Foresight: Remote Sensing For Driverless Cars Using Small Unmanned Aerial Vehicles

Alex Wallar Javier Alonso-Mora Raphael Chang Daniela Rus

***Abstract*—**

Definition 1. *A laser scan is a sequence of points, $L \subset \mathbb{R}^2$, such that if a polygon, P, is constructed with these points, there exists a point, $c \in \mathbb{R}^2$, such that $\forall x \in L$, the line segment from c to x is entirely contained in P.*

Definition 2. *A break in the laser scan is a line segment (L_i, L_{i+1}) such that $||L_i - L_{i+1}||_2 > \delta$ where δ is the break threshold. The set of all these breaks will be referred to as $\tilde{L}$.*

Definition 3. *A blind spot, $B(L_i, L_{i+1})$, is the set of points contained between the line segments, (L_i, L_{i+1}) and $(L_i + \epsilon \cdot \hat{n}, L_{i+1} + \epsilon \cdot \hat{n})$ where $\hat{n}$ is the unit normal of the line segment (L_i, L_{i+1}), ϵ is a parameter governing the size of the region, and (L_i, L_{i+1}) is a break in the laser scan.*

Definition 4. *The blind region, $\mathcal{B}$, for a laser scan, L, is defined as $\mathcal{B} = \bigcup_{(i,j) \in \tilde{L}} B(i,j)$.*

Definition 5. *The sensor projection for our robot, $\psi(x, \theta)$ is defined as the set of points that the robot is able to observe from configuration x with yaw, θ. The optimal sensor projection for a given configuration and blind region is defined as $\psi^*(x, \mathcal{B}) = \psi(x, \theta^*(x, \mathcal{B}))$ with $\theta^*(x, \mathcal{B}) = \underset{0 < \theta \leq 2\pi}{\arg\max}\, \textsc{Area}(\mathcal{B} \cap \psi(x, \theta))$*

Definition 6. *A path, $\rho \subset V \times [0, 2\pi]$, is a sequence of tuples, (x, θ), consisting of state and yaw, such that $\forall i : (\rho_{i,x}, \rho_{i+1,x}) \in E$, where $G = (V, E)$ is a finite sampled graph within the polygon constructed using the laser scan, L, that represents the connectivity of the free space.*

Lemma 1. *If a path is returned from Algo. 1, it will have a total cost less than C.*

Proof. At each iteration, the cost to reach each neighbour of x is computed. This is added to the total cost of current path. If the cost of the path from x_0 to x' is larger than C, it is discarded from the search. Thus only path with cost less than C, can be returned from Algo. 1. □

Lemma 2. *Algo. 1 will finish in a finite number of steps if $\forall (i,j) \in E : \textsc{CostToGo}(i,j) \geq 1$ and $C < \infty$*

Proof. Since the cumulative cost being added to search is monotonically increasing, if a given path is not returned, its cost will eventually be greater than C in a finite amount of time since, $C < \infty$ and all individual costs must be greater than 1. If no path is returned by the algorithm, it means that all paths in G have not reached the optimality threshold with their costs being less than C. Since G is a finite graph, there are a finite amount of paths and therefore, to determine if no path will be returned takes a finite number of steps. If a path is returned, Algo. 1 is returning the first path that meets the optimality threshold which would occur before all paths in G are exhausted, thus returning in a finite number of steps. □

Algorithm 1

Input:

- x_0: The initial position of the robot
- $\mathcal{B}$: The blind region
- $G = (V, E)$: The finite sampled graph within the laser scan polygon
- γ, C: The optimality and cumulative cost thresholds respectively for the algorithm's termination

Output:

- $\rho \subset V \times [0, 2\pi]$: A sequence of tuples representing the path

$Q \leftarrow \{(x_0, \mathcal{B} \setminus \psi^*(x_0, \mathcal{B}), 0)\}$
while $|Q| > 0$ **do**
  $(x, \mathcal{B}', c) \leftarrow \underset{q \in Q}{\arg\min}\, \textsc{Area}(q_{\mathcal{B}'})$
  if $\textsc{Area}(\mathcal{B} \setminus \mathcal{B}') > \gamma \cdot \textsc{Area}(\mathcal{B})$ **then**
    $\rho \leftarrow \{\}$, $\hat{x} \leftarrow x'$, $\hat{\mathcal{B}} \leftarrow \mathcal{B}'$, $\hat{c} \leftarrow c$
    while $\textsc{HasParent}(\hat{x}, \hat{B}, \hat{c})$ **do**
      $\rho \leftarrow \{(\hat{x}, \theta^*(\hat{x}, \hat{\mathcal{B}}))\} \cup \rho$
      $(\hat{x}, \hat{\mathcal{B}}, \hat{c}) \leftarrow \textsc{Parent}(\hat{x}, \hat{\mathcal{B}}, \hat{c})$
    return ρ
  for all x' **where** $(x, x') \in E$ **do**
    $c' \leftarrow c + \textsc{CostToGo}(x, x')$
    if $c' < C$ **then**
      $Q \leftarrow Q \cup \{(x', \mathcal{B}' \setminus \psi^*(x', \mathcal{B}'), c')\}$
      $\textsc{Parent}(x', \mathcal{B}' \setminus \psi^*(x', \mathcal{B}'), c') \leftarrow (x, \mathcal{B}', c)$
  $Q \leftarrow Q \setminus (x, \mathcal{B}', c)$
return `false`

Lemma 3. *At each iteration, the residual blind region, $\mathcal{B}_n = \mathcal{B} \setminus \bigcup_{(x_i, \theta_i) \in \rho} \psi(x_i, \theta_i)$ where ρ is the path from x_0 to x_n.*

Proof. At the n^{th}, the blind region in the tuple being added to Q is

$$\mathcal{B}_n = \mathcal{B} \setminus \psi^*(x_0, \mathcal{B}) \setminus \psi^*(x_1, \mathcal{B}_0) \setminus \ldots \setminus \psi^*(x_n, \mathcal{B}_{n-1})$$

where $\mathcal{B}_i$ is the residual blind region for the i^{th} step in path. Now we can rearrange to produce

$$\mathcal{B}_n = \mathcal{B} \setminus \bigcup_{i=0}^{n-1} \psi^*(x_i, \mathcal{B}_{i-1})$$

Now since $\psi^*(x_i, \mathcal{B}_{i-1}) = \psi(x_i, \theta^*(x_i, \mathcal{B}_{i-1})) = \psi(x_i, \theta_i)$, because the optimal yaw is added to the path for a given residual blind region,

$$\mathcal{B}_n = \mathcal{B} \setminus \bigcup_{(x_i,\theta_i)\in\rho} \psi(x_i, \theta_i)$$

□

Theorem 1. *If there exists a path, ρ, such that* $\textsc{Area}(\mathcal{B} \cap \bigcup_{(x,\theta)\in\rho} \psi(x,\theta)) > \gamma \cdot \textsc{Area}(\mathcal{B})$ *and ρ can be executed by the robot with a total cost less than C, Algo. 1 will return such a path in a finite number of steps.*

Proof. Using Lemmas 1 and 2, we know that any path returned from Algo. 1 will have a total cost less than C and it will be returned in a finite number steps. We also know from Lemma 3, that the blind region at the n^{th} iteration is $\mathcal{B} \setminus \bigcup_{(x_i,\theta_i)\in\rho} \psi(x_i, \theta_i)$. Now, note that,

$$\begin{aligned}
\mathcal{B} \backslash \mathcal{B}_n &= \mathcal{B} \backslash (\mathcal{B} \setminus \bigcup_{(x_i,\theta_i)\in\rho} \psi(x_i, \theta_i)) \\
&= \mathcal{B} \cap \bigcup_{(x_i,\theta_i)\in\rho} \psi(x_i, \theta_i)
\end{aligned}$$

Now since, the algorithm only returns a path if $\textsc{Area}(\mathcal{B} \backslash \mathcal{B}_n) > \gamma \cdot \textsc{Area}(\mathcal{B})$ and using the result above, the algorithm will only return a path such that $\textsc{Area}(\mathcal{B} \cap \bigcup_{(x_i,\theta_i)\in\rho} \psi(x_i, \theta_i)) > \gamma \cdot \textsc{Area}(\mathcal{B})$

□

REFERENCES