Online Motion Planning MA-INF 1314
Bug-Algorithm

Elmar Langetepe
University of Bonn

Repetition: Pledge Algorithm with sensor errors

Pledge-like curve!

Def. $\mathcal{K}$ class of curves in $\mathcal{C}_{\text{frei}} \cup \mathcal{C}_{\text{halb}}$, with the following conditions:

1. Parameterized curve with turn-angles and position:
   $C(t) = (P(t), \varphi(t))$ mit $P(t) = (X(t), Y(t))$
2. Curve surrounds obstacel by Left-Hand-Rule
3. Leavs point is a vertex of an obstacle
4. $\mathcal{C}_{\text{free}}$-condition holds:
   $\forall t_1, t_2 \in C : P(t_1), P(t_2) \in \mathcal{C}_{\text{free}} \Rightarrow |\varphi(t_1) - \varphi(t_2)| < \pi$
5. $\mathcal{C}_{\text{half}}$-condition holds:
   $\forall h_i, t \in C : P(t) = P(h_i) \Rightarrow \varphi(t) - \varphi(h_i) < \pi$

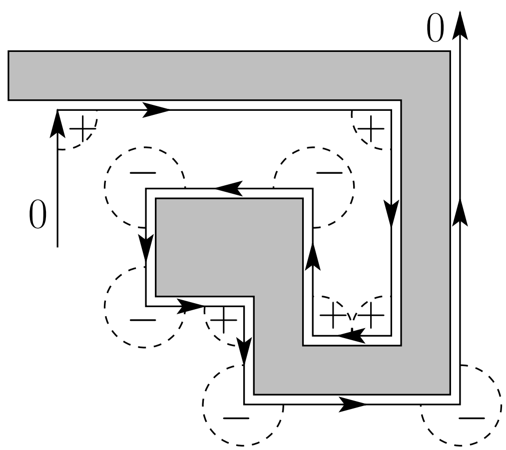
Rep.: Proof correctness

- **Lemma** Curves from $\mathcal{K}$ do not self-intersect.
- **Lemma** Curves from $\mathcal{K}$ hit any edge once.
- **Lemma** For any curve from $\mathcal{K}$: Obstacle will no longer be left, then the curve is enclosed by the obstacle.
- **Theorem** Curves from $\mathcal{K}$ escape, if this is possible.

Rep.: Applications of the model

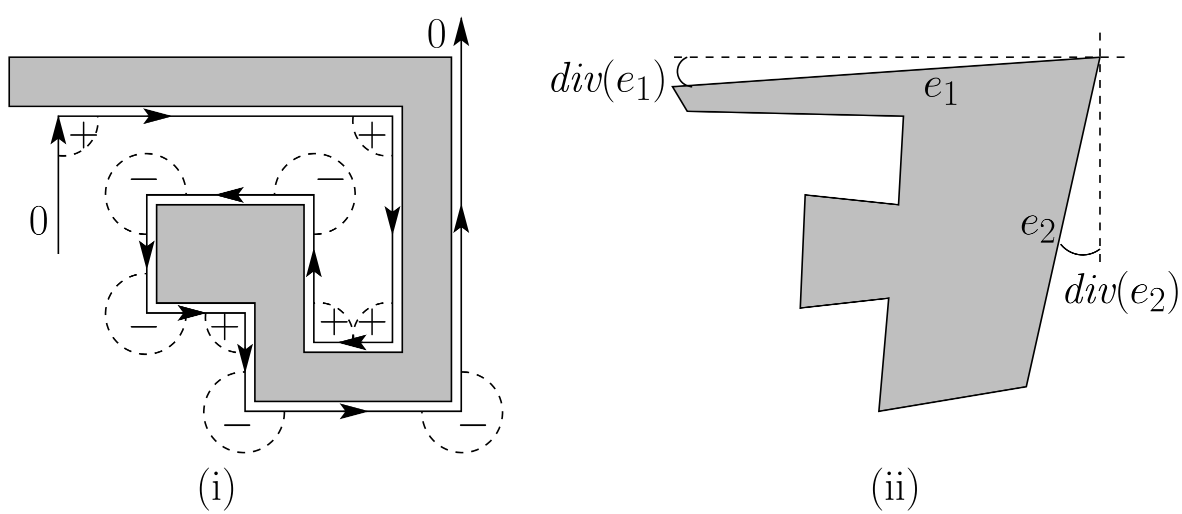
Corollary Compass with deviation maximal $\frac{\pi}{2}$ is sufficient for escaping from a labyrinth.

Corollary Axis-parallel scene, hold the direction in the range $(-\frac{\pi}{2}, +\frac{\pi}{2})$ and distinguish between horizontal and vertical. Escape!



Rep: Pseudo orthogonal

- Small devaitions at the vertices! From global coordinates!
- 1. Condition: Numbers convex vert. $=$ reflex vert. $+ 4$
- Small deviations!
- div(e) : $e = (v, w)$ smallest deviation from horizontal/vertical line passing durch v und w
- $\mathrm{div}(P) := \max_{e \in P} \mathrm{div}(e) \leq \delta$, **Def.:** δ-pseudo orthogonal scene



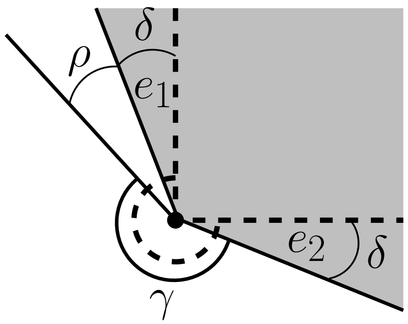
Rep: δ-pseudo orthogonal

Corollary δ-pseudo-orthogonal scene P. Measure angles with precision ρ s.th. $\delta + \rho < \frac{\pi}{4}$. Deviation in the free space always smaller than $\frac{\pi}{4} - 2\delta - \rho$ from global starting direction. Escape from a labyrinth is guaranteed

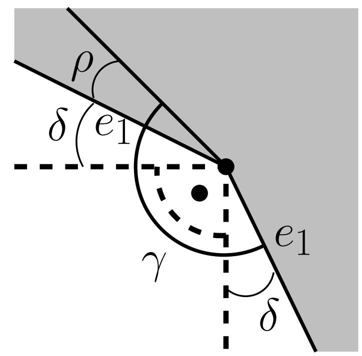
1. Distinguish reflex/convex corners: Counting the turns!
2. Max. global deviation of starting direction: Intervall π
3. Distinguish: Horizontal/Vertical

Rep.: δ-pseudo orthogonal scene

- Precision ρ with $\delta + \rho < \frac{\pi}{4}$
- Free-space max. deviation $\frac{\pi}{4} - 2\delta - \rho$
- 1. Distinguish reflex/convex corners: Worst-case



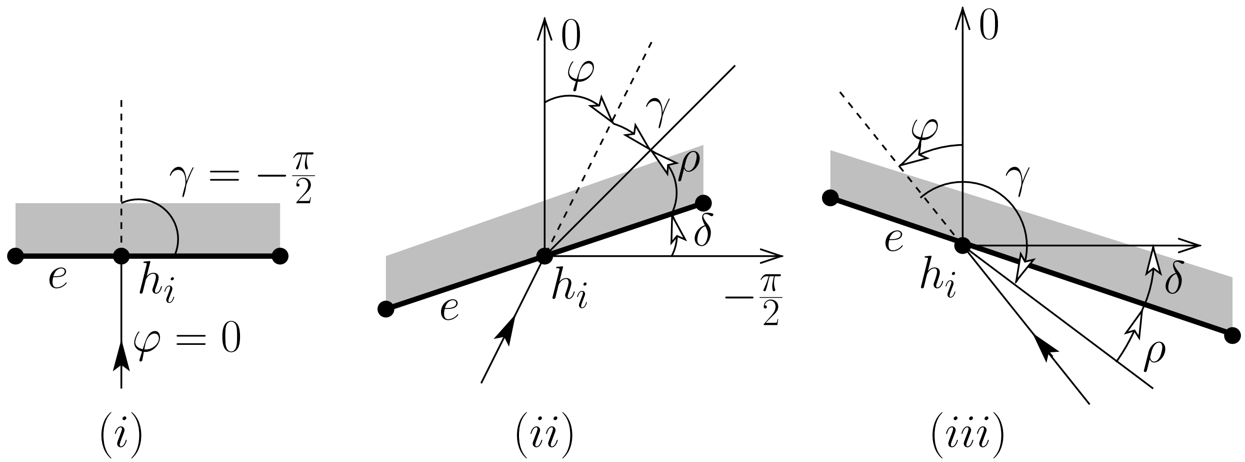
convex vertex



reflex vertex

Szene δ-pseudo orthogonal

- Precision ρ with $\delta + \rho < \frac{\pi}{4}$
- Free-space max. deviation $\frac{\pi}{4} - 2\delta - \rho$
- 3. Horizontal/vertical: Worst-case

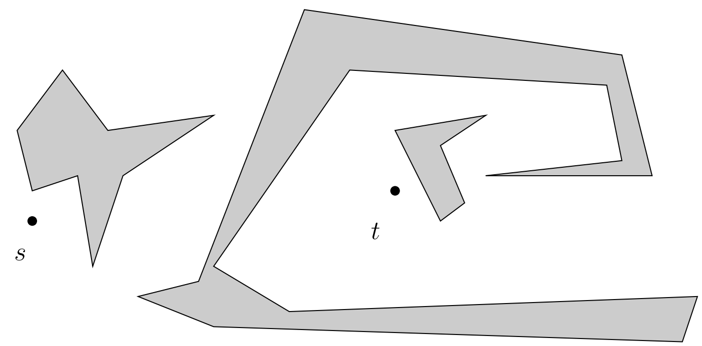


Szene δ-pseudo-orthogonal

- Precision ρ with $\delta + \rho < \frac{\pi}{4}$
- Free-Space deviation $\frac{\pi}{4} - 2\delta - \rho$
- 2. Max. global deviation of starting direction: Intervall π
- Leave in $[-\delta, \delta]$
- Deviation for the next hit: $\frac{\pi}{4} - 2\delta - \rho$

Find a target point

- Searching for a given goal: Navigation
- Polygonal environment: Finite number of polygons
- Touch sensor: Hand-Rules
- Start s, target t, coordinates are given
- Finite storage: I.e. Own coordinates
- BUG Algorithms: Sojourner



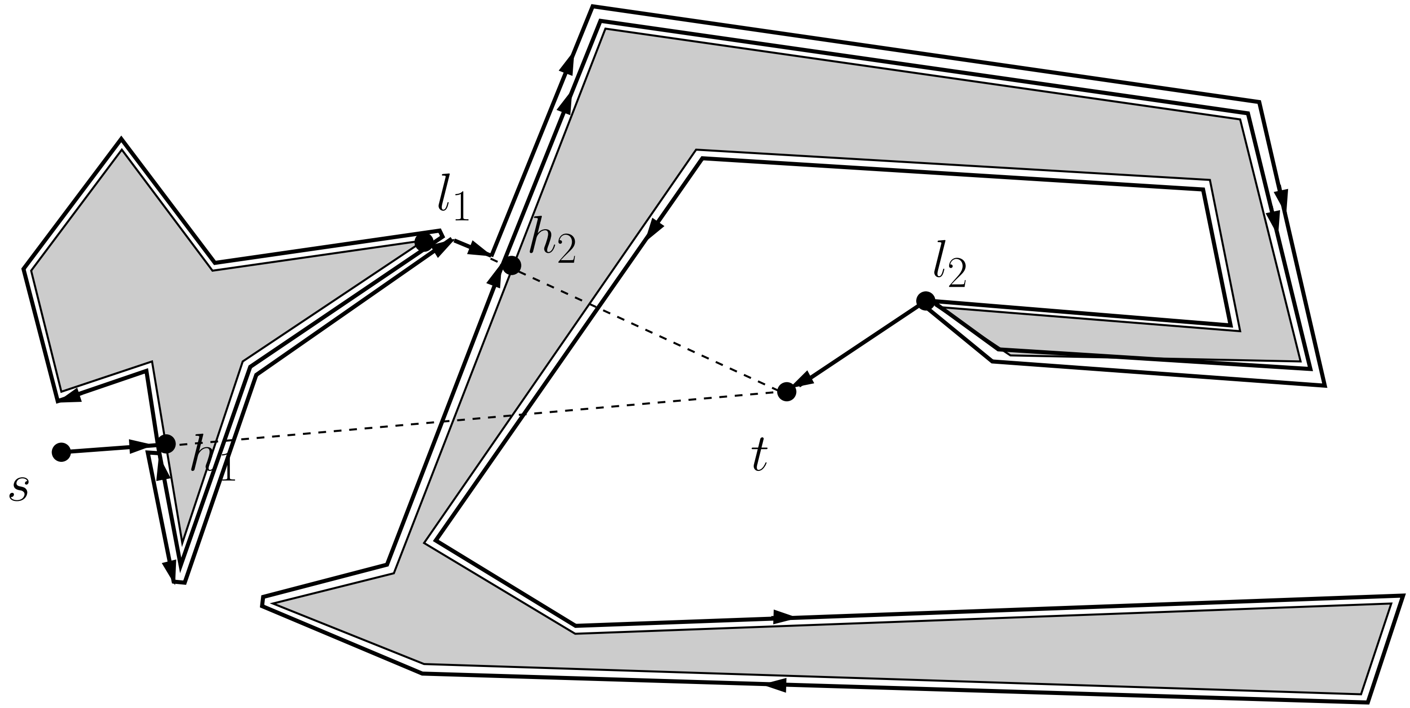
Notations

- $|pq|$ distance between p and q
- $D := |st|$ distance from start to goal
- Π_S path of strategy S from start to goal
- $|\Pi_S|$ length of the path Π_S
- $\mathtt{UP_i}$ perimeter of obstacle P_i.
- Actions:

  1. Move into direction of the target
  2. Follow the wall

- Leave-Points l_i, Hit-Points h_i

BUG1 strategy: Lumelsky/Stepanov



BUG1 strategy: Lumelsky/Stepanov

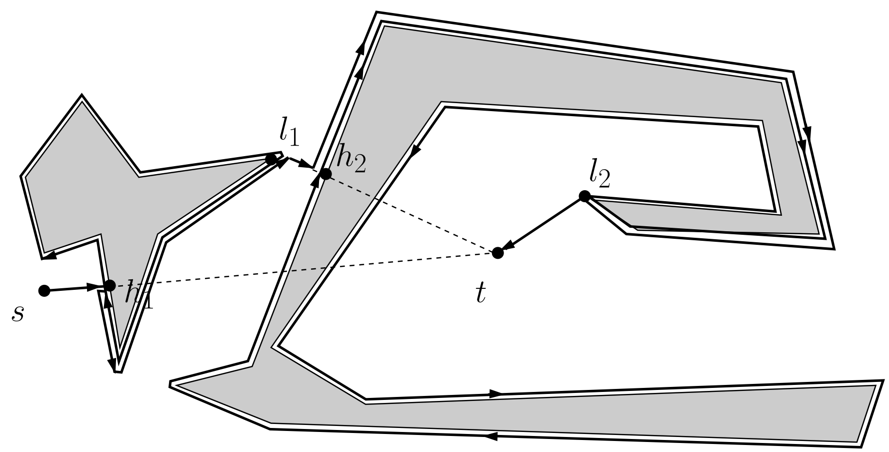
0. $l_0 := s$, $i := 1$
1. From l_{i-1} move into target direction, until
   (a) Goal is reached: Stop!
   (b) An obstacle is met at h_i.
2. Surround the obstacle O in cw order — continuously calculate and store the point l_i on O closest to t —, until
   (a) Goal is reached: Stop!
   (b) h_i is visited again!
3. Move along the shortest path along O to l_i.
4. Increment i, GOTO 1.

Correctness BUG1 strategy

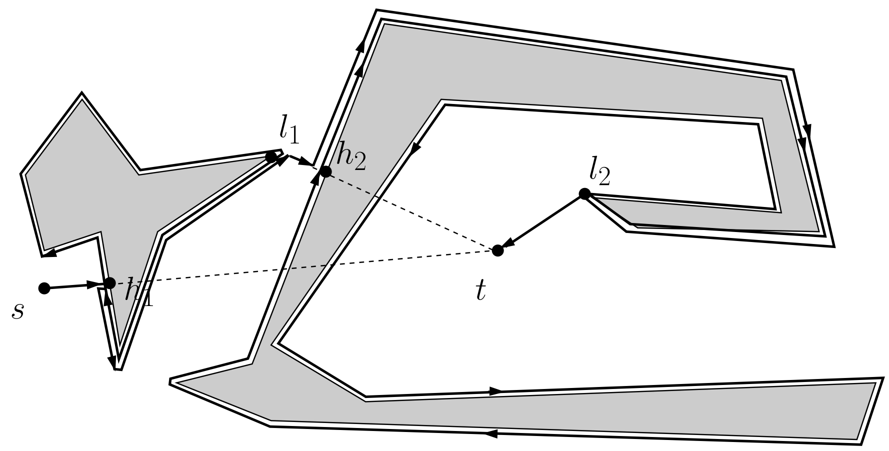
Theorem The strategy BUG1 finds a path from s to t, if such a path exists.

Proof:

- Sequence of Hit- and Leave-Points h_i, l_i
- $|st| \geq |h_1 t| \geq |l_1 t| \ldots \geq |h_k t| \geq |l_k t|$



Theorem Correctness BUG1 strategy



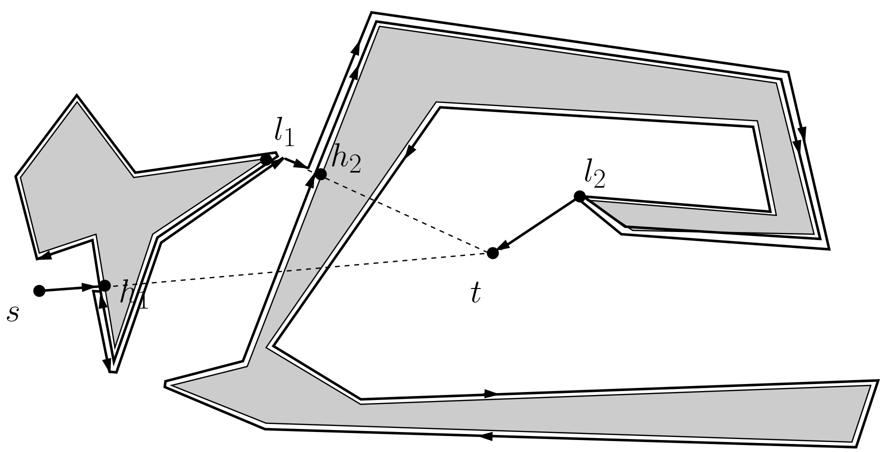
- Point with smallest distance to t: Leave-Point l_i
- No free movement to $t \Rightarrow$ enclosed
- $l_i \neq l_j$, new obstacle!
- Finitely many obstacles $\Rightarrow$ correctness

Path length BUG1 strategy

Theorem Let $\Pi_{\texttt{Bug1}}$ be the path from s to t, calculated by the BUG1-strategy. We have: $|\Pi_{\texttt{Bug1}}| \leq D + \frac{3}{2}\sum_i \texttt{UP}_\texttt{i}$.

Proof:

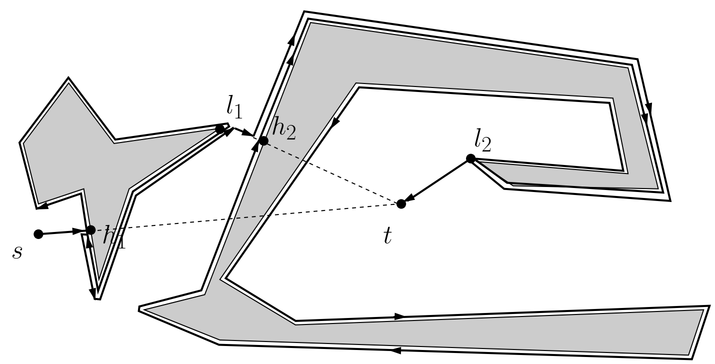
- Subdivision: Free space path, surrounding
- Surrounding, then shortest path to l_i
- $\frac{3}{2}\sum \texttt{UP}_\texttt{i}$
- Finally: Path D' between the obstacles



Theorem $|\Pi_{\texttt{Bug1}}| \leq D + \frac{3}{2}\sum_i \texttt{UP}_\texttt{i}$.

Proof: D' between the obstacles

$$\begin{aligned} D' &= |sh_1| + |\ell_1 h_2| + \ldots + |\ell_{k-1} h_k| + |\ell_k t| \\ &\leq |sh_1| + |\ell_1 h_2| + \ldots + |\ell_{k-1} h_k| + |h_k t| \\ &= |sh_1| + |\ell_1 h_2| + \ldots + |\ell_{k-1} t| \\ &\ldots \\ &\leq |sh_1| + |\ell_1 t| \leq |sh_1| + |h_1 t| = |st| = D \end{aligned}$$



Lower bound?

- Show: Bug1 is $\frac{3}{2}$-competitive
- Surround the obstacles along the path
- **Corollary** Bug1 is $\frac{3}{2}$-competitive
- Adversary strategy for the model
- Actions:

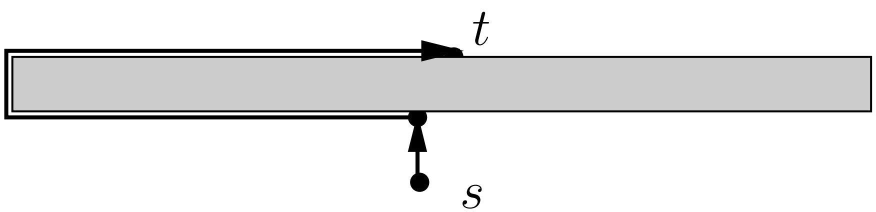
  1. Move into direction
  2. Follow the wall

- Leave-Points l_i, Hit-Points h_i

Lower bound

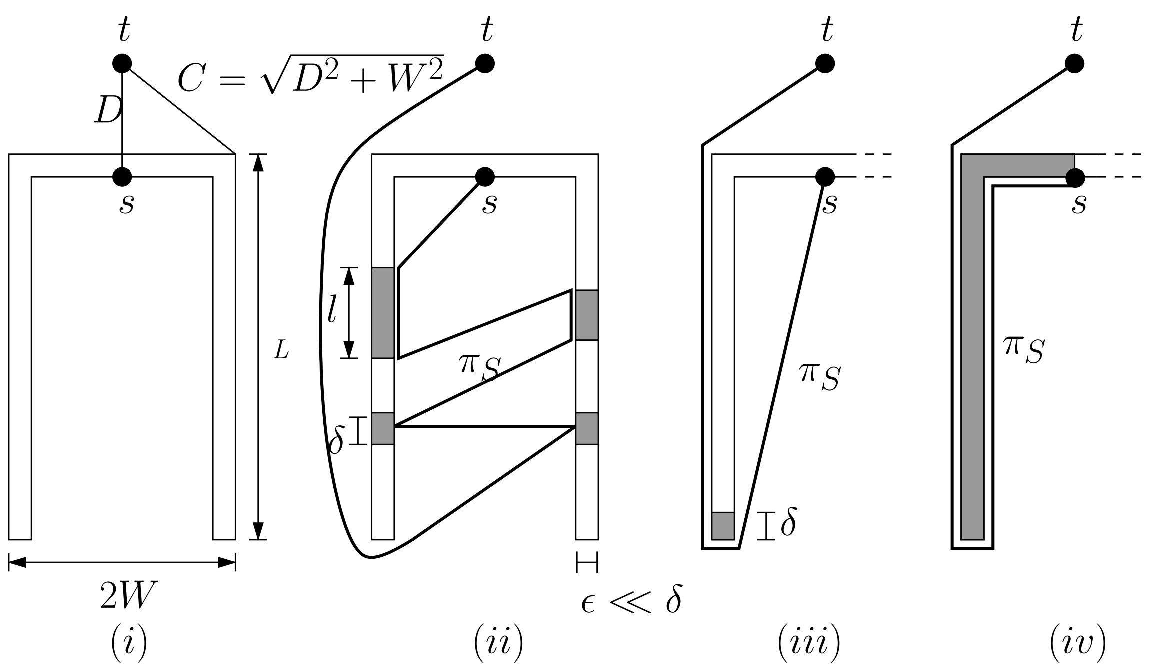
Theorem For any strategy S (due to the action-model), and for any $K > 0$, there exist a strategy with arbitrary $D > 0$, sucht that for any $\delta > 0$: $|\Pi_S| \geq K \geq D + \sum \mathtt{UP_i} - \delta$.

Arbitrarily large path!



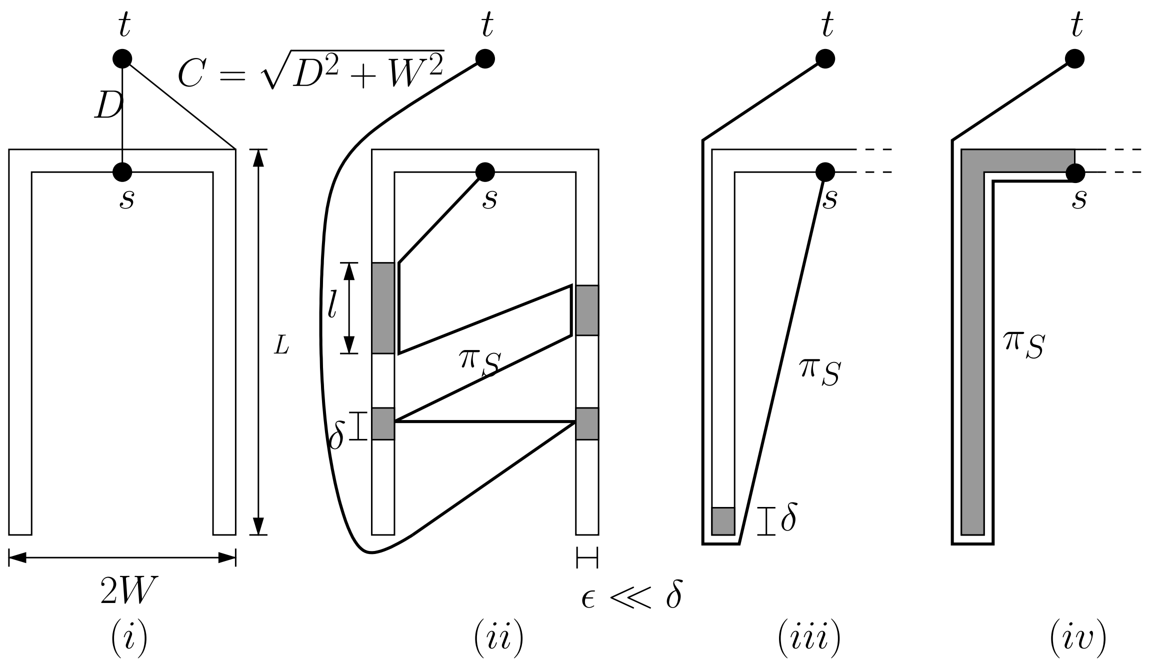
$$|\Pi_S| \geq K \geq D + \sum \mathtt{UP_i} - \delta.$$

- Virtual horse-shoe, Width $2W$, Thickness $\epsilon \ll \delta$, Length L, Distance D
- Virtual gets precise: Touch the wall!
- For any strategy S



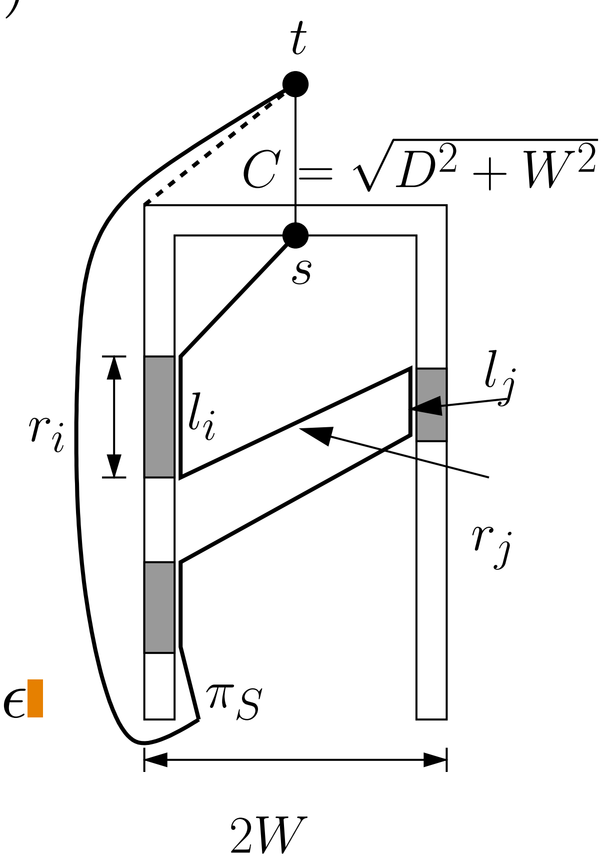
$$|\Pi_S| \geq K \geq D + \sum \mathtt{UP_i} - \delta.$$

- Idea: $D + W - \sqrt{D^2 + W^2} \leq \delta/2$ and
  $L + W - \sqrt{L^2 + W^2} \leq \delta/2$, L, W large enough!
- $|\Pi_S| \geq \sqrt{L^2 + W^2} + L + \sqrt{D^2 + W^2}$
  $\geq D + W + L + W - \delta = D + 2(L + W) - \delta$



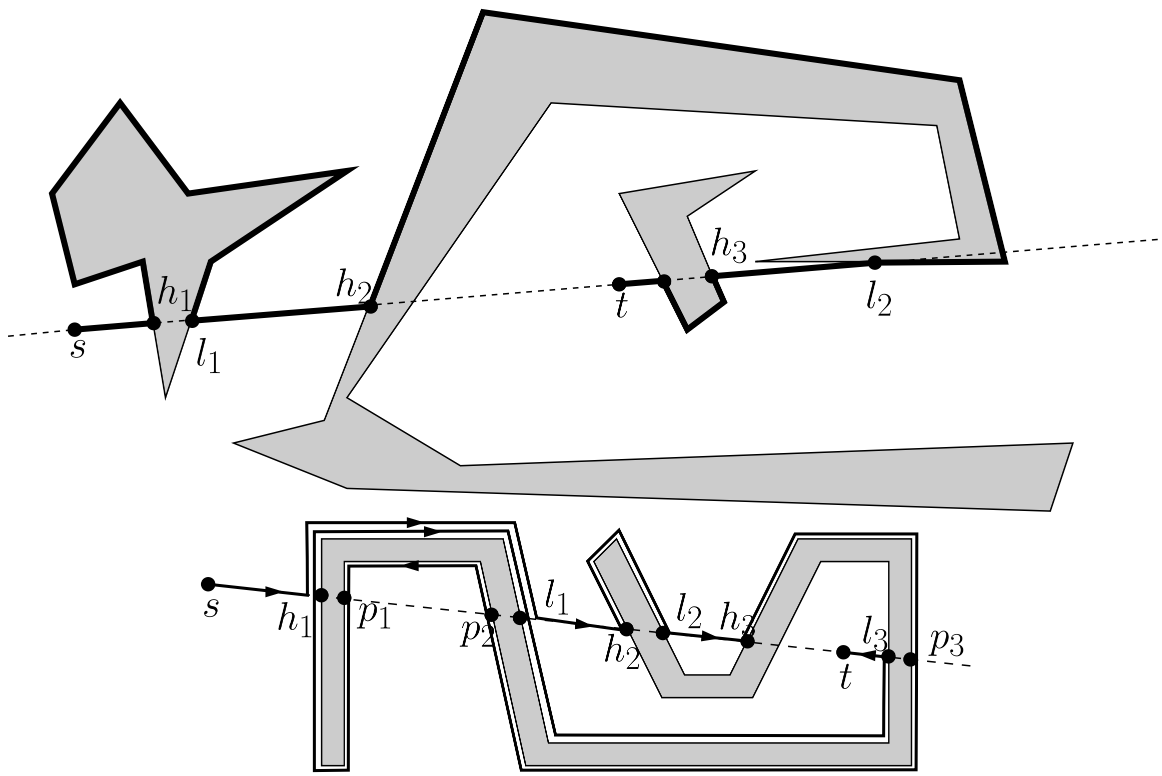
$$|\Pi_S| \geq K \geq D + \sum \mathtt{UP_i} - \delta.$$

- Problem: Left and right part! Peri. $4(L+W)$
- Inside horse-shoe: $|\Pi_{I_1}| \geq \sum \frac{1}{2}\mathtt{UP_i}$ non-overlapping
- $|\Pi_{I_2}| \geq \sum \mathtt{UP_j}$ overlapping, r_j path back
- Outside horse-shoe: $|\Pi_A| \geq L + C$ with $C = \sqrt{D^2 + W^2}$
- $L_{A_1} \geq \sum \frac{1}{2}\mathtt{UP_i}$ for non-overlapping
- Altogether: $|\Pi_S| \geq \sqrt{D^2+W^2} + \sum \mathtt{UP_i} - 2n\epsilon$
- $n \leq \frac{2L}{\delta}$, $\epsilon \leq \delta^2/(4L)$ gives $2n\epsilon \leq \delta/2$
- $|\Pi_S| \leq D + W + \sum \mathtt{UP_i} - \delta$



BUG2 strategy

Line G passing st, target direction, surround obstacle, shortest curr. distance on G, move to target



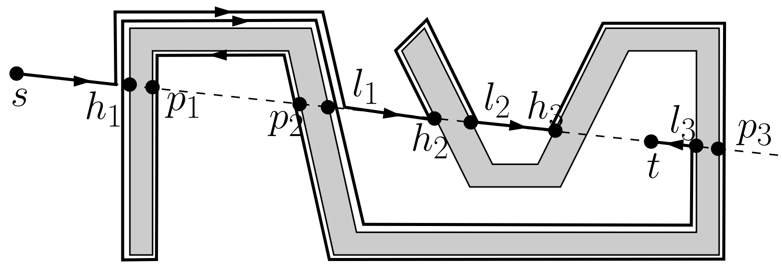
BUG2 strategy

0. $l_0 := s$, $j := 1$
1. From l_{j-1} move toward target, until
   (a) Goal is reached: Stop!
   (b) Obstacle is met at h_j.
2. Surround obstacle cw order, until
   (a) Goal is reached: Stop!
   (b) Line G passing st is visited at q, $|qt| < |h_j t|$ and $\overline{qt}$ locally free for a move
   $l_j := q$, $j := j + 1$ and GOTO 1.
   (c) h_j is reached again, no point q of case b) was found. Reaching the goal is impossible.

BUG2 strategy: Analysis

- Structural properties
- Correctness and performance
- **Lemma** Bug2 visits finitely many obstacles

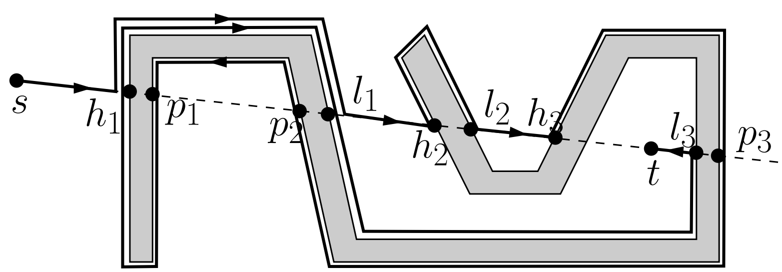
Proof, by precondition for the scene!



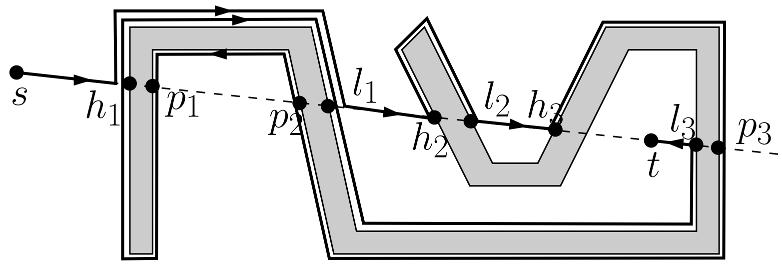
BUG2 strategy: Property

Lemma Let n_i denote the number of intersections between G (line passing st) and the obstacle P_i. Any boundary point of P_i is visited at most $\frac{n_i}{2}$ time.

- Bug2 defines pairs (h_j, l_j) of hit- and leave points
- Jumping cond.: $|h_j t| > |l_j t| > |h_{j+1} t|$.
- Any intersection with P_i is only once a leave or a hit point
- Meet current hit point $\Rightarrow$ Stop



Bug2 visits boundary points max $\frac{n_i}{2}$ times

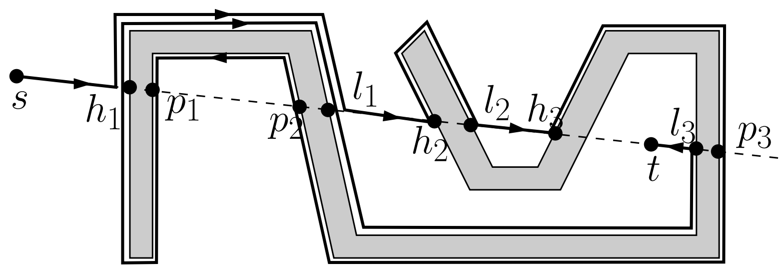


- Pairs (h_j, l_j) of hit- leave points
- $\frac{n_i}{2}$ pairs (h_j, l_j)
- Only then a surrounding is started
- Point on the boundary only $\frac{n_i}{2}$ times

BUG2 strategy: Correctness

Corollary Bug2 visits the goal, if this is possible.

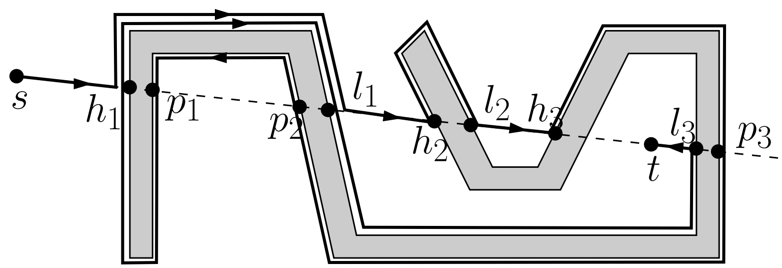
- Finitely many visits, finitely many surroundings!
- Either goal is found or current hit point is visited again
- Current hit point ⇒ no free path from a better point on the boundary. Goal is enclosed!



BUG2 strategy: Performance

Theorem Let $\Pi_{\texttt{Bug2}}$ denote the path from s to t designed by BUG2. We have $|\Pi_{\texttt{Bug2}}| \leq D + \sum_i \frac{n_i \texttt{UP}_\texttt{i}}{2}$. Proof:

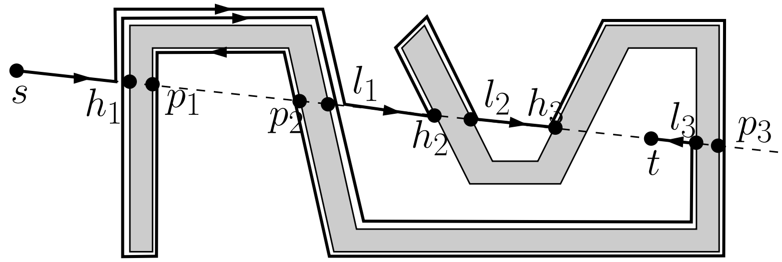
- Subdivision: Surroundings, Free path
- $\sum_i \frac{n_i \texttt{UP}_\texttt{i}}{2}$ follows from the **Lemma**
- Length D' between ostacles



BUG2 strategy: Performance

- Length D' between obstacles
- Analogously **BUG1 Theorem** $D' \leq D$
- Altogether:

$$|\Pi_{\mathrm{Bug2}}| \leq D + \sum_i \frac{n_i \mathtt{UP_i}}{2}.$$

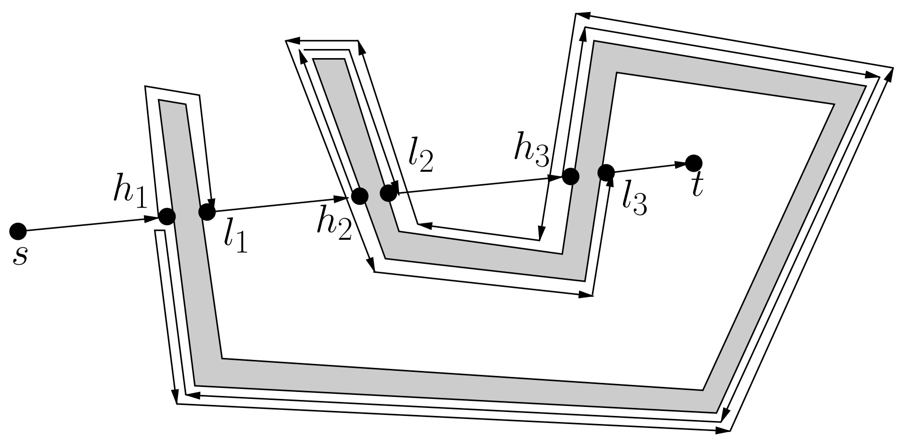


Compare BUG2 and BUG1

- BUG2 not always better, sometimes worse (Exercise)
- Convex polygons: Optimal
- Many further variants!
- Visibility/Local improvements!

Change I

- Bug1 fully surrounds
- Bug2 avoids, but visits many times
- Change make use of old Leave/Hit Points, **One** order change!



Pseudocode: Change I

0. $\ell_0 := s$, $i := 1$
1. Move from ℓ_{i-1} along line passing st toward goal, until
   (a) Goal is reached: Stop!
   (b) Obstacle is met at h_i.
2. Surround ostacle cw order, until
   (a) Goal is reached: Stop!
   (b) Line G passing st is visited at q, $|qt| < |h_j t|$ and $\overline{qt}$ locally free for a move, $l_j := q$, $j := j + 1$ and GOTO 1.

(c) A hit- or leave point h_j or ℓ_j with $j < i$ is met. Move back to h_i, use ccw order until (a), (b) oder (d) happens.

(d) h_j is reached again, no point q of case b) was found. Reaching the goal is impossible.