Wegeplanung: Zellunterteilungsverfahren

- Überblick
- Trapezzerlegung
- Approximative Zellunterteilung

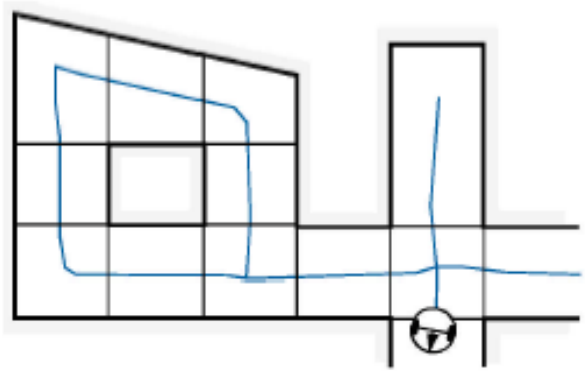
Überblick

Zellunterteilungsverfahren:

- Zerlege den vom Roboter befahrbaren Bereich in Zellen
- ergibt einen Zellnachbarschaftsgraphen

Exakte Zellunterteilung:

- zerlege befahrbaren Bereich in möglichst einfache Flächenstücke (z.B Trapeze)
- Flächenstücke sind im Allgemeinen unterschiedlich groß



Approximative Zellunterteilung:

- zerlege befahrbaren Bereich in Flächenstücke gleicher Form (z.B. Rechtecke)
- Zerlegung ist im Allgemeinen nicht exakt
- Rechtecke können unterschiedliche Größen haben

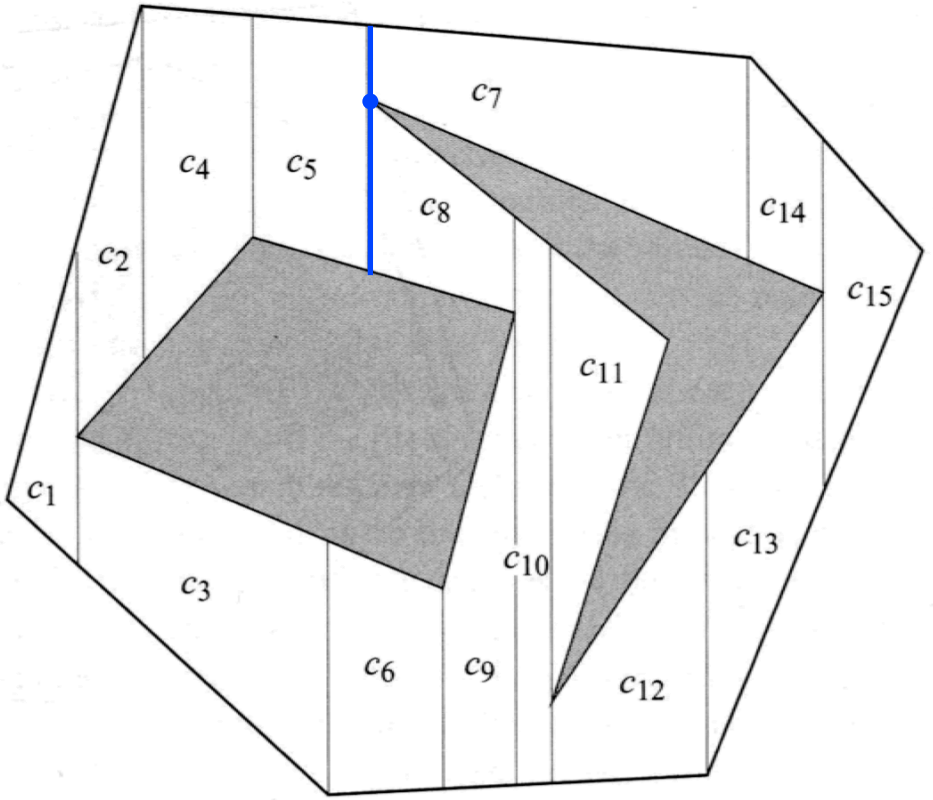
Trapezzerlegung

Polygonale Umgebung

- Hindernisse haben Polygon-Form
- Annahme: Ecken haben unterschiedliche x-Koordinaten

Trapezzerlegung

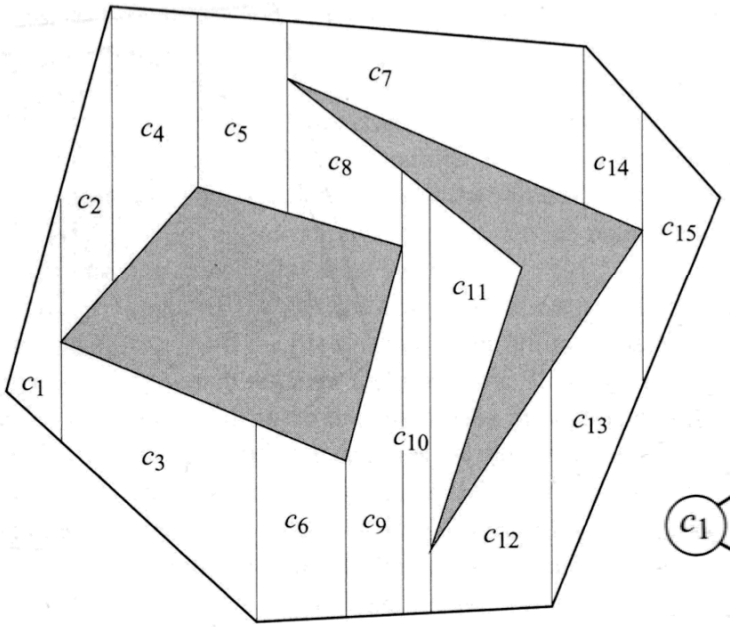
- von jeder Ecke wird
  eine vertikale Linie aufwärts und
  eine vertikale Linie abwärts
  bis zu ersten Hinderniskante
  gezogen
- Es ergeben sich Trapeze.
- Trapeze können zu Dreiecke
  degeniert sein
  (d.h. einer der parallelen Seiten
  hat die Länge 0; z.B. c_{11})

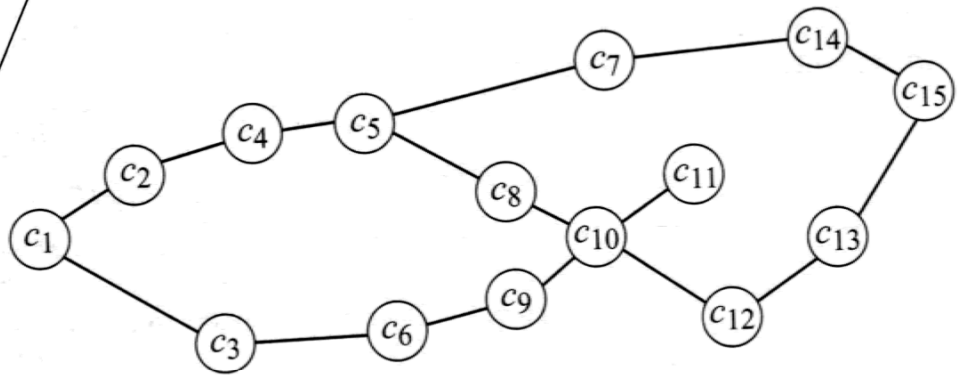


Trapezgraph

Nachbarschaft der Trapeze

- Jedes Trapez hat bis zu 2 linke und bis zu 2 rechte Nachbarn (d.h. gemeinsame vertikale Kante)
- Nachbarschaften ergeben einen Graphen.

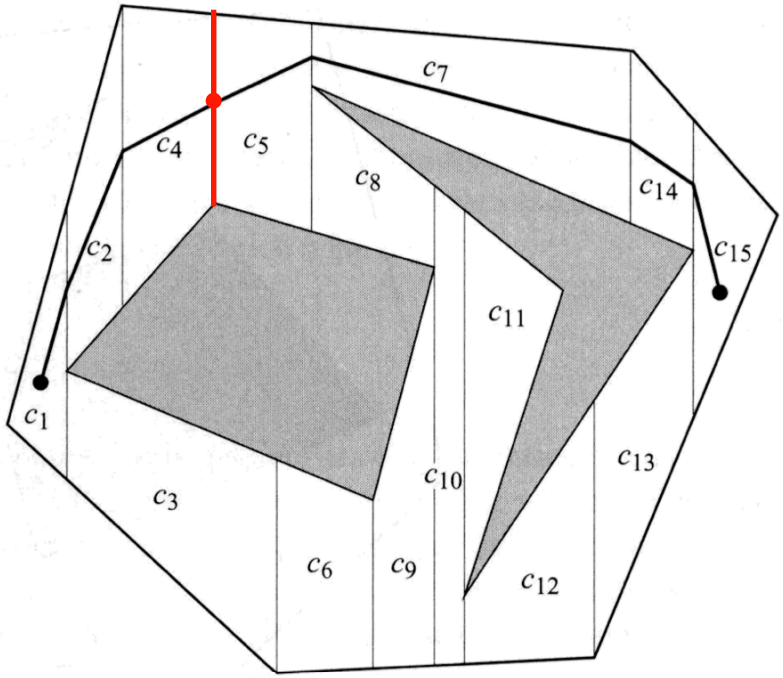


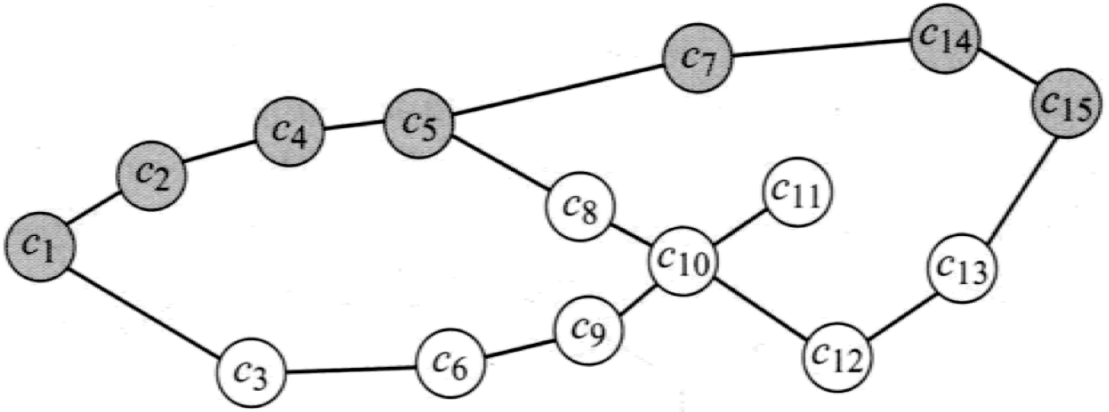


Wegeplanung

Algorithmus:

- Bestimme Trapez, in dem der Start- bzw. der Zielpunkt liegt.
- Suche kürzesten Weg im Trapezgraphen (z.B mit A*)
- Geplanter Weg geht über die Mittelpunkte der vertikalen Trapezkanten

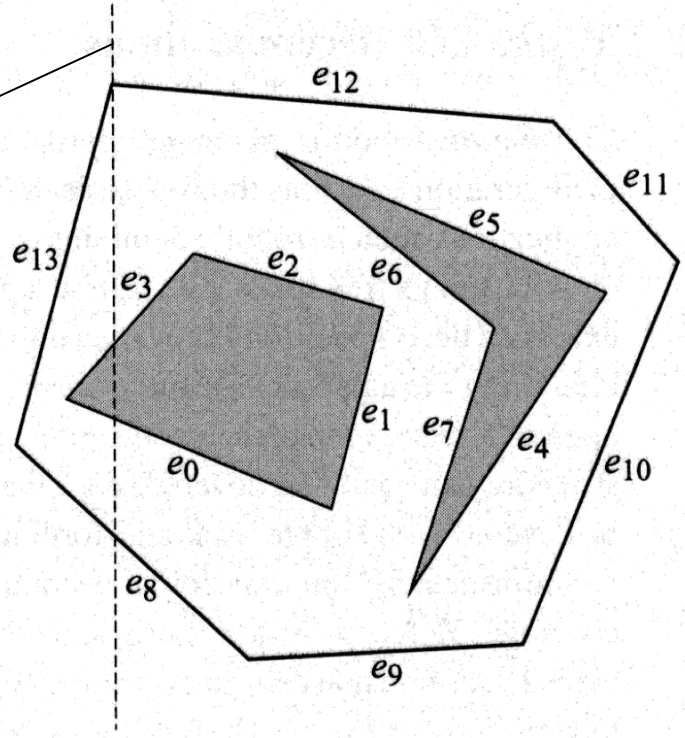




Trapezzerlegung mit Sweep-Line-Verfahren (1)

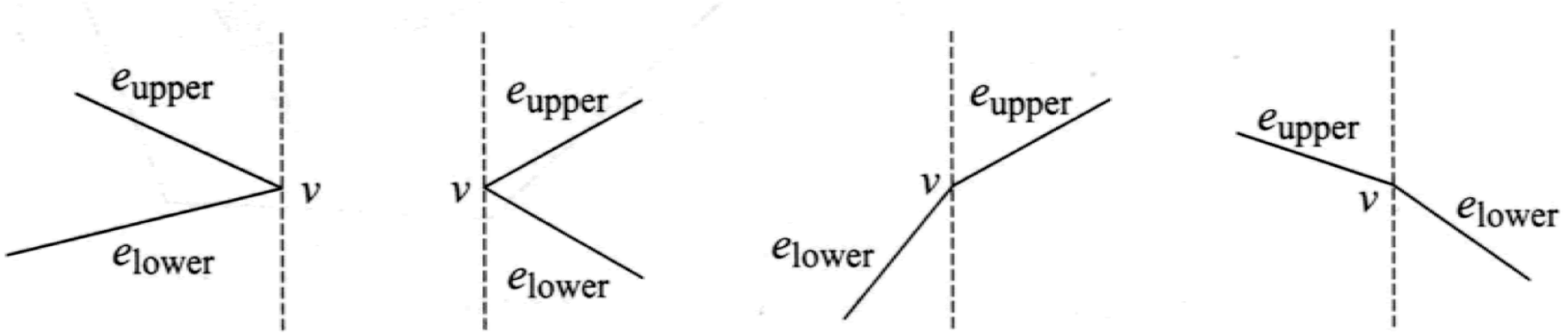
- Sortiere alle Hindernisecken nach der x-Koordinate
- Bewege vertikale Sweep-Line l von links nach rechts über alle Ecken
- Dabei werden alle Hinderniskanten, die l schneiden bzw. deren linke Ecke auf l liegt, in einer effizienten Suchstruktur S (z.B. balanzierter Suchbaum) gespeichert.
- Damit hat das Verfahren die Laufzeit O(n log n), wobei n die Anzahl der Ecken ist.

Sweep-Line ℓ mit den schneidenden Kanten
$S = \{e_8, e_0, e_3, e_{12}\}$

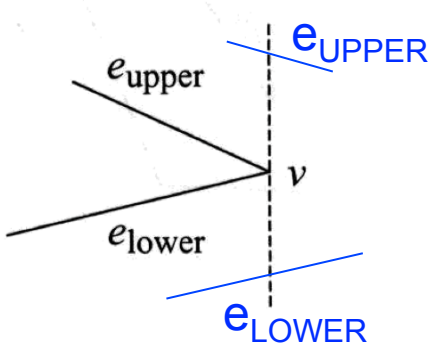


Trapezzerlegung mit Sweep-Line-Verfahren (2)

- Jede Ecke v hat zwei ausgehende Kanten e_{lower} und e_{upper}, wobei eine Kante oberhalb und eine Kante unterhalb von v liegt.
- Sobald die Sweep-Line ℓ auf eine Ecke v trifft, gibt es 4 Möglichkeiten für die Lage von e_{lower} und e_{lupper} :



- Die unmittelbar oberhalb von e_{upper} liegende Kante heiße e_{UPPER} und die unmittelbar unterhalb von e_{lower} liegende Kante heiße e_{LOWER}.
- Mit e_{LOWER} , e_{lower} , e_{upper} und e_{UPPER} sind alle Informationen zur Bildung der Trapeze gegeben.



Trapezzerlegung mit Sweep-Line-Verfahren (3)

Die Suchstruktur S mit den Hinderniskanten, die l schneiden, wird in den 4 Fällen wie folgt verändert:

e_{upper} und e_{lower} liegen links von l:

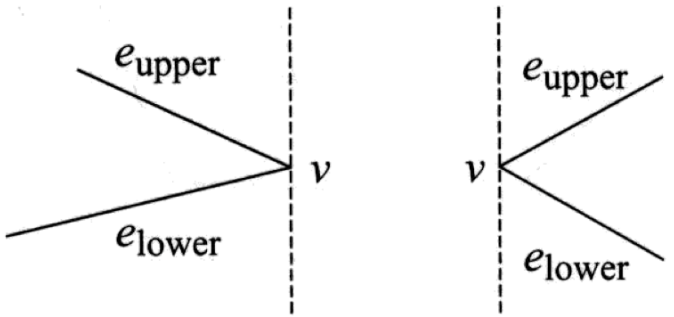
- e_{lower} und e_{upper} werden aus S gelöscht:

  $\{ \ldots, e_{LOWER}, e_{lower}, e_{upper}, e_{UPPER}, \ldots \} \rightarrow \{ \ldots, e_{LOWER}, e_{UPPER}, \ldots \}$

e_{upper} und e_{lower} liegen rechts von l:

- e_{lower} und e_{upper} werden in S eingefügt:

  $\{ \ldots, e_{LOWER}, e_{UPPER}, \ldots \} \rightarrow \{ \ldots, e_{LOWER}, e_{lower}, e_{upper}, e_{UPPER}, \ldots \}$



Trapezzerlegung mit Sweep-Line-Verfahren (4)

e_{lower} liegt links und e_{upper} rechts von ℓ:

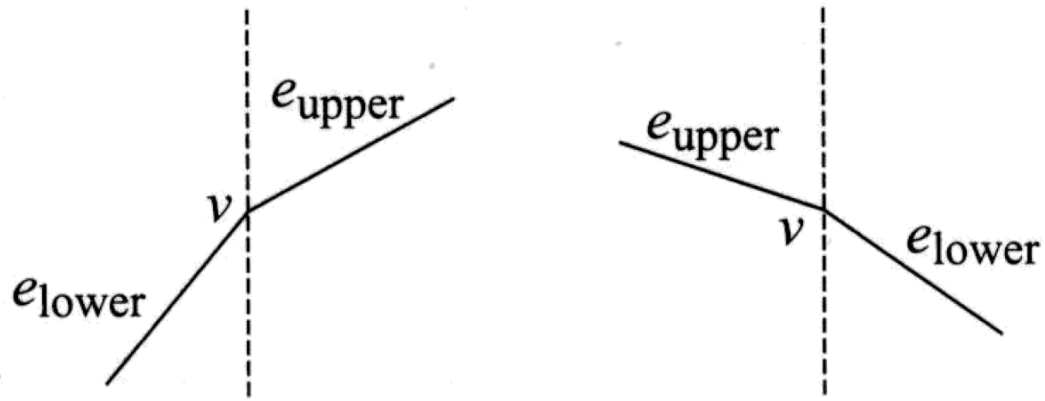
- lösche e_{lower} aus S und füge e_{upper} in S ein:

  $\{ \ldots, e_{LOWER}, e_{lower}, e_{UPPER}, \ldots \} \rightarrow \{ \ldots, e_{LOWER}, e_{upper}, e_{UPPER}, \ldots \}$

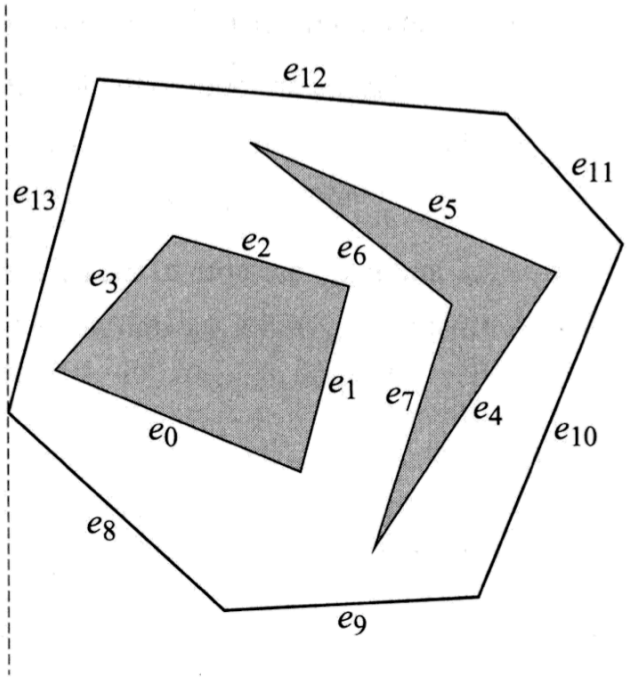
e_{lower} liegt rechts und e_{upper} links von ℓ:

- lösche e_{upper} aus S und füge e_{lower} in S ein:

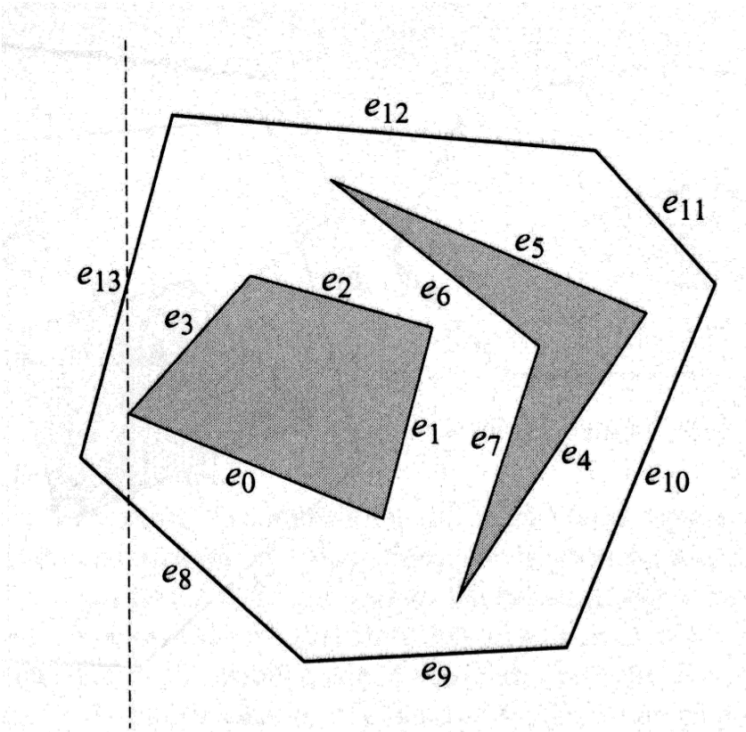
  $\{ \ldots, e_{LOWER}, e_{upper}, e_{UPPER}, \ldots \} \rightarrow \{ \ldots, e_{LOWER}, e_{lower}, e_{UPPER}, \ldots \}$



Beispiel (1)

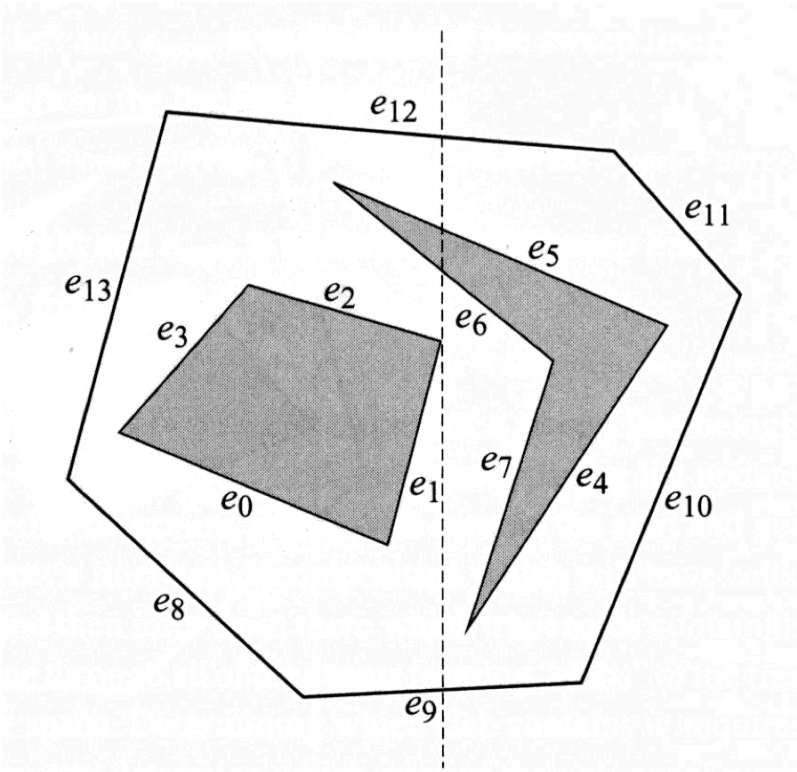


S: { } → {e_8, e_{13}}



S: {e_8, e_{13}} → {e_8, e_0, e_3, e_{13} }

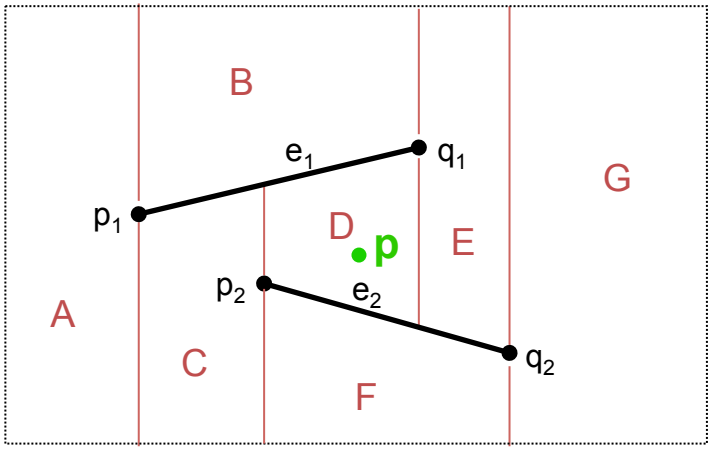
Beispiel (2)



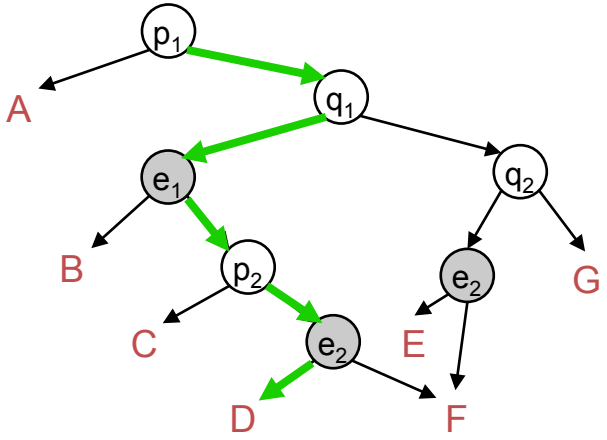
S: $\{e_9, e_1, e_2, e_6, e_5, e_{12}\} \rightarrow \{e_9, e_6, e_5, e_{12}\}$

Punktlokalisierung

- Wie wird zu einem Punkt p (z.B. Start- bzw. Zielpunkt) dasjenige Trapez gefunden, in dem p liegt?
- Dazu spezielle Suchstruktur für Punktlokalisierung aufbauen:
  - azyklischer Suchgraph mit genau einem Blatt für jedes Trapez
  - Suche nach einem Punkt p rekursiv (wie bei binären Suchbäumen):
    - bei Ecken-Knoten: links oder rechts weitersuchen
    - bei Kanten-Knoten: oberhalb oder unterhalb weitersuchen
- Aufbau des azyklischen Suchgraphen in O(n log n).
- Laufzeit für Suchzugriff in O(log n).



Trapezzerlegung



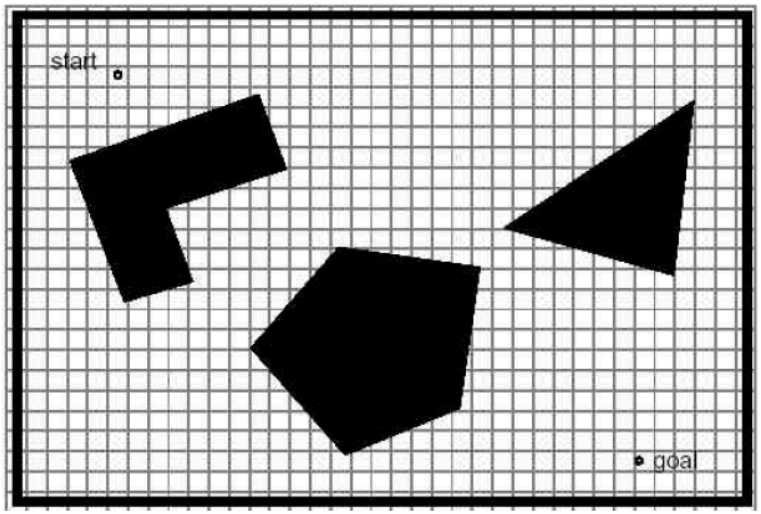
azyklischer Suchgraph für Punktlokalisierung

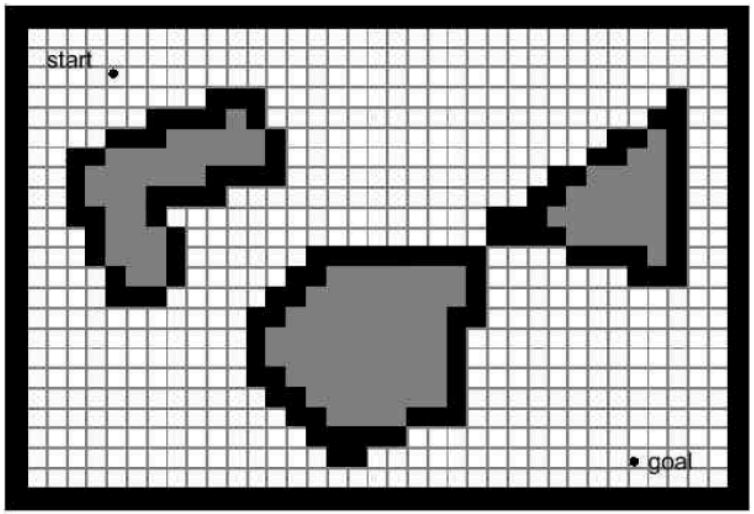
Wegeplanung: Zellunterteilungsverfahren

- Überblick
- Trapezzerlegung
- Approximative Zellunterteilung

Fixe Zellaufteilung

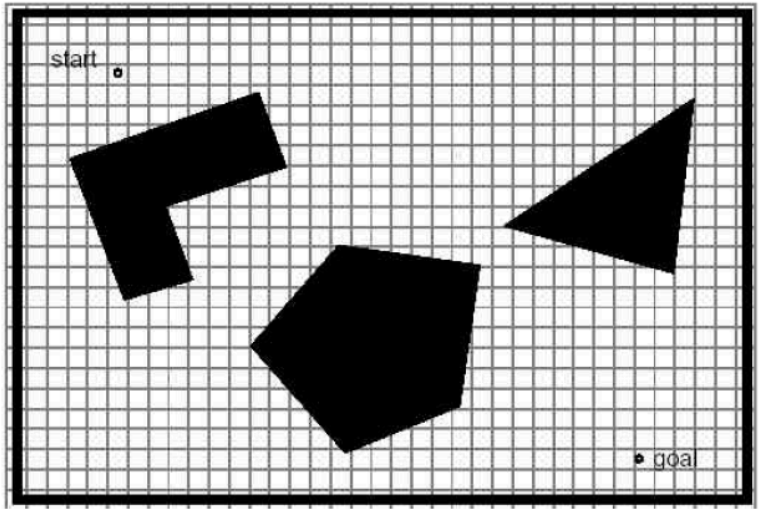
- Quadrate (Rechtecke) haben alle die gleiche Größe
- Enge Passagen können verschwinden

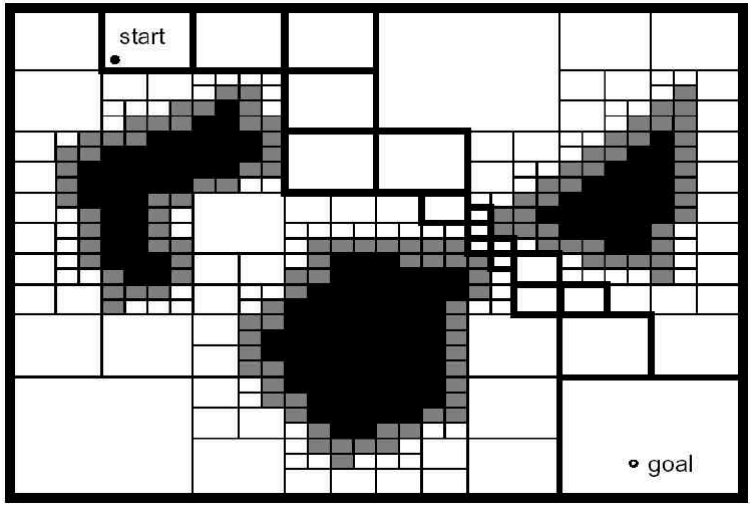




Adaptive Zellaufteilung

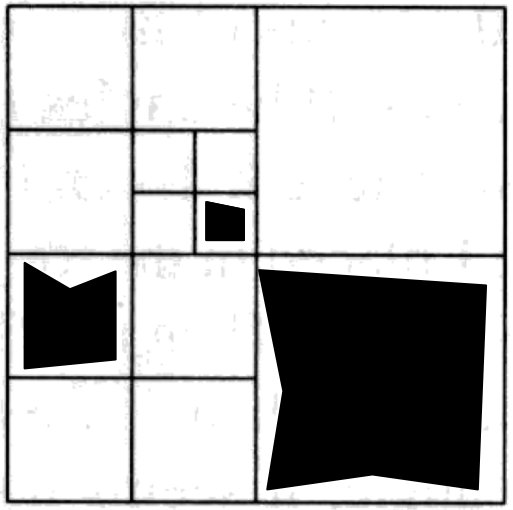
- Quadrate (Rechtecke) können unterschiedliche Größen haben
- Enge Passagen weniger problematisch





Punktlokalisierung mit einem QuadTree

- Adaptive Zellaufteilung



- QuadTree zur effizienten Lokalisierung eines Punktes.
- Suche wie bei binärem Suchbaum