Vorlesung

Geometrische Algorithmen

Bewegungsplanung bei unvollständiger Information

Sven Schuierer

Überblick

1. Suchen nach einem Zielpunkt
2. Strategie Bug
3. Kompetitive Strategien
4. Suche nach Loch im Lattenzaun
5. Kompetitive Suche in einem einfachen Polygon
6. Online Navigation—andere Aufgaben
7. Zielkompaß und Tastsensor

1 Suchen nach einem Zielpunkt

Umgebung (Arbeitsraum):

- planarer Bereich mit polygonalen Hindernissen $\mathcal{H}$
- statisch
- nicht bekannt

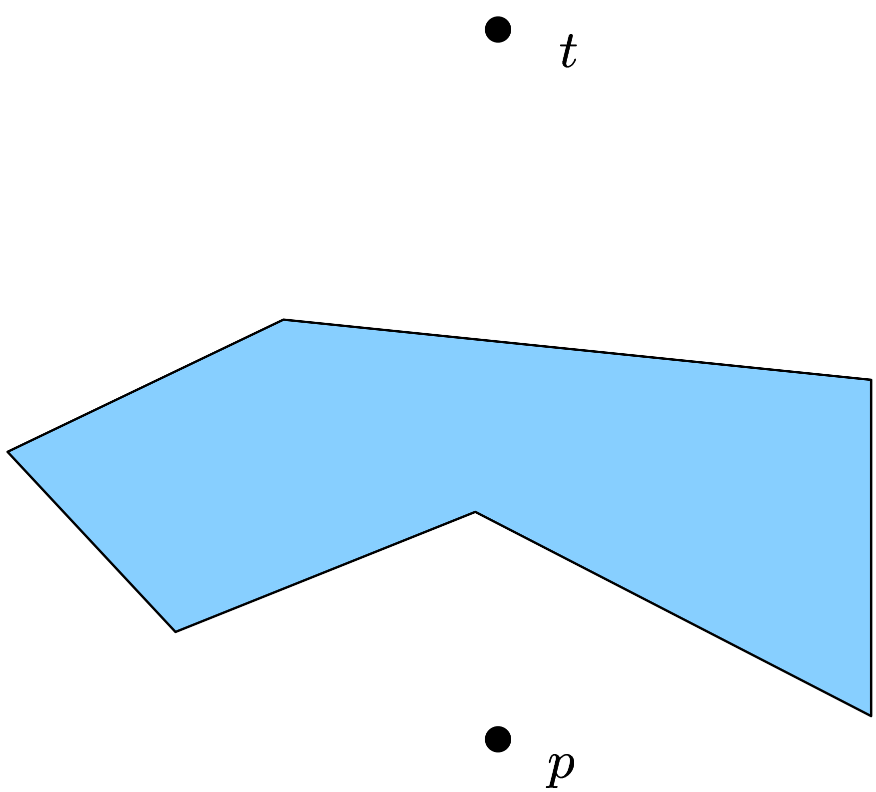
Roboter $\mathcal{R}$:

- punktförmig
- in alle Richtungen beweglich
- kennt seine Koordinaten
- kennt Koordinaten des Zielpunktes t
- Tastsensor

2 Strategie Bug

Lumelsky und Stepanov (1987):

- laufe auf das Ziel zu bis ein Hindernis $\mathcal{P}$ getroffen wird
- umrunde $\mathcal{P}$ und merke den nächsten Punkt p^* auf $\mathcal{P}$ zu t
- gehe zu p^* und laufe weiter auf das Ziel zu



Strategie Bug

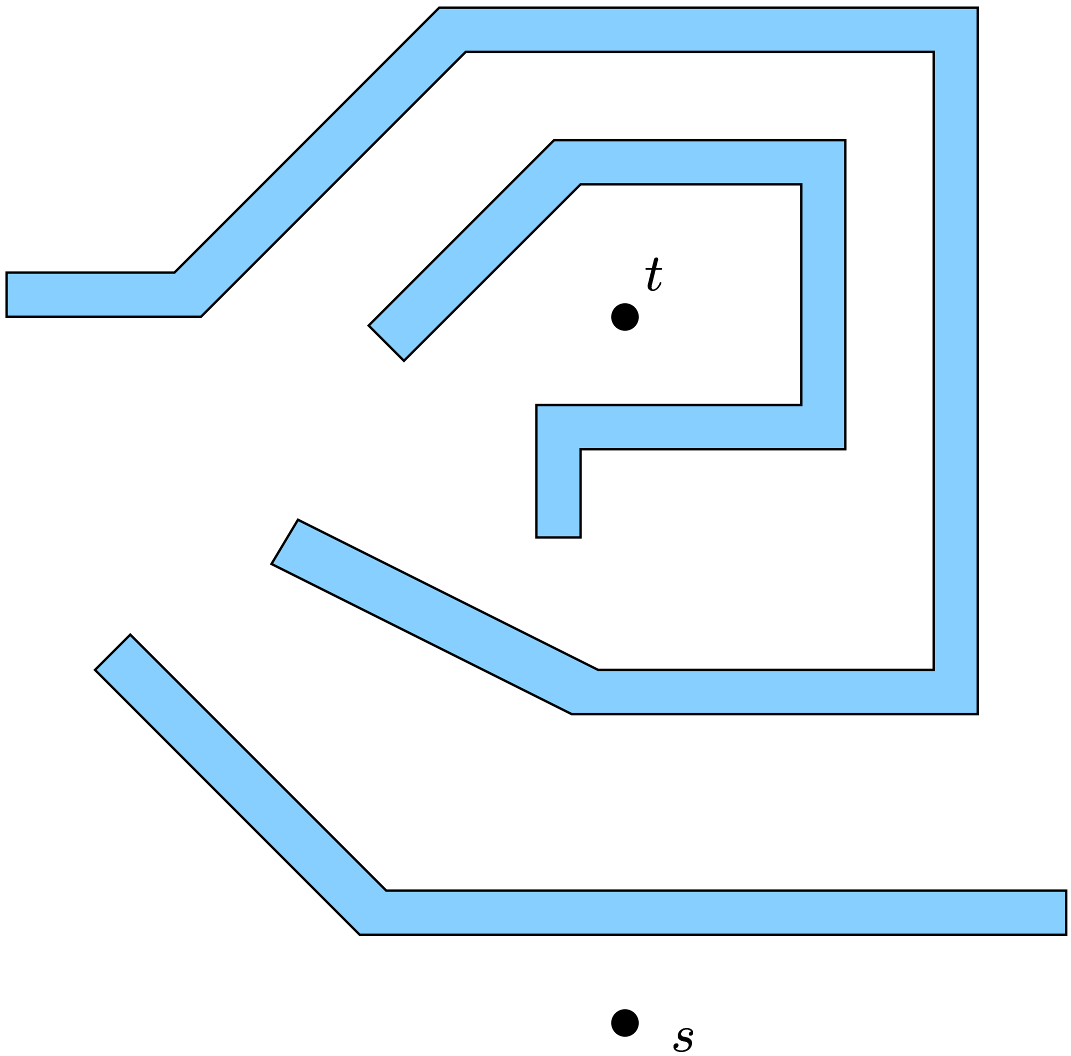
Algorithmus *Bug*

var r = die akutelle Position des Roboters

```
repeat
  repeat
    laufe auf Zielpunkt t zu
  until Wandkontakt mit Hindernis 𝒫
  p_hit ← r /* Auftreffpunkt */
  p* ←  r /* zu t nächster Punkt auf 𝒫 */
  repeat
    repeat
      rücke entlang der Wand von 𝒫 vor
    until |rt| ist lokales Minimum or r = p_hit
    if |rt| < |p*t| then p* ← r
  until r = p_hit

  gehe auf kürzestem Weg längs der Wand
     von 𝒫 zu p*
until r = t
```

Bug—Beispiel



Bug—Korrektheit

Satz

Die Strategie Bug findet stets einen Weg vom Startpunkt s zum Zielpunkt t, wenn solch ein Weg überhaupt existiert.

Beweis:

Annahme: Ein Weg existiert

$\boldsymbol{p_1, p_2, \ldots}$ Punkte, in denen der Roboter auf ein Hindernis trifft

$\boldsymbol{\mathcal{P}_1, \mathcal{P}_2, \ldots}$ getroffene Hindernisse

$\boldsymbol{p_1^*, p_2^*, \ldots}$ zu t nächstgelegene Punkte auf Hindernissen $\mathcal{P}_1, \mathcal{P}_2, \ldots$

Es gilt:

$$|p_i t| \geq |p_i^* t|$$

und

$$|p_i^* t| \geq \left|p_{i+1} t\right|.$$

Bug—Korrektheit

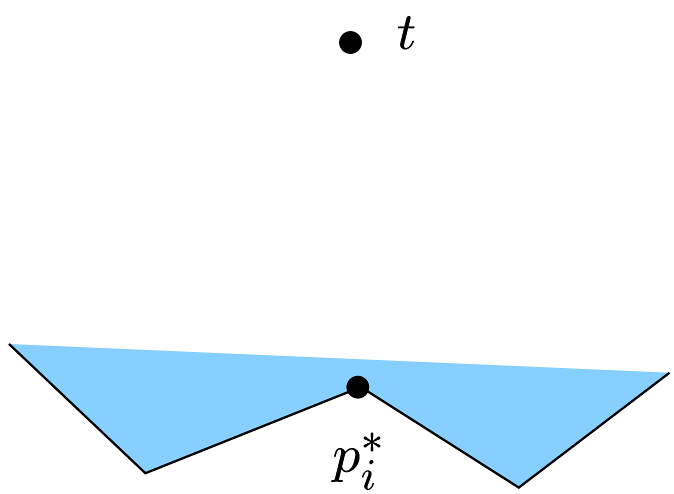
Beweis: (Fortsetzung)

Behauptung:

Bei p_i^* ist der Weg in Richtung t frei

Annahme:

Inneres von $\mathcal{P}_i$ liegt in Richtung t bei p_i^*



$\Rightarrow$ Rand von $\mathcal{P}_i$ schneidet $\overline{p_i^* t}$ ein zweites Mal

$\Rightarrow$ zu t nächster Schnittpunkt p^* gehört zur gleichen Zusammenhangskomponente wie s und p_i^*

$\Rightarrow$ p^* gehört zum gleichen Teil des Randes von $\mathcal{P}$ wie p_i^*—**Widerspruch**

Bug—Korrektheit

Beweis: (Fortsetzung)

Also:

$$|p_i^* t| > \left|p_{i+1} t\right|$$

und damit

$$|p_i^* t| > \left|p_{i+1}^* t\right|$$

$\Rightarrow$ p_i^* und p_{i+1}^* liegen auf verschiedenen Hindernissen

Es gibt nur endliche viele Hindernisse!

Bug—Weglänge

Satz

Wenn ein nach der Strategie Bug vorgehender Roboter von s aus t erreicht, so ist die Länge des zurückgelegten Weges höchstens

$$|st| + \frac{3}{2}\sum_{i=1}^{n} U_i,$$

wobei U_i der Umfang des i-ten Hindernisses ist.

Bug—Weglänge

Beweis:

$m =$ Anzahl der getroffenen Hindernisse
$p_0^* = s$ und $p_{m+1} = t$

Weg des Roboters setzt sich zusammen aus:
- Bewegungen von p_i^* nach p_{i+1} und
- Hindernisumrundungen von p_i nach p_i^*

Jedes Hindernis wird einmal vollständig umrundet und dann höchstens noch einmal zur Hälfte

$$\Rightarrow \text{ Hindernisumrundungen} \leq \frac{3}{2} \sum_{i=1}^{n} U_i$$

Bug—Weglänge

Bewegungen von p_i^* nach p_{i+1}:

$$|st| \geq \sum_{i=0}^{m} \left| p_i^* p_{i+1} \right| \qquad (*)$$

Wir zeigen durch Induktion:

$$|st| \geq \sum_{i=0}^{k-1} \left| p_i^* p_{i+1} \right| + |p_k t|$$

Damit folgt $(*)$

$k = 1$

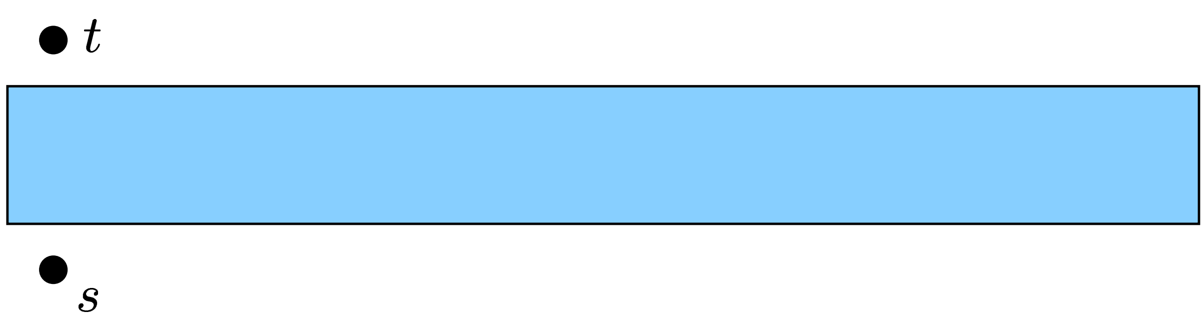
$k \geq 1$

$$|p_k t| \geq |p_k^* t| = \left| p_k^* p_{k+1} \right| + \left| p_{k+1} t \right|$$

$\Rightarrow$ Behauptung $\square$

Bug—Weglänge

Bug kann beliebig schlecht werden



7 Zielkompaß und Tastsensor

Hemmerling, 1994

Drei Grundbefehle:

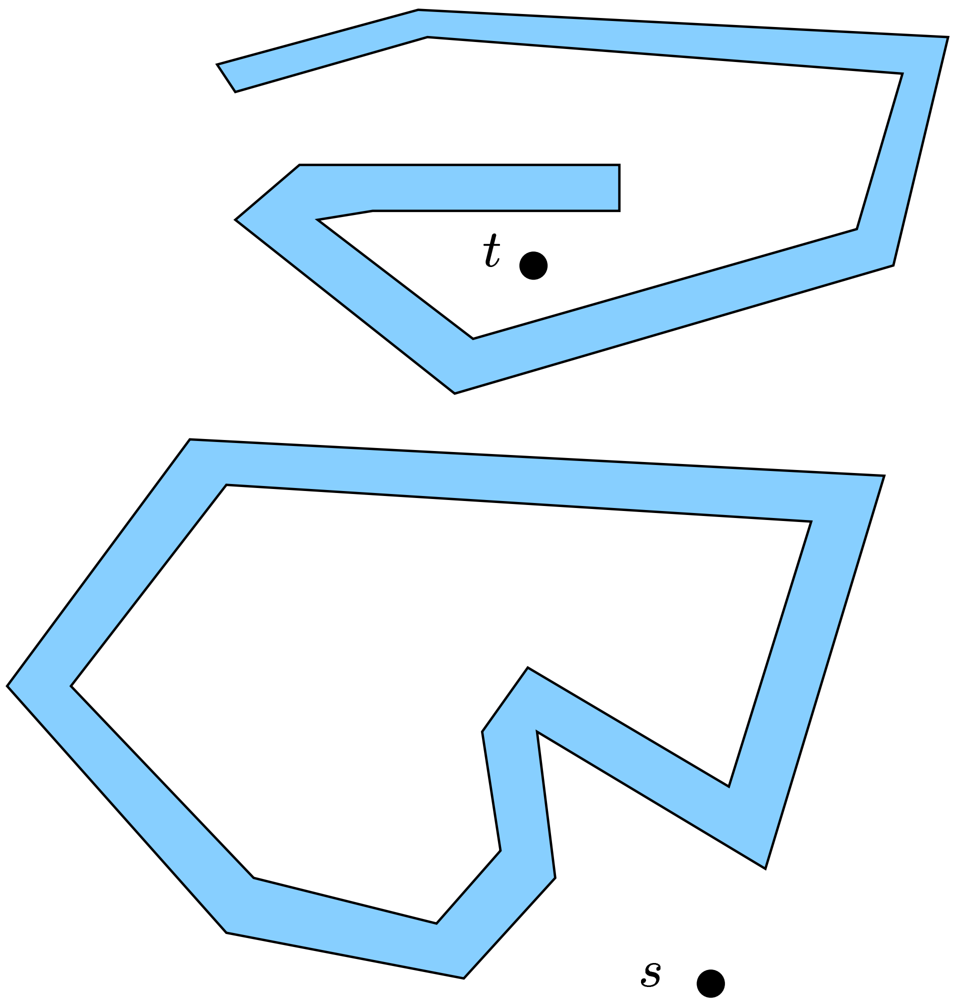
- T: Laufe von einer Wandecke geradlinig auf t zu, bis t oder andere Wand erreicht ist
- L: Laufe von einem Wandpunkt im Gegenuhrzeigersinn an der Wand entlang, bis nächster Eckpunkt erreicht ist
- R: Wie L nur im Uhrzeigersinn

$\Rightarrow$ Alphabet: $\Sigma = \{T, L, R\}$

Steuerwort: Wort $w \in \Sigma^*$

Steuerwort—Beispiel

$$w(s) = TL^2TR^5T$$



Erreichbare Endpunkte

Betrachte $s \in \mathcal{C}_{free}$:

W durch ein Steuerwort beschriebener Weg beginnend bei s

W kann nur endlich viele Endpunkte haben:

- Startpunkt s
- Endpunkt t
- Ecken der Hindernisse
- Endpunkte von freien T-Bewegungen, die an s oder einer Hindernisecke starten

$\Rightarrow$ Endpunkte $p_1, p_2, \ldots, p_m$

Für jeden Punkt p_i:

$w(p_i)$ ein Steuerwort, das den Roboter von p_i ans Ziel bringt

Universelle Steuerworte

Lemma

Sei $\mathcal{H}$ eine polygonalen Umgebung und s eine Startposition und t eine Zielposition im freien Raum von $\mathcal{H}$, so daß s und t miteinander verbindbar sind.

Dann gibt es ein endliches universelles Steuerwort $w(s, t, \mathcal{H})$ über Σ, das den Roboter von jedem von s aus erreichbaren Punkt p_i zum Ziel t führt.

Universelles Steuerwortes

Beweis:

$w_1 = w(p_1)$ führt von p_1 zum Ziel

w_1 angewandt auf p_2 führt zu q_2
(nicht ausführbare Befehle werden übersprungen)

$\Rightarrow$ $w_2 = w_1 w(q_2)$ führt von p_1 und p_2 zum Ziel!

Induktiv:

w_i führt von $p_1, \ldots, p_i$ zum Ziel

w_i angewandt auf p_{i+1} führt zu q_{i+1}

$\Rightarrow$

$$w_{i+1} = w_i w(q_{i+1})$$

führt von $p_1, \ldots, p_{i+1}$ zum Ziel.

$\Rightarrow$ $w(s, t, \mathcal{H}) = w_m$ führt von $p_1, \ldots, p_m$ zum Ziel

Zielkompaß und Tastsensor

Satz

Im Prinzip genügen Zielkompaß und Tastsensor, um in unbekannter Umgebung einen Zielpunkt zu finden.

Beweis:

Problem: $w^* = w(s, t, \mathcal{H})$ ist unbekannt

Erzeuge der Reihe nach alle endlichen Wörter über Σ und laufe ihnen nach

$\Rightarrow$ Irgendwann wird w^* erzeugt

$\Rightarrow$ Unabhängig von der gegenwärtigen Position führt w^* zum Ziel

3 Kompetitive Strategien

Strategie S

$L_{opt}(s,t,P)$ = Länge des kürzesten Weges von s nach t in P

$L_S(s,t,P)$ = Länge des vom Robotes zurückgelegten Weges unter Verwendung der Strategie S (beginnend bei s in P, um t zu finden)

Definition

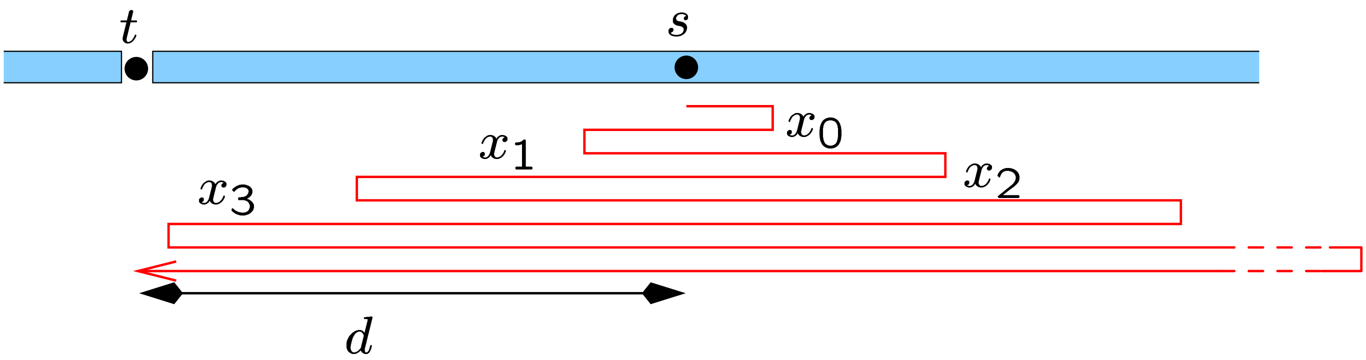
Eine Strategie S heißt c-kompetitiv, falls gilt:

$$L_S(s,t,P) \leq c \cdot L_{opt}(s,t,P),$$

für alle Startpositionen s, Zielpositionen t und Umgebungen P.

c wird das kompetitive Verhältnis genannt.

4 Suche nach Loch im Lattenzaun



Strategie:

gehe x_0 Schritte nach rechts
gehe x_1 Schritte nach links
gehe x_2 Schritte nach rechts
•
•
•

Distanz zum Ziel:

$$d \geq 1$$

Strategie I

Schritt i: gehe Distanz $i + 1$

Annahme:

Schritt $n + 2$: t wird gefunden

Vom Roboter zurückgelegte Distanz

$$L_S(t) = 2 \sum_{i=0}^{n+1} (i + 1) + d = 2 \frac{(n + 3)(n + 2)}{2} + d$$

Kompetitives Verhältnis:

$$\begin{aligned} C_S &= \sup_{n \geq 0} \sup_{n+1 < d \leq n+3} \frac{2 \sum_{i=0}^{n+1} (i + 1) + d}{d} \\ &= \sup_{n \geq 0} 1 + \frac{(n + 3)(n + 2)/2}{n + 1} \\ &= \infty \end{aligned}$$

Strategie II

Schritt i: gehe Distanz 2^i

Annahme:

Schritt $n+2$: t wird gefunden

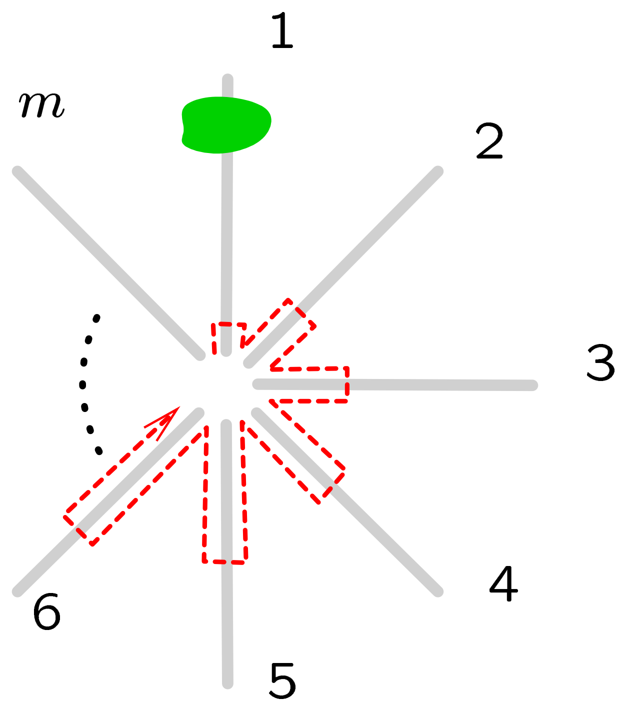
Vom Roboter zurückgelegte Distanz:

$$L_S(t) = 2\sum_{i=0}^{n+1} 2^i + d = 2(2^{n+2} - 1) + d$$

Kompetitives Verhältnis:

$$\begin{aligned} C_S &= \sup_{n\geq 0} \sup_{n+1<d\leq n+3} \frac{2\sum_{i=0}^{n+1} 2^i + d}{d} \\ &= \sup_{n\geq 0} 1 + \frac{2(2^{n+2} - 1)}{2^n} \\ &= 9 \end{aligned}$$

m-Wege Suche



m-**Wege Suche**:

- Besuche Strahlen zyklisch
- Erhöhe Schrittweite jedes Mal um den Faktor $m/(m-1)$

$$x_i = \left(\frac{m}{m-1}\right)^i$$

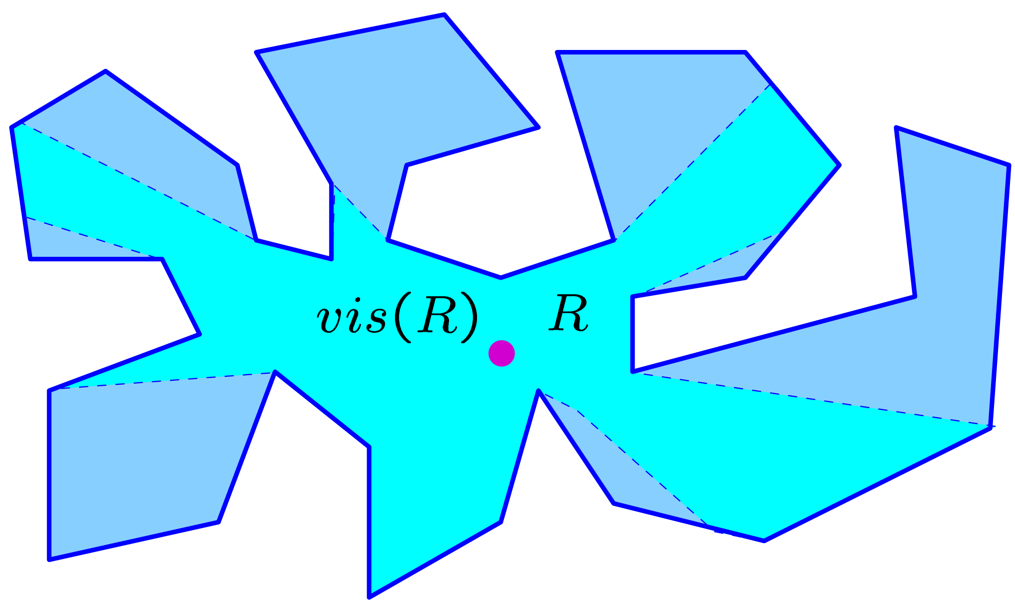
Damit

$$C_S = 1 + 2\frac{m^m}{(m-1)^{m-1}} \leq 1 + 2em$$

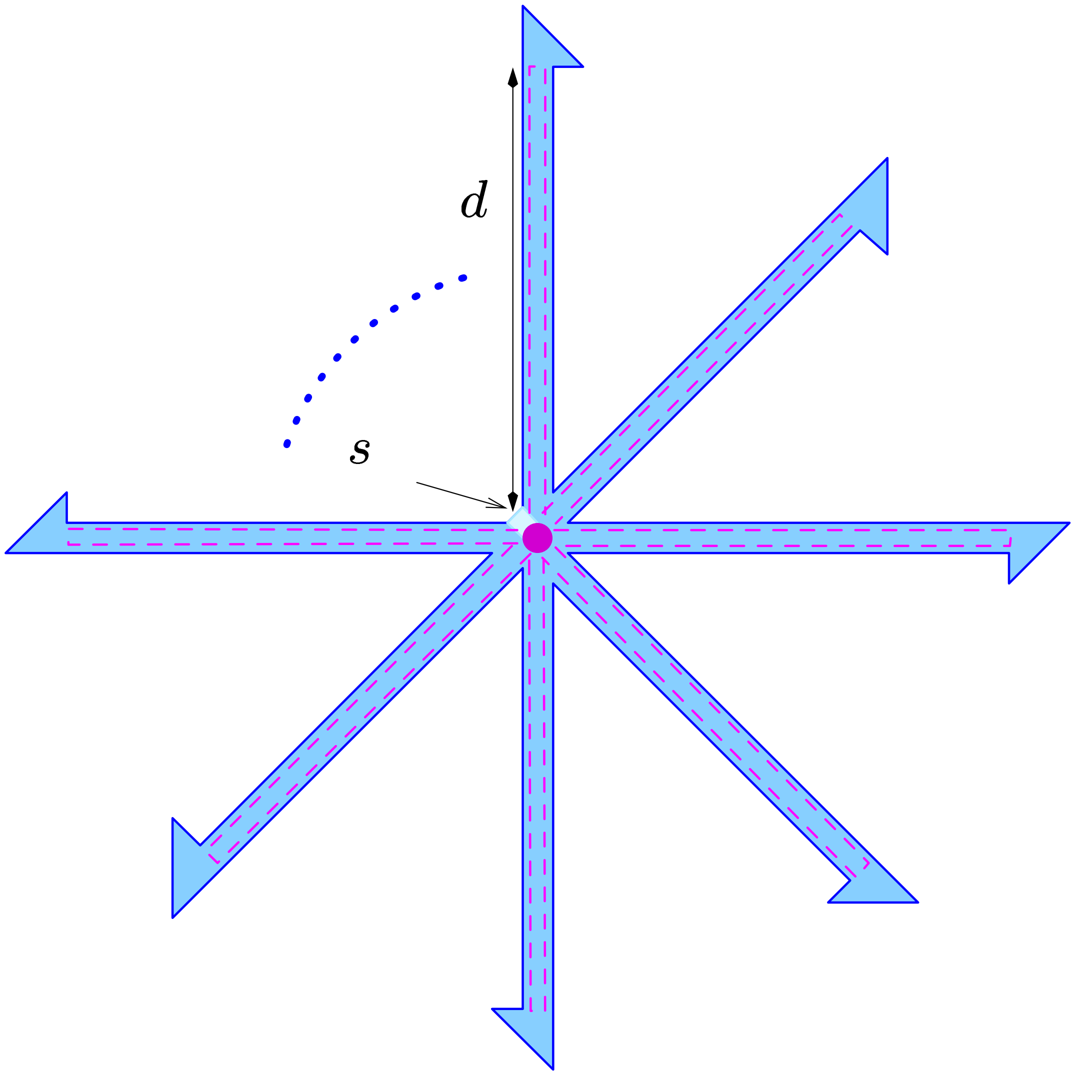
5 Kompetitive Suche in einfachen Polygonen

Robotermodell

- punktförmig
- exakte Bewegungen
- Kenntnis der eigenen Position
- Kenntnis der eigenen Orientierung (Kompaß)
- Distanzsensoren zur Bestimmung des Sichtbarkeitspolygons (z.B. Sonar, Ladar)
- keine Karte der Umgebung



Untere Schranke

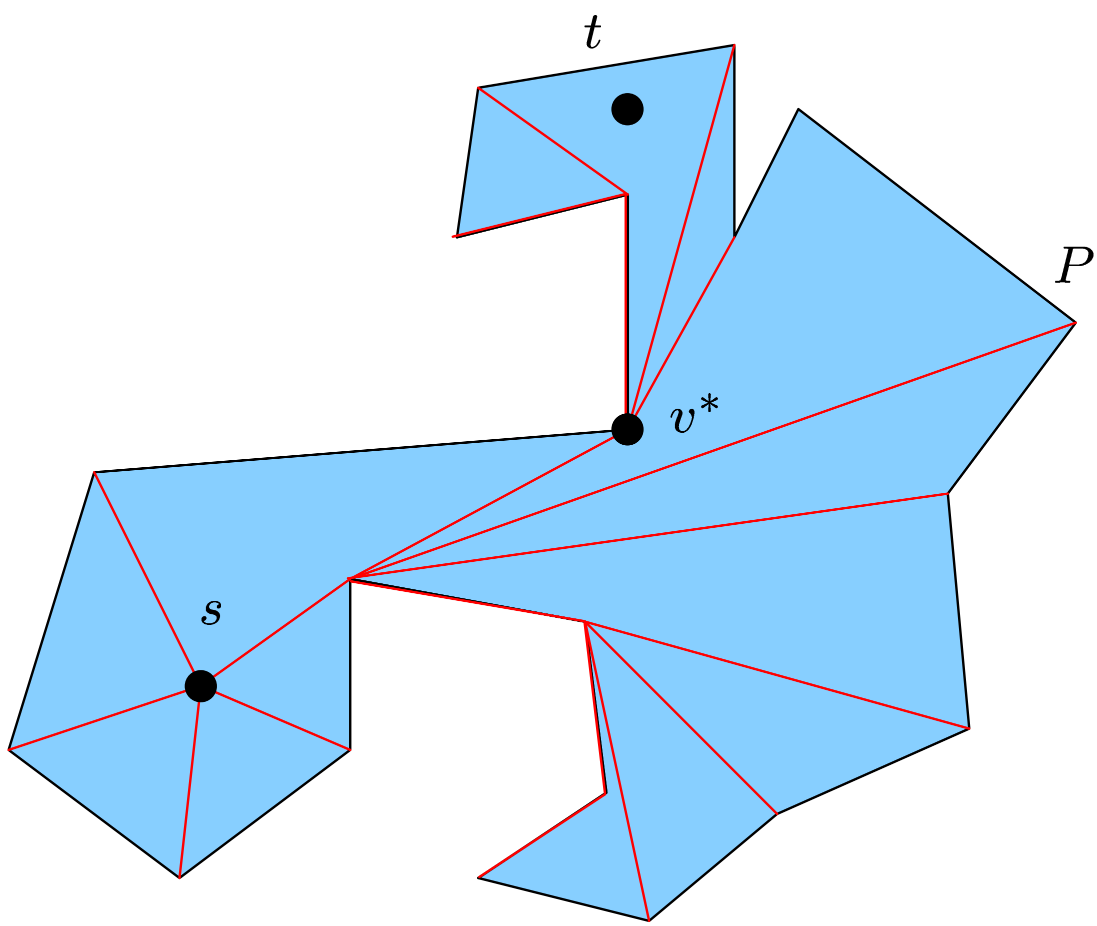


Kompetitives Verhältnis:

$$\frac{(2n/4 - 1)d}{d} = n/2 - 1$$

Suche in einfachen Polygonen

Kürzester Wege Baum



Der kürzeste Weg von s nach t geht über eine Ecke v^* von P (falls t nicht von s aus sichtbar ist)

Kompetitives Verhältnis

t ist von v^* aus sichtbar

Kompetitives Verhältnis, um t zu finden:

$$\begin{aligned}\frac{L_S(s,t)}{L_{opt}(s,t)} &= \frac{L_S(s,v^*)+d(v^*,t)}{L_{opt}(s,v^*)+d(v^*,t)}\\ &\leq \frac{L_S(s,v^*)}{L_{opt}(s,v^*)}\end{aligned}$$

$\Rightarrow$ Es genügt, ein Online-Verfahren für die Suche nach v^* anzugeben.

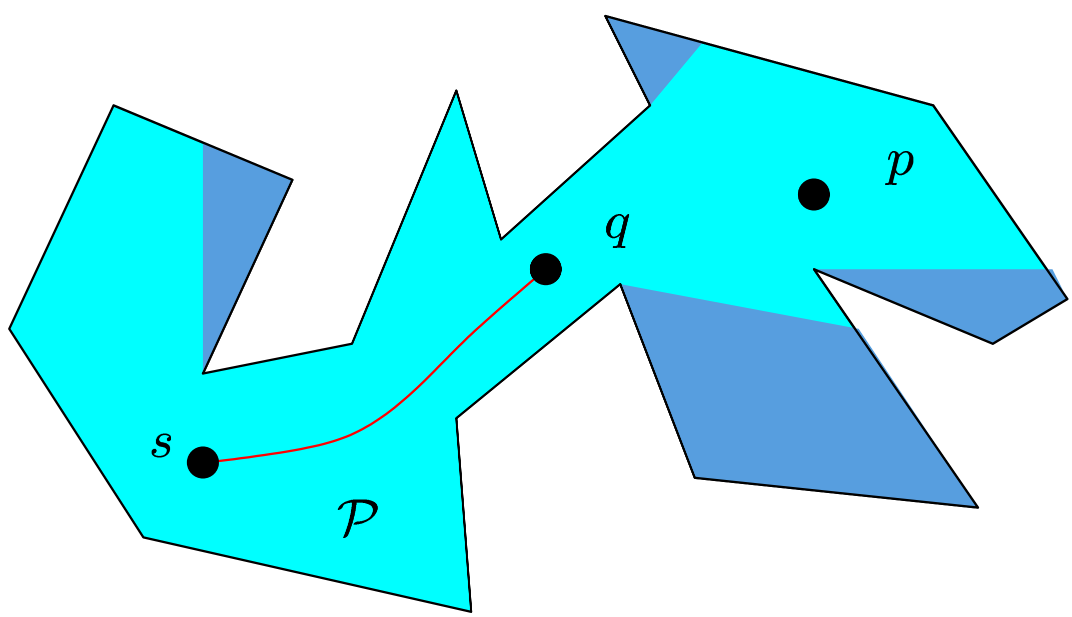
Kürzeste Wege

Lemma

Falls der Roboter den Punkt p gesehen hat, dann kann er den kürzesten Weg $shp(s,q)$ von s nach p berechnen.

Beweis:

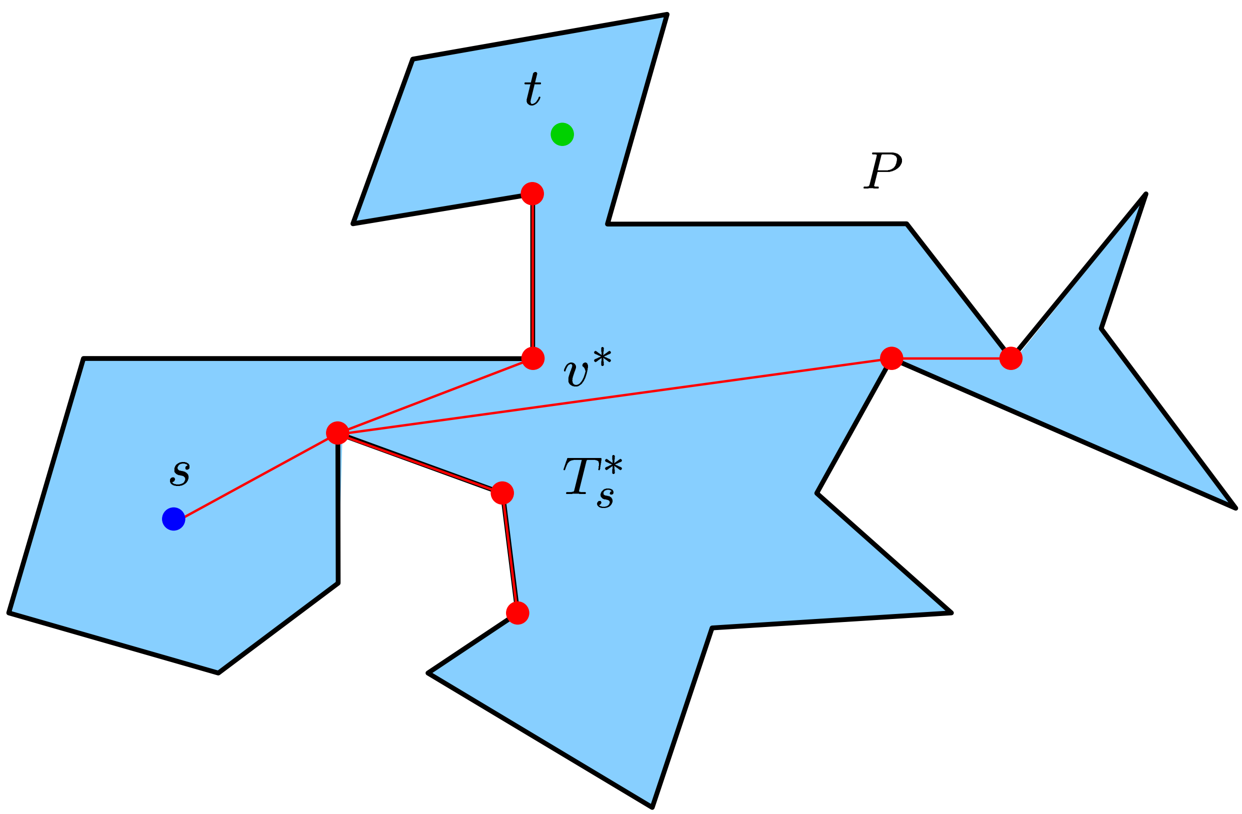
Roboter sieht p am Punkt q
Sei $\mathcal{P}$ der Weg des Roboters von s nach q
Betrachte Sichtbarkeitspolygon von $\mathcal{P}$



Kompetitives Verhältnis

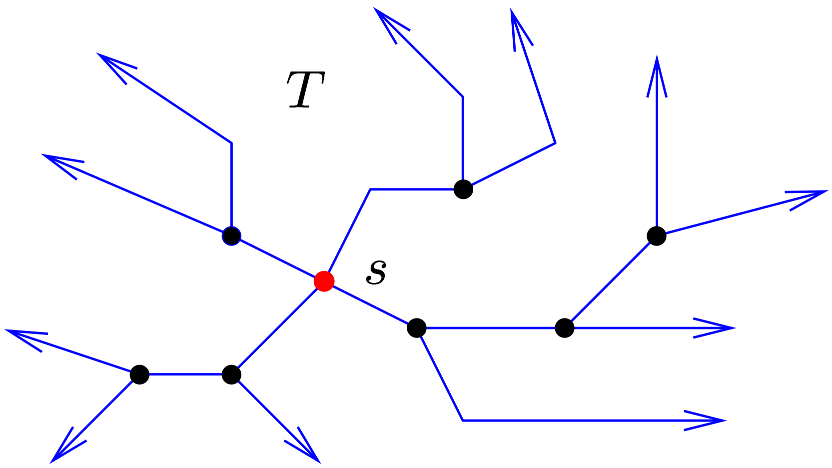
Suche auf dem kürzesten Wege Baum T_s von s

Nur interne Knoten von T_s müssen berücksichtigt werden



Geometrische Bäume

Geometrischer Baum T



m zusammenlaufende Strahlen sind auch ein geometrischer Baum:

$\Rightarrow$ Untere Schranke für die Suche in geometrischen Bäumen

Idee:

T hat m Blätter:

Suche auf den m Wegen von der Wurzel zu den Blättern mit Hilfe der m Wege Suche

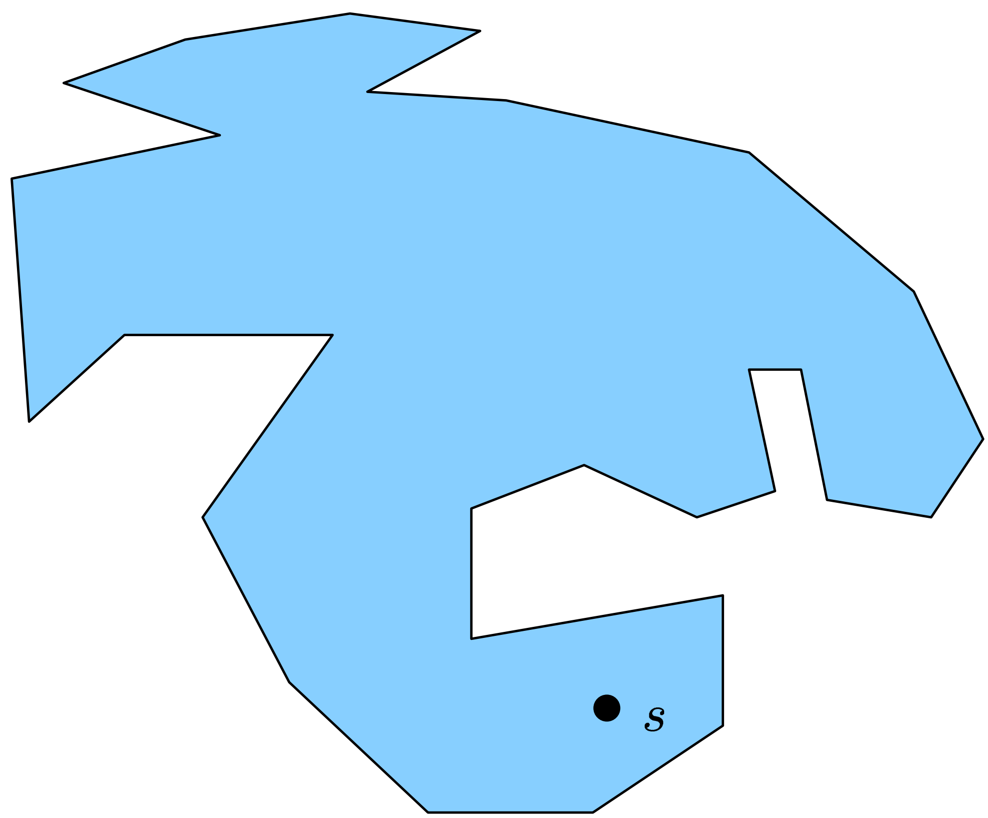
Suchen in einfachen Polygonen

Problem: m ist nicht bekannt

Lösung (Icking 94):

- Verdopple die Suchtiefe in jeder Runde
- halte die Suchtiefe konstant während einer Runde

Zyklische Liste L der kürzesten Wege zu allen Eckpunkten, die schon einmal sichtbar waren

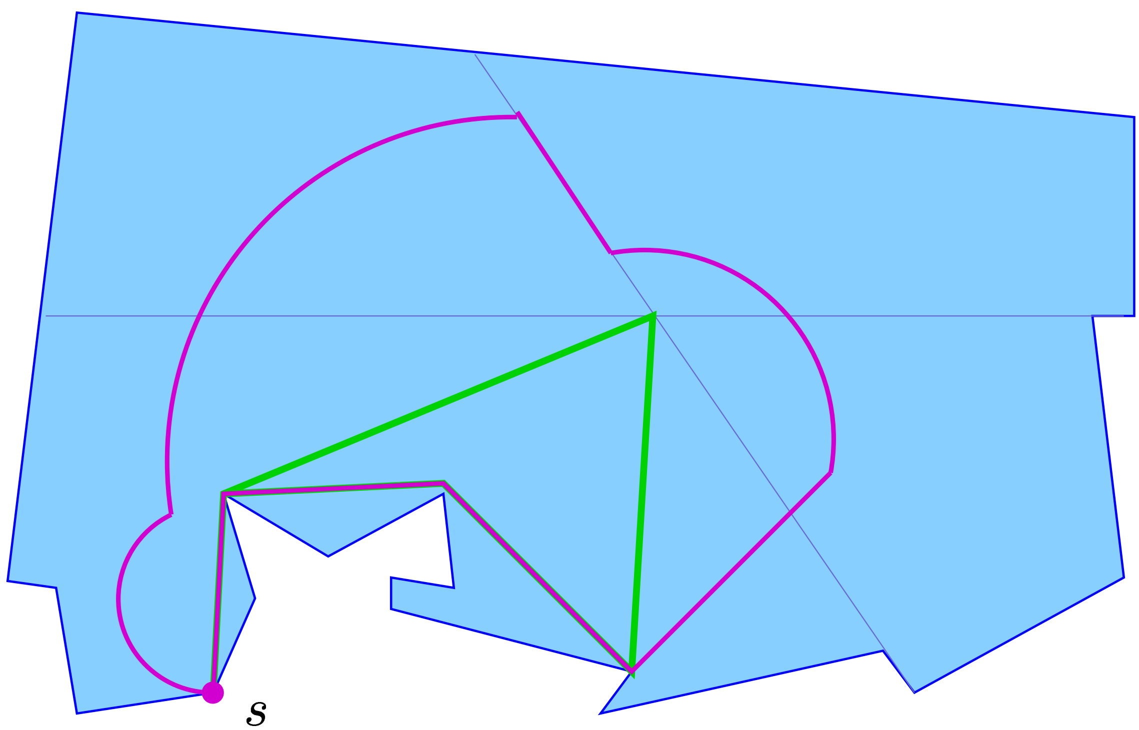


Suchen in einfachen Polygonen

Kompetitives Verhältnis:

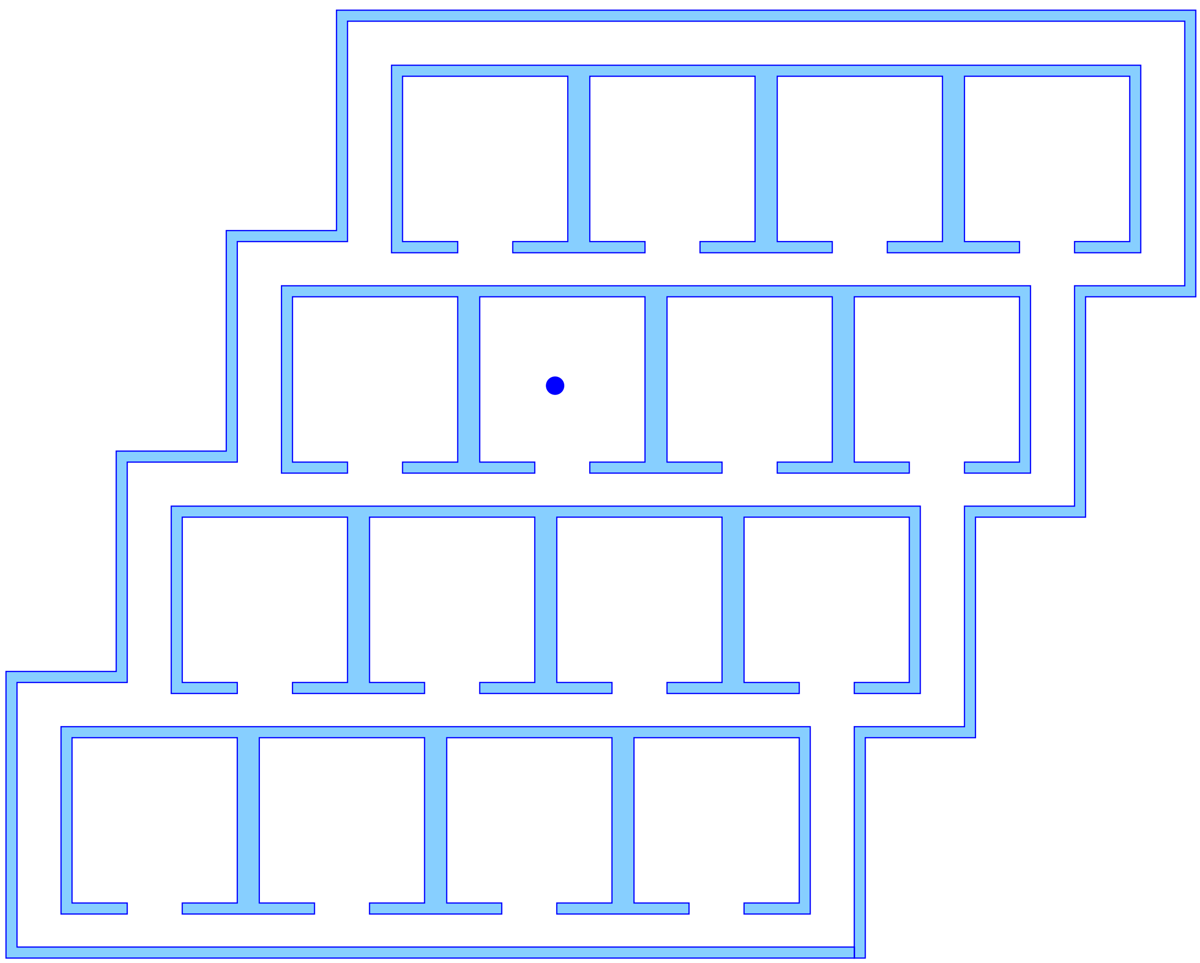
6 Online Navigation—andere Aufgaben

Erkunden der Umgebung (Kartierung)



Online Navigation—andere Aufgaben

Bestimmung des eigenen Standorts (Lokalisation)



Lokalisation