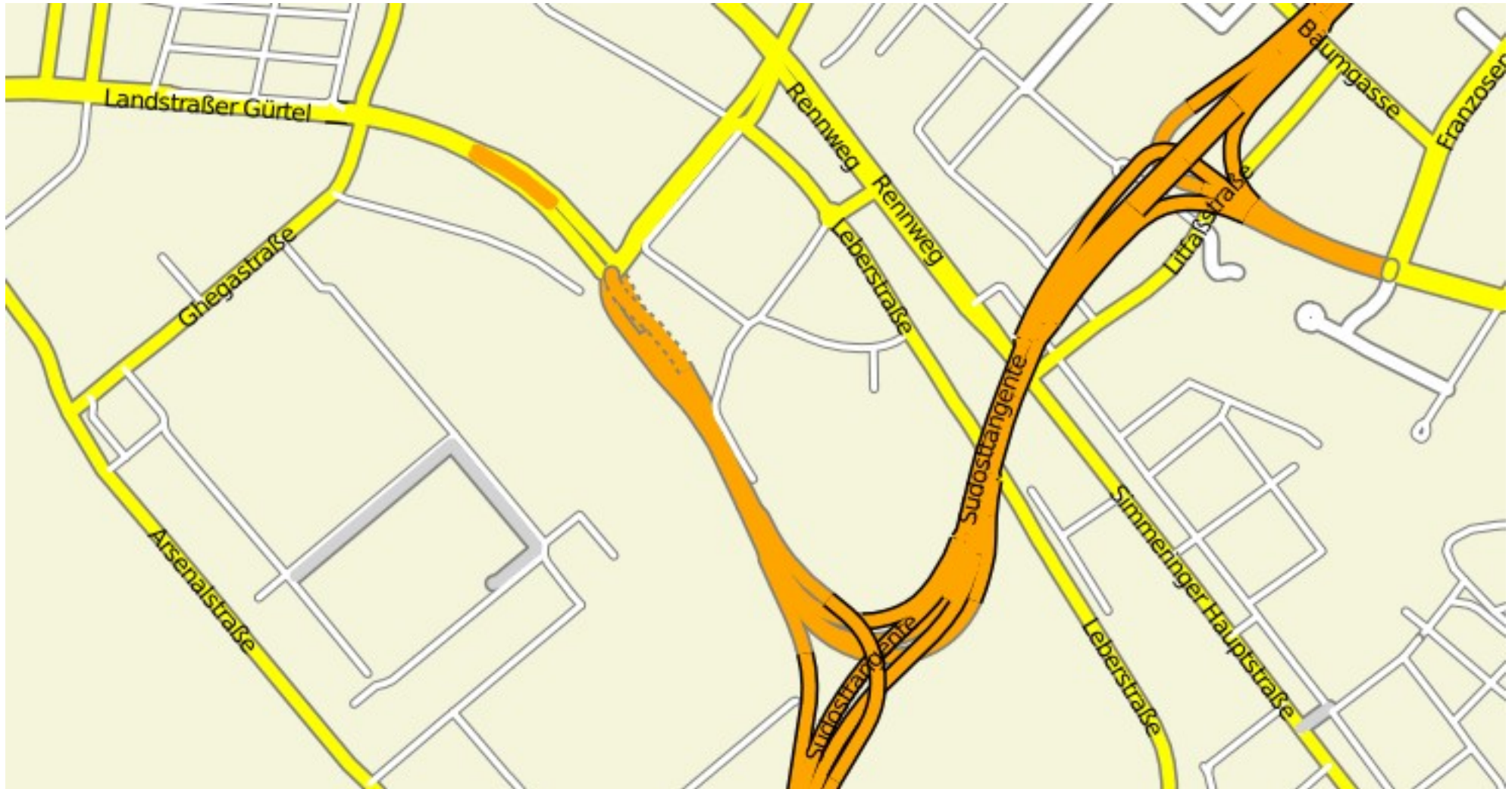


Beispiel: Straßennetz

```
canvas {
  fill-color: beige;
  background-color: beige;
}
way[highway] {
  linecap: round;
  casing-color: grey;
  text-color: black;
  font-size: 12;
}
way[highway=motorway],
way[highway=trunk],
way[highway=motorway_link],
way[highway=trunk_link] {
  color: orange;
}
way[highway=primary],
way[highway=secondary],
way[highway=tertiary] {
  color: yellow;
}
way[highway=unclassified],
way[highway=residential] {
  color: white;
}
way[highway=pedestrian],
way[highway=living_street] {
  color: lightgrey;
}
way|z8-9[highway=motorway],
way|z8-9[highway=trunk],
way|z9-10[highway=motorway_link],
way|z9-10[highway=trunk_link],
way|z10-11[highway=primary],
way|z11-12[highway=secondary],
way|z12-13[highway=tertiary],
way|z13-14[highway=unclassified],
way|z13-14[highway=pedestrian],
way|z14-15[highway=residential],
way|z14-15[highway=living_street] {
  width: 3;
  casing-width: 1;
}
way|z10-[highway=motorway],
way|z10-[highway=trunk],
way|z11-[highway=motorway_link],
way|z11-[highway=trunk_link],
way|z12-[highway=primary],
way|z13-[highway=secondary],
way|z14-[highway=tertiary],
way|z15-[highway=unclassified],
way|z15-[highway=pedestrian],
way|z16-[highway=residential],
way|z16-[highway=living_street] {
  width: 6;
  casing-width: 1.5;
  text: name;
  text-position: line;
}
way[bridge] {
  casing-color: black;
  linecap: none;
}
way[tunnel] {
  casing-dashes: 3,3;
  linecap: none;
}
```

- Funktioniert mit: JOSM, pgmapcss, Overpass (teilweise)

Beispiel: Straßennetz



<http://pgmapcss.openstreetbrowser.org/?style=5aed4&zoom=15&lat=48.1836&lon=16.4015>
<http://overpass-turbo.eu/s/3j0>

Danke für die Aufmerksamkeit!

<http://plepe.at/185>