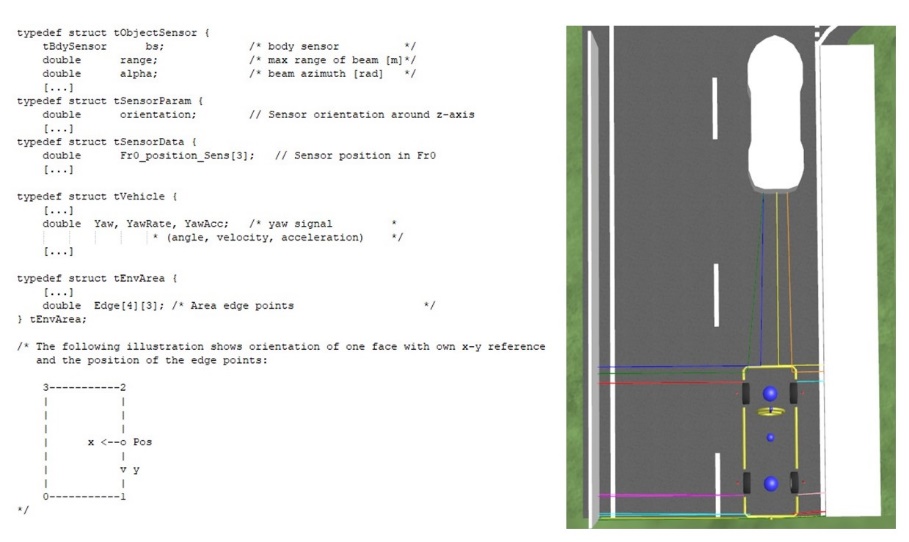
**NPs outside of CM**

The calculation of Nearest Points is based on CarMaker's Traffic Objects (TO, defined by adjacent vertical rectangles), sensors and ego vehicle information.



**Figure 1.** CarMaker information and example NPs indicated as straight lines from sensors to TOs

**Normal to TO**

The Nearest point is calculated as the point of intersection between i) the normal from the sensor position to the plane defined by the TO surface (or tile, outlined by area edge points, Fig. 1) and ii) the TO surface plane. (see *Occlusion documentation* for plane-line intersection).

**Distance to TO edge**

If the intersection point falls outside the TO tile (Fig. 2, yellow circle; see *Occlusion documentation* for the “*point in polygon”* problem), the NP is taken as the projection of the sensor position onto the closest edge of the tile (Fig. 2, green circle, sensor position 1). In case the projection of the sensor falls outside the tile edge, the NP is taken as the closest corner of the tile (Fig. 2, orange circle, sensor position 2).

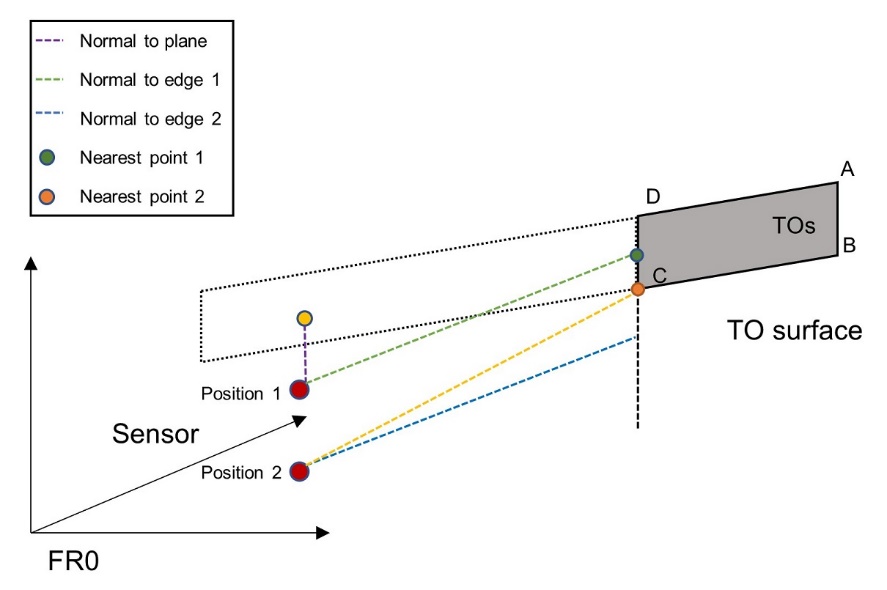
The projection of the sensor onto the tile edge (i.e., CD) is calculated as:

CS = S - C; //Vector from C to point

CD = D - C; // Vector from C to D

S\_proj = C + dotProd(CS, CD) / dotProd(CD, CD) \* CD;

Where C, D, S are arrays containing the coordinates of the sensor and tile corners in 3D (Fig. 2), and CS and CD are vectors from corner C to the sensor and from C to D, respectively.

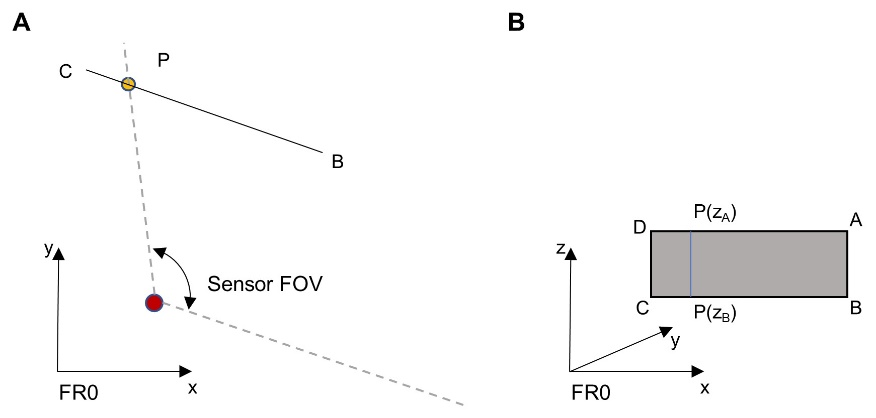


**Figure 2.** Representation of Nearest Point position. Position 1: normal to TO plane (purple line) falls outside TO surface; Position 2: sensor projection onto TO tile edge (blue line) falls outside edge segment.

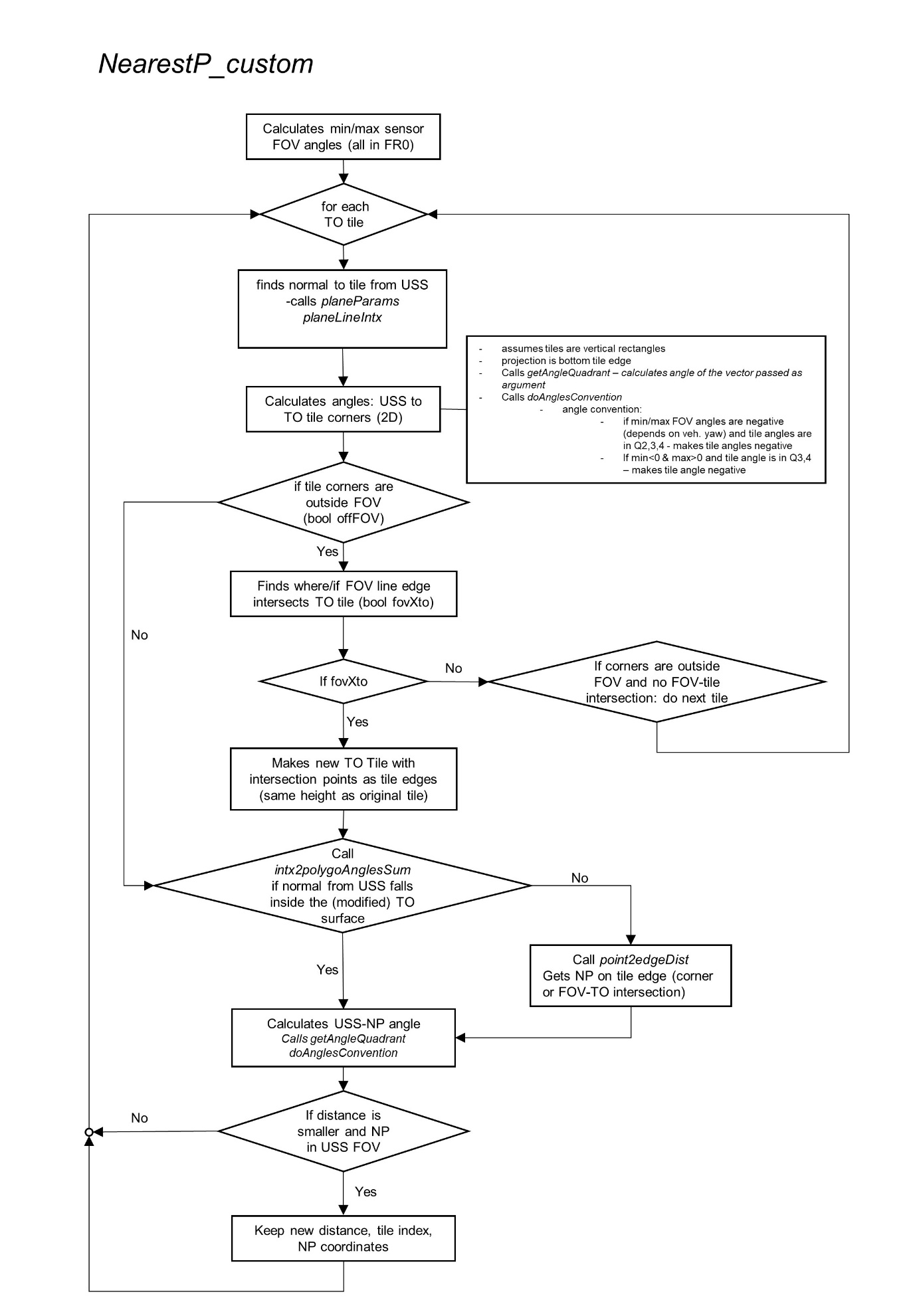
**Sensor FOV**

The NP position is calculated for TO tiles situated either completely or partly in the sensor’s field of view (FOV).

The angles of the sensor FOV edges (i.e., azimuth limits) and the intersection points of FOV edges with TO are determined in 2D (assuming that each TO is formed by adjacent vertical rectangles - tiles). After determining the intersection points between FOV edges and the TO tile projection (Fig. 3A), the tile is re-constituted in 3D with intersection points take as new tile edges (Fig. 3B). The NP is determined as described above, using the new tile corners.



**Figure 3.** TO tile situated partly outside the sensor’s FOV. **A** An intersection point between FOV edge and TO tile is determined in 2D. **B** An in-FOV-tile is reconstructed in 3D using the 2D coordinates of the intersection point.

**Diagram of Nearest Point C++ implementation**