Robart GmbH
Friedhofstraße 4, 4020 Linz, Austria
Tel.: +43 732 944 440-00
office@robart.cc, www.robart.cc

Efficient Linesegment Algorithms

Assume a set of 20 to 100 linesegments, denoted by s_i defined in Cartesian coordinates via their start and end points as

$$\boldsymbol{s}_i = \begin{pmatrix} x_{s,i} \\ x_{e,i} \\ y_{s,i} \\ y_{e,i} \end{pmatrix} \quad (1)$$

where, e.g., $x_{e,I}$ denotes the *x* coordinate of the end point of the *i*th line segment. These line segments change with every measurement update, but typically only slightly. However, new line segments might be added, line segments might change in length, or be deleted. We can use this information to make potential pre-processing steps more efficient.
In the following we denote as the distance of a point $x=(x,y)^T$ to a line segment s_i as the minimum distance in the typical two norm sense, i.e.,

$$d_i^2(\boldsymbol{x}) = \underset{\boldsymbol{x}_s \in \boldsymbol{s}_i}{argmin} \|\boldsymbol{x} - \boldsymbol{x}_s\|^2 \quad (2)$$

Furthermore we define as the distance between two line segments s_i , s_j the distance between one point taken on s_i and another taken on s_j such that the distance between these points is minimal, i.e.,

$$d_{i,j}^2 = \underset{\boldsymbol{x}_i \in \boldsymbol{s}_i, \boldsymbol{x}_j \in \boldsymbol{s}_j}{argmin} \|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2 \quad (3)$$

It is worth noting that if the two segments do not intersect, and are not parallel, that the closest distance will always be taken from a start or end point of at least one of the segments.

Given the line segments we are looking for efficient algorithms to calculate the following:

- Given a point $x=(x,y)^T$ determine the closest line segment from the set of given line segments.
- Given a point x determine the set of all line segments that are in a distance between d_{lower} and d_{upper} .
- Determine relative measures between line segments, such as
    - which pairs of line segments s_i , s_j come closer than a distance d_{lower} .