6.3 Space Discretization - Avoidance Grid

Operation Space: The *Operation Space* is a space where UAS can effectively surveillance its surroundings.

The*Discrete Situation Evaluation* is bounded to*UAS* specific*position* and*orientation* in fixed*time* t_i. To enable*deterministic evaluation* the*operation space* needs to be segmented into*finite set* of space portions. The*finite operation space segmentation* is usually done by*Grid* segmentation, which distributes space into portions with solid boundary.

The *Main Sensor* is *LiDAR* (problems **??**.-**??**.). The *effective occupancy computation* [1] is given by clustering *LiDAR* field of vision into *polar coordinates grid*.

The *point* scanned by *LiDAR*, where *UAS position* is center of *local coordinate frame* and *UAS heading is defining the main axes* is given as:

$$point = [distance, horizontal^\circ, vertical^\circ].$$

Note. For polar/euclidean transformations and local/global coordinate frames refer to background theory (sec. **??**).

The *right side* of UAS *horizontal*° $]-\pi, 0[$, the *left side* of UAS $horizontal^\circ \in [0, \pi]$, the *down side* of UAS *vertical*° $]-\pi, 0[$, the *top side* of UAS $vertical^\circ \in [0, \pi]$

LiDAR Reading Space Segmentation: The *polar space* can be separated into cells, which bounds the portion the space, similar to *euclidean* space grid. The *reason* for this segmentation is *LiDAR* reading density[1]. The *polar space portions* state can be assessed directly, the *polar* $\rightarrow$ *euclidean* coordinate frame transformation is not time-effective. The *polar space* assessment of *Lidar Data* has minimal complexity and it is cost effective. [2].

Definition 1. *Space partition - cell The* cell *is a portion of space in UAS local polar coordinate frame, given by:*

1. Distance Range - *bounded by* $distance_{start} < distance_{end}$ *in* $\mathbb{R}^+$.
2. Horizontal Range - *bounded by* $horizontal^\circ_{start} < horizontal^\circ_{start} \in]-\pi, \pi]$.
3. Vertical Range - *bounded by* $vertical^\circ_{start} < vertical^\circ_{start} \in]-\pi, \pi]$.

The bounded space *for* cell *is defined as:*

[1] Example rotary LiDAR Velodyne VL-16 specs: `https://www.cadden.fr/wp-content/uploads/2017/02/Velodyne_VLP-16-Puck.pdf`

$$
\begin{aligned}
&BoundedSpace(cell) = \dots \\
&\qquad \left\{ \begin{array}{l} point \in \mathbb{R}^3 \;\; where: \\ \qquad \left(\begin{array}{rcr} cell.distance_{start} < & point.distance \leq & cell.distance_{end}, \\ cell.horizontal^{\circ}_{start} < & point.horizontal^{\circ} \leq & cell.horizontal^{\circ}_{end}, \\ cell.vertical^{\circ}_{start} < & point.vertical^{\circ} \leq & cell.vertical^{\circ}_{end} \end{array} \right) \end{array} \right\}
\end{aligned} \tag{6.1}
$$

For one LiDAR Scan *the* hits set *is given as* set *of* all points *which lands in bounded cell space:*

$$
LidarHits(cell) = \{point \in LidarScan : point \in BoundedSpace(cell)\} \tag{6.2}
$$

The passing hits *for cell are hits which are going trough the cell (passing), but it lands in distance greater than* $cell.distance_{end}$*, defined as:*

$$
\begin{aligned}
&PassingHits(cell) = \dots \\
&\qquad \left\{ \begin{array}{l} point \in LidarScan \;\; where: \\ \qquad \left(\begin{array}{rcr} cell.distance_{end} < & point.distance & , \\ cell.horizontal^{\circ}_{start} < & point.horizontal^{\circ} \leq & cell.horizontal^{\circ}_{end}, \\ cell.vertical^{\circ}_{start} < & point.vertical^{\circ} \leq & cell.vertical^{\circ}_{end} \end{array} \right) \end{array} \right\}
\end{aligned} \tag{6.3}
$$

Note. The *cells* with same *distance* range form *layers.* The *greater* the distance from *coordinate frame* origin the *greater* volume of the cell.

Effective Operation Space - Avoidance Grid: Let start with example, the UAS (fig. 6.1).

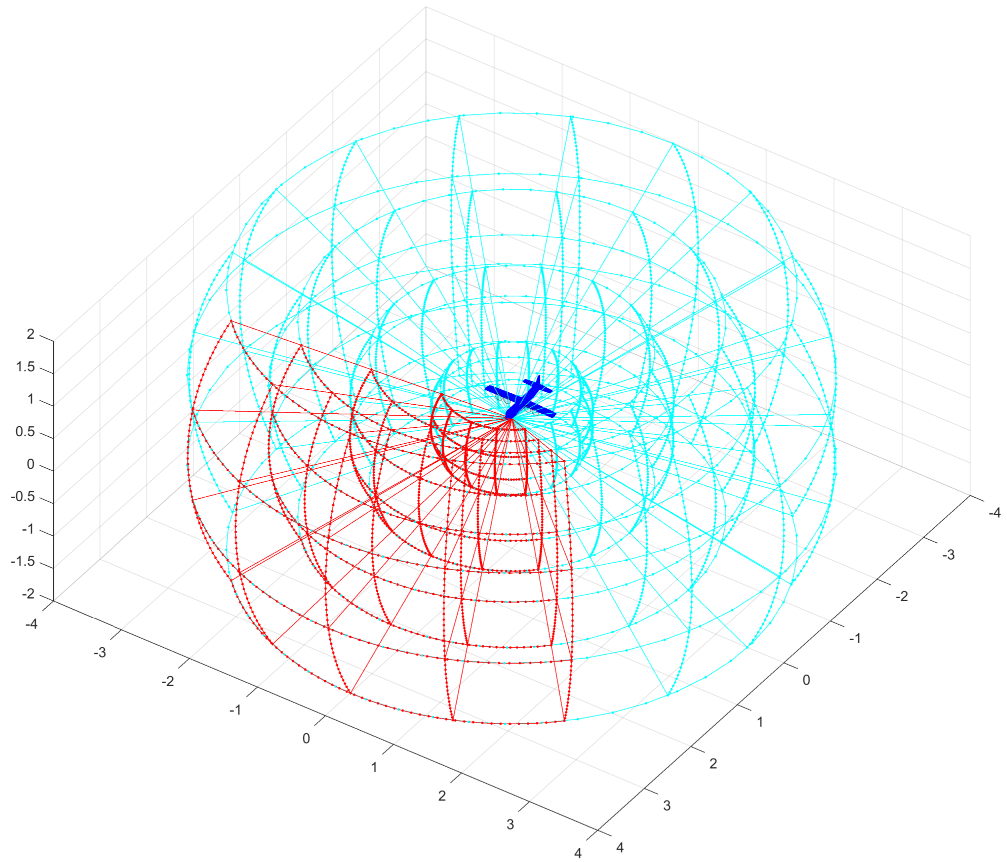

Figure 6.1: Example: The *LiDAR* reading segmentation - cells.

The *full LiDAR Swipe* (cyan and red lines) of *UAS* (blue plane) has *shape* of conical cylinder. Under *ideal circumstances* the *LiDAR swipe* would have *ball shape*, but in real cases the *craft body portion* where *LiDAR* is mounted is unused.

The *frontal portion* (red line) is a set of cells where *UAS* can make maneuver. The *red portion* size is determined by [3]:

1. *Sensors ranges* - the union of *effective sensor ranges* defines the maximal *effective space boundary*, because there is no reason to asses situation over effective *sensor range*.

2. *Information sources* impact - there is no real impact on *effective space boundary*.

3. *UAS maneuverability* - the *Reach Set* (sec. **??**) gives optimal *effective space boundary*, because there is no need to assess the situation out of *reachable space* or its vicinity.

4. Computation power - the *Reach Set Evaluation* and *Intersection* algorithms are *scaling* with *effective space boundary*.

5. Airworthiness requirements - the *regulations* can impose some minimal requirements on *effective space boundary*.

Definition 2. *Avoidance Grid The* effective space boundary *(fig. 6.1 red lines) given by a portion of space in UAS local polar coordinate frame, bounded by:*

1. Distance Range - *bounded by* $distance_{start} < distance_{end}$ *in* $\mathbb{R}^+$.
2. Horizontal Range - *bounded by* $horizontal^\circ_{start} < horizontal^\circ_{start} \in]-\pi, \pi]$.
3. Vertical Range - *bounded by* $vertical^\circ_{start} < vertical^\circ_{start} \in]-\pi, \pi]$.

Separated into layers *depending on the distance and* layer count*:*

$$\begin{aligned} layer^i_{start} &= (i-1) \times \frac{distance_{end} - distance_{start}}{layerCount} \\ layer^i_{end} &= i \times \frac{distance_{end} - distance_{start}}{layerCount} \end{aligned} \quad ; \quad i \in 1 \dots I \tag{6.4}$$

Layer horizontal/vertical separations separations defined by horizontal/vertically cell count:

$$\begin{aligned} horizontal^j_{start} &= (j-1) \times \frac{horizontal^\circ_{end} - horizontal^\circ_{start}}{horizontalCount} \\ horizontal^j_{end} &= j \times \frac{horizontal^\circ_{end} - horizontal^\circ_{start}}{horizontalCount} \end{aligned} \quad ; \quad j \in 1 \dots J \tag{6.5}$$

$$\begin{aligned} vertical^k_{start} &= (k-1) \times \frac{vertical^\circ_{end} - vertical^\circ_{start}}{verticalCount} \\ vertical^k_{end} &= k \times \frac{vertical^\circ_{end} - vertical^\circ_{start}}{verticalCount} \end{aligned} \quad ; \quad k \in 1 \dots K \tag{6.6}$$

Then $cell_{i,j,k}$ *given by (def. 1) is member cell of Avoidance Grid for boundaries:*

1. Cell Distance Range *(eq. 6.4) depending on layer index* i.
2. Cell Horizontal Range *(eq. 6.5) depending on horizontal index* j.
3. Cell Vertical Range *(eq. 6.6) depending on horizontal index* k.

The example of Avoidance Grid Cells *is given in (fig. 6.1 red boundary).*

The Avoidance Grid *is then given as set of cells:*

$$AvoidanceGrid = \bigcup cell_{i,j,k} \forall i \in 1 \dots I, j \in 1 \dots J, k \in 1 \dots K \tag{6.7}$$

Trajectory Intersection: The *trajectory* intersection with *Avoidance Grid* is solved in context of *Reach Set Approximation* (def. **??**).

Note. The *trajectory intersection* function does not have an impact on *Reach Set Approximation*, because its done prior the flight.

Grid Scaling: For *Sensor Field* there is *effective sensor boundary* given as set:

$$Boundary(Sensor \in SensorField) = \{points \in \mathbb{R}^3 : \text{where reliable}\} \tag{6.8}$$

The *Boundary* for sensor fields is then given as *union of all singe sensor boundaries*:

$$Boundary(SensorField) = \bigcap_{\forall Sensors} Boundary(Sensor \in SensorField) \tag{6.9}$$

Depending on boundary properties it can be projected into maximal avoidance grid boundary values:

$$Boundary(SensorField) \to AvoidanceGrid : \begin{matrix} \max(distanceRange) \\ \max(horizontalRange) \\ \max(verticalRange) \end{matrix} \tag{6.10}$$

Our approach taken worst LiDAR performance into account [4] and following parameters for avoidance grid were calculated:

1. distance range $[0m, 10m]$,
2. horizontal range $]-180^\circ, 180^\circ]$,
3. vertical range $[-30^\circ, 30^\circ]$.

The *count of layers* is derived from *average distance traveled by one movement application*:

$$layerCount = \frac{|distanceRange|}{\text{avg.} \quad length(movement \in MovementSet)} \tag{6.11}$$

The *layer length* is based on *our movement set* (tab. **??**, **??**) the average movement length is 1 m, therefore the *layer count* is 10.

The *efficient boundary* is given by *Reach Set*. Estimate reach set coverage space using *ellipsoidal toolbox* [5] up to given *sensor field* maximal distance:

$$Boundary(ReachSet) = Ellipsoid(UASSystem, distance) \tag{6.12}$$

The values for *Reach Set Boundary* with distance 10 m was following:

1. distance range $[0m, 10m]$,
2. horizontal range $[-45^\circ, 45^\circ]$,
3. vertical range $[-45^\circ, 45^\circ]$,

The *Avoidance Grid* boundary is given as *intersection* of all boundaries:

$$Boundary(AvoidanceGrid) = Boundary(ReachSet) \cap Boundary(SensorField) \quad (6.13)$$

The values for *Avoidance Grid Boundary* for our UAS system (sec. **??**) following:

1. distance range $[0m, 10m]$,
2. horizontal range $[-45°, 45°]$,
3. vertical range $[-45°, 45°]$,
4. layer count 10, layer distance 1m.

The *horizontal cell count* and *vertical cell count* was estimated by *rule of thumb* to have value 7 and 5.

Cell in Avoidance Grid Properties: For each cell $\vec{p} \in \mathbb{R}^3$ in the there are properties to be checked:

1. *Is there visibility to the cell ?* - how good is an observation of the cell by Sensor Field.
2. *Is there threat present ?* - how sure the data fusion is that there is eminent threat in the cell.
3. *Is the cell reachable ?* - if there is any trajectory which can get UAS to that cell without too much threat along the way.

The answers to these questions will be given later (tab. **??**).

Bibliography

[1] Florian Homm, Nico Kaempchen, Jeff Ota, and Darius Burschka. Efficient occupancy grid computation on the gpu with lidar and radar for road boundary detection. In *Intelligent Vehicles Symposium (IV), 2010 IEEE*, pages 1006–1013. IEEE, 2010.

[2] Sandeep Gupta, Holger Weinacker, and Barbara Koch. Comparative analysis of clustering-based approaches for 3-d single tree detection using airborne fullwave lidar data. *Remote Sensing*, 2(4):968–989, 2010.

[3] Osmar R Zaïane and Chi-Hoon Lee. Clustering spatial data when facing physical constraints. In *Data Mining, 2002. ICDM 2003. Proceedings. 2002 IEEE International Conference on*, pages 737–740. IEEE, 2002.

[4] Roberto Sabatini, Alessandro Gardi, and Mark A Richardson. Lidar obstacle warning and avoidance system for unmanned aircraft. *International Journal of Mechanical, Aerospace, Industrial and Mechatronics Engineering*, 8(4):702–713, 2014.

[5] Alex A Kurzhanskiy and Pravin Varaiya. Ellipsoidal toolbox (et). In *Decision and Control, 2006 45th IEEE Conference on*, pages 1498–1503. IEEE, 2006.