TransitMapper Documentation
Draft

Patrick Brosi

19th March 2017

1 Basic Usage

The input file is a graph given as a (simplified) Geo-JSON file, consisting of nodes (represented as “Point”-features) and edges (represented as “LineString”-features). Each edge has a collection of unique “lines” that travel through it. The transitmapper will render these lines in a way that resembles a transit map. The input is read from `stdin`.

```
$ transitmapper -o test.svg < test.json
```

See below for an example input.

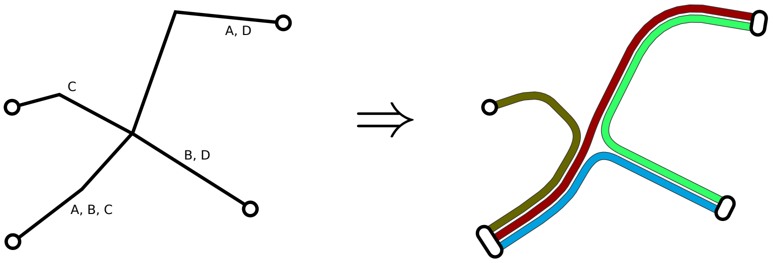

Figure 1: Simple example output

2 Command line parameters

The following command line parameters are accepted by `transitmapper` (see also `--help`).

`--line-width=N` The default width of a line, in output units. 20 by default.

`--line-spacing=N` The default spacing between lines, in output units. 10 by default.

`--render-station-names` Output the station names (experimental).

`--render-node-fronts` Output node fronts, useful for debugging.

`--resolution=D` Output resolution. 0.1 by default.

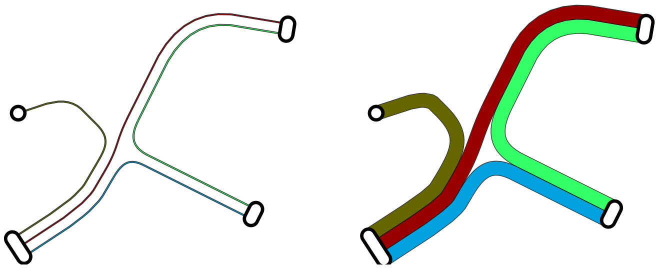

Figure 2: Different settings of `--line-width` and `--line-spacing`

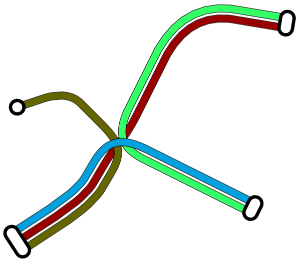

Figure 3: Output without ordering optimization (`-N`)

`--no-optim` (`-N`) Disable line-ordering optimization.

`--input-smoothing=D` Level of input-data smoothing. 3 by default.

3 JSON Format

See also ??. The input format is a lightweight subset of GeoJSON. At the top level, the input JSON must contain a `FeatureCollection` object:

```
{
  "type": "FeatureCollection",
  "features": [...]
}
```

A `feature` can either be a node (a `point`) or an edge (a `LineString`).

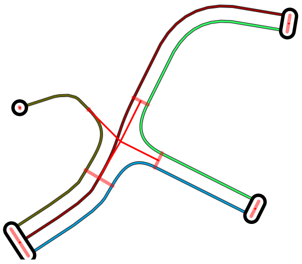

Figure 4: Node-front rendering (`--render-node-fronts`)

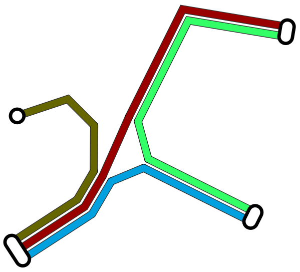

Figure 5: Without any line smoothing (`--input-smoothing=0 --bezier-prec=0`)

3.1 Nodes

A node consists of a geometry (the node's coordinates) and some properties.

```
{
  "geometry": {
        "coordinates": [0, 0],
        "type": "Point"
  },
  "properties": {
        "id": "1",
        "station_id": "1"
  }
}
```

`Type` is always `"point"`. The `coordinates` are given as an `[x,y]` array.

3.1.1 Node Properties

`id` A dataset-unique string identifier for this node. Will be referenced later in edges.

`station_id` If this node is a station, the station's unique id (optional).

`station_label` If this node is a station, the station's name (optional). Either `station_id` or `station_label` has to be set for a node to become a station.

`excluded_line_connections` A list of lines that are not connected in this node, even though two or more edges containing the line start/end in the node. The `line_id` (see ??) as well as the two edges this line is not connected in has to be given. Both edges are identified by the adjacent node. Example:

```
"excluded_line_connections": [
  {
        "edge1_node": "2",
        "edge2_node": "3",
        "route": "A"
  },
  {
        "edge1_node": "4",
        "edge2_node": "6",
        "route": "B"
  }
]
```

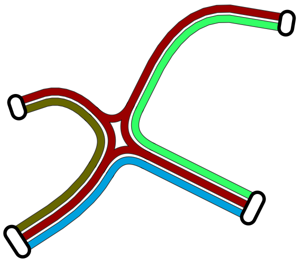

Figure 6: The red line is not connected between the two main axes (via `exluded_line_connections`)

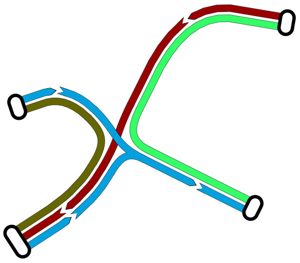

Figure 7: Directed line occurances (via `direction`)

3.2 Edges

An edge consists also of a geometry (a linestring) and some properties.

```
{
  "geometry": {
        "coordinates": [
          [0, 0], [500, 900], [1000, 950]
        ],
        "type": "LineString"
  },
  "properties": {
        "from": "1",
        "to": "2",
        "lines": [
          {"color": "00a1de"},
          {"color": "990000"}
        ]
  }
}
```

Properties `from` and `to` hold the IDs of the nodes this edges connects.

`id` A dataset-unique string identifier for this node. Will be referenced later in edges.

4 Example Input

```
{
  "type": "FeatureCollection",
  "features": [
```

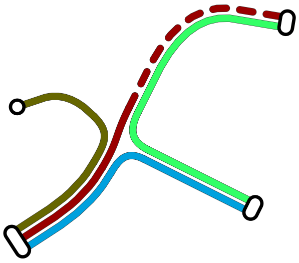

Figure 8: Dashed line (via `"dash-array"="4 2"`)

```
    {
      "geometry": {
        "coordinates": [0, 0],
        "type": "Point"
      },
      "properties": {
        "id": "1",
        "station_id": "1"
      }
    },
    {
      "geometry": {
        "coordinates": [1000, 1000],
        "type": "Point"
      },
      "properties": {
        "id": "2",
        "station_id": "2"
      },
      "type": "Feature"
    },
    {
      "geometry": {
        "coordinates": [
          [0, 0], [500, 900], [1000, 950]
        ],
        "type": "LineString"
      },
      "properties": {
        "from": "1",
        "to": "2",
        "lines": [
          {"color": "00a1de"},
          {"color": "990000"}
        ]
      }
    }
  ]
}
```