1 The algorithms

1.1 Network Algorithm

1.1.1 Background definitions

- $\mathbb{R}^2$ is the two dimensional Euclidian space.
- P is the given set of points in $\mathbb{R}^2$
- L_1 distance is the distance between two points in taxicab geometry.
- Rectilinear minimum spanning tree $RMST$is the tree spanning all points in P such that the sum of its edge L_1 distances is the minimum.
- RN_{min} is a minimum reconstruction of the road network that adds missing straight road segments to the $RMST$.
- RN_{com} is a complete reconstruction of the road network in which juctions and roundabouts are connected in a correct way.

1.1.2 Outline

The algorithm consists of the following steps:

1. Given a set V of points in $\mathbb{R}^2$ create a graph $G(V, E)$ where $V = P$ and $E = V \times V$.
2. Compute the $RMST$ of G.
3. Mark points that are only in one segment and find RN_{min}.
4. Find segments with a divergent angle and compute RN_{com}.

1.1.3 Description

Given the set V of points in $\mathbb{R}^2$ we first generate a graph G in which every point $p \in V$ is connected to every point $q \in V$. Then the algorithms computes the $RMST$, see algorithm **??**, using kruskal's algorithm [**?**], for G resulting in a connected graph that gives a good approximation of the road network but some straight road segments may not be connected, see figure **??**.

We now find a set of points S for which every $s \in S$ is a endpoint of a road segment, see algorithm **??**. For these points a edge $e \in E$ is found such that $e = (s, x)$ where $x \in V$. The slope of the edge determines the direction for the road segment for which s is the endpoint. A line l_1 perpendicular to the slope is calculated and it is checked whether a nearby point $x \in V$ is below l_1 for e with a negative slope or if x is above l_1 for e with a positive slope, see figure **??**.

Finally we calculate RN_{com}, see algorithm **??**. First we find all segments $e \in E$ for which the slope of the following segment $e' \in E$ is divergent. For the segment e we draw a line in the same direction and look for a intersection with a segment $x \in E$, see figure **??**. If the distance between e and x is sufficiently small segment e' is removed from RN_{min} and a new segment from e to x is added to RN_{com}.

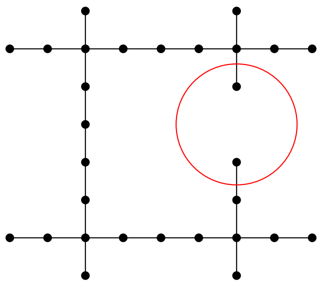

Figure 1: $RMST$ for P with a missing road segment indicated by the red circle.

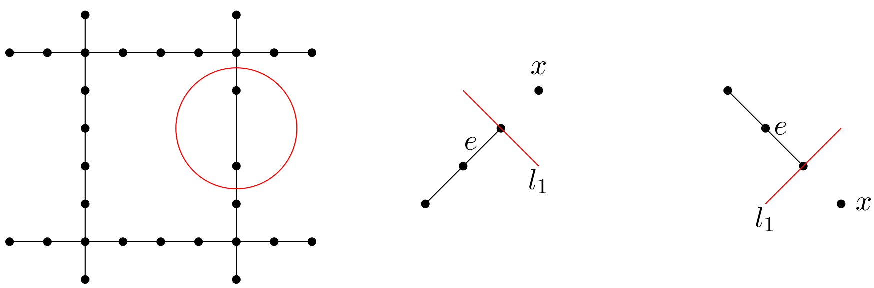

Figure 2: Left: RN_{min} road segment in circle is connected. Middle: Slope of e is positive, check if x is above l_1. Right: Slope of e is negative, check if x is below l_1.

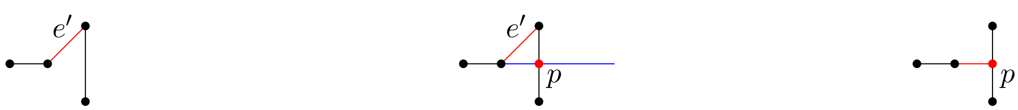

Figure 3: Calculation steps for RN_{com}. Left: Edge e' with divergent angle in red. Middle: Blue line intersects at p. Right: e' is remove from RN_{min} and new edge s is added to RN_{com}

Algorithm 1 Calculate $RMST$ for graph $G(V, E)$

  $A = \emptyset$
  Sort by increasing L_1 distance(E)
  for all $v \in V$ **do**
    MAKE-SET(v)
  end for
  for all $u, v \in E$ **do**
    if FIND-SET(u) $\neq$ FIND-SET(v) **then**
      $A = A \cup \{(u, v)\}$
      UNION(u, v)
    end if
  end for
  return A

Algorithm 2 Calculate RN_{min} for $RMST(V, E)$

```
A = RMST
for all v ∈ V do
  if v is a endpoint and adjacentnodes(v) ≠ ∅ then
    e = FIND-SEGMENT(V)
    repeat
      if adjacentnodes[i].direction = e.direction then
        A.add(new Segment(e,adjacentnodes[i]))
      end if
      i++
    until new segment is added
  end if
end for
return A
```

Algorithm 3 Calculate RN_{com} given RN_{min})

```
A = RN_min
for all e ∈ E do
  e′ = e.NEXT-SEGMENT
  if e′.getSlope ≠ e.getSlope then
    if Direction e intersect with segment x ∈ E then
      if Distance between e and x < 0.2f then
        p = new Point at intersection with x
        A = A − e′
        s= new Segment from e to p
        A = A + s
      end if
    end if
  end if
end for
return A
```

References

[1] J.B. Kruskal. On the shortest spanning subtree of a graph and the traveling salesman problem. In *Proceedings of the American Mathematical Society*,7: 48-50, 1956.

[2] D. Chen, L.J. Guibas, J. Hershberger, J. Sun. Road Network Reconstruction for Organizing Paths. In *Proceedings of 21st ACM-SIAM Symposium on Discrete Algorithms*, 10: 1309-1320, 2010.