Ostbayerische
Technische Hochschule
Amberg-Weiden

HERZLICH WILLKOMMEN

D*-Algorithmus

Tobias Lettner | Amberg

D*-Algorithmus
Agenda

Ostbayerische Technische Hochschule Amberg-Weiden

1. Grundlegendes
2. Warum D*-Algorithmus?
3. Vorteile D*-Algorithmus
4. Funktionsweise
5. Live-Demo
6. Erweiterungen des D*-Algorithmus

D*-Algorithmus
1. Grundlegendes

Ostbayerische Technische Hochschule Amberg-Weiden

- Anthony Stentz, CMU Pittsburgh
- Veröffentlichung 1994 - The D* Algorithm
- D* (Dynamic A*) - Erweiterung des A*-Algorithmus
- Dynamisch → effiziente Anpassung der Kosten bei Veränderungen im Graphen.

anthony-stentz

Quellen: The D*Algorithm for Real-Time Planning of Optimal Traverses (cmu.edu), D* - Wikipedia

D*-Algorithmus
2. Warum D*-Algorithmus?

Ostbayerische Technische Hochschule Amberg-Weiden

- A* und Dijkstra sind in ihrer Grundform unflexibel
  - Unfähigkeit, auf Veränderungen im Graphen dynamisch zu reagieren
  - Erfordern oft das Verwerfen aller Ergebnisse und einen Neustart

- Realitätsnahe Anwendung im Umgang mit Robotern und Agenten:
  - Entstehung großer Karten/Graphen
  - Kontinuierliche Anpassung und Aktualisierung dieser Karten teils erforderlich

Quellen: The D*Algorithm for Real-Time Planning of Optimal Traverses (cmu.edu), D* - Wikipedia

D*-Algorithmus
3. Vorteile D*-Algorithmus

Ostbayerische Technische Hochschule Amberg-Weiden

- Dynamische Reaktion auf Veränderungen:
  - Ergebnisse müssen nicht verworfen werden.
  - D* kann laufend aktualisiert werden.
  - Bei großen Graphen/Karten deutlich effizienter als A*.
    - nur Teile des Graphen müssen neu berechnet werden.

Quellen: The D*Algorithm for Real-Time Planning of Optimal Traverses (cmu.edu), D* - Wikipedia

D*-Algorithmus
4. Funktionsweise

Ostbayerische Technische Hochschule Amberg-Weiden

Expansion

1. Vom Ziel zum Start (Erinnerung: Bei A* von Start zu Ziel)
    1. Jeder expandierte Knoten zeigt auf den nächsten zum Ziel führenden Knoten (Backpointer)
    2. Jeder expandierte Knoten kennt die exakten Kosten zum Ziel
    3. -› Optimaler Pfad für jeden möglichen Startknoten

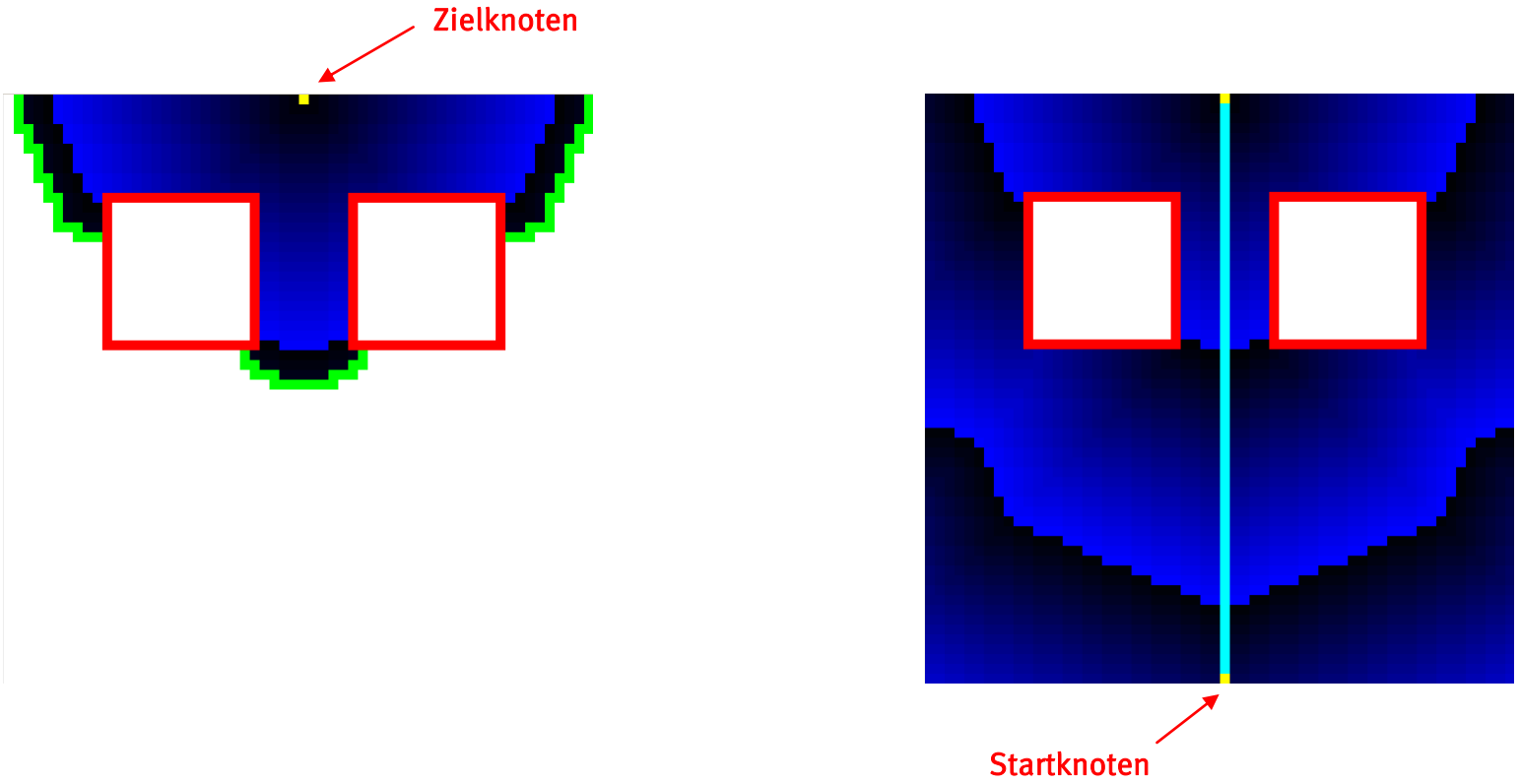
2. Wird der Startknoten expandiert, kann der Algorithmus abgebrochen werden.
    1. Pfad ergibt sich durch Rückverfolgung der Backpointer

Quellen: The D*Algorithm for Real-Time Planning of Optimal Traverses (cmu.edu), D* - Wikipedia

D*-Algorithmus
4. Funktionsweise

Ostbayerische Technische Hochschule Amberg-Weiden

Expansion



Quellen: The D*Algorithm for Real-Time Planning of Optimal Traverses (cmu.edu), D* - Wikipedia

D*-Algorithmus
4. Funktionsweise

Ostbayerische Technische Hochschule Amberg-Weiden

Handhabung bei Hindernissen – RAISE Welle

1. Hindernis tritt auf:
    1. Betroffenen Knoten werden in die Open List aufgenommen
    2. Betroffene Knoten werden als RAISE gekennzeichnet.
2. Vor Kostenanpassung, werden rekursiv die Nachbarn überprüft, ob die Kosten der Knoten reduziert werden können.
3. Falls nein propagiert der RAISE Zustand zu allen Nachbarknoten. (Knoten mit Backpointer)

Diese wird als **RAISE Welle** bezeichnet.

Quellen: The D*Algorithm for Real-Time Planning of Optimal Traverses (cmu.edu), D* - Wikipedia

D*-Algorithmus
4. Funktionsweise

Ostbayerische Technische Hochschule Amberg-Weiden

Handhabung bei Hindernissen – LOWER Welle

1. Wenn in der RAISE Welle ein Knoten gefunden wird, der die Kosten reduzieren kann, wird der Zustand des Knoten auf LOWER gesetzt.
2. Dies löst die **LOWER Welle** aus
    1. Kosten werden aktualisiert
    2. Backpointer werden aktualisiert

RAISE und LOWER Zustände sind das Herzstück des D*-Algorithmus

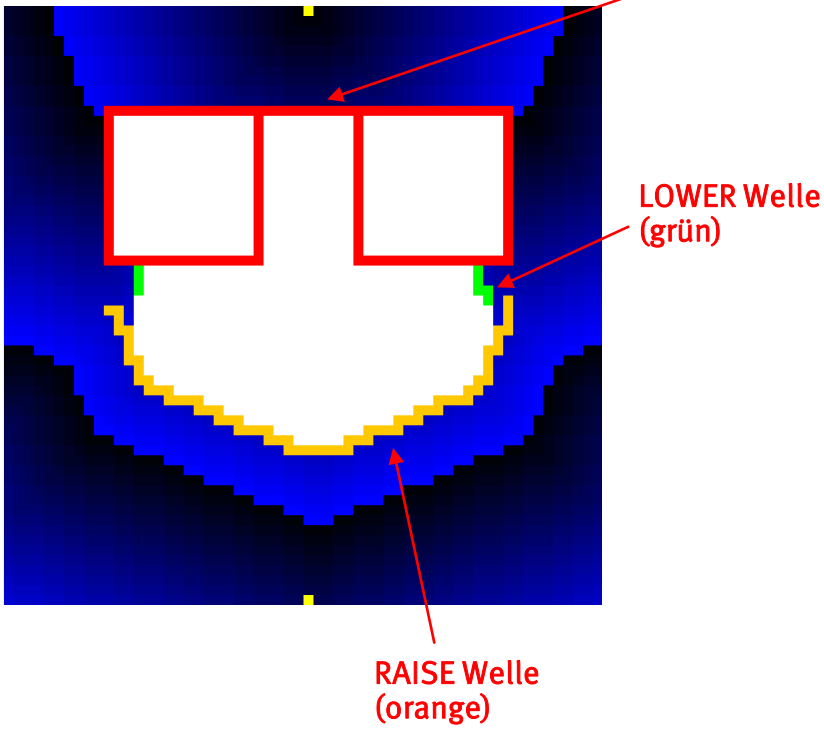
→ Es müssen nur die Knoten, die von der Welle betroffen sind aktualisiert werden.

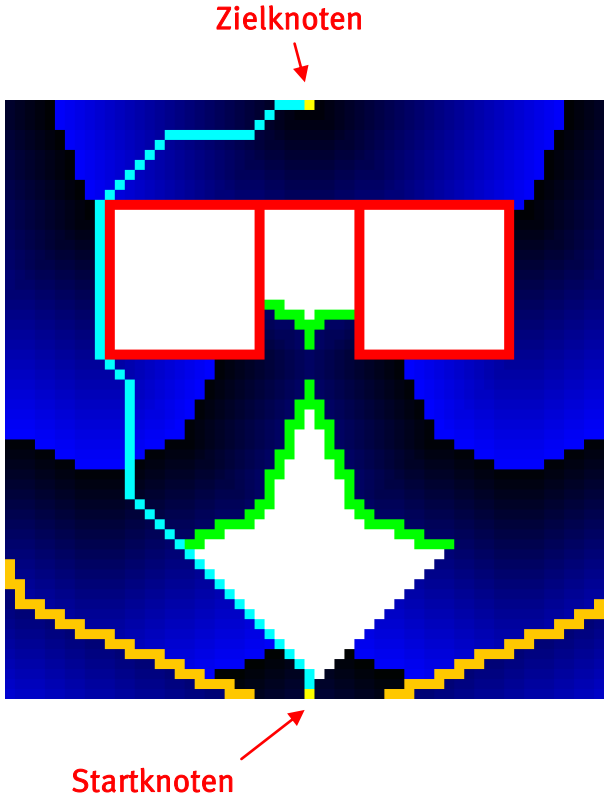
Quellen: The D*Algorithm for Real-Time Planning of Optimal Traverses (cmu.edu), D* - Wikipedia

D*-Algorithmus
4. Funktionsweise

Ostbayerische Technische Hochschule Amberg-Weiden

Handhabung bei Hindernissen





Quellen: The D*Algorithm for Real-Time Planning of Optimal Traverses (cmu.edu), D* - Wikipedia

D*-Algorithmus
5. Live-Demo

Ostbayerische Technische Hochschule Amberg-Weiden

Live-Demo

Quellen: The D*Algorithm for Real-Time Planning of Optimal Traverses (cmu.edu), D* - Wikipedia

D*-Algorithmus
6. Erweiterungen des D*-Algorithmus

Ostbayerische Technische Hochschule Amberg-Weiden

- Focused D*
  - Basiert auf dem D*-Algorithmus
  - Verwendet Heuristik, um die Ausbreitung von RAISE und LOWER auf den Roboter zu fokussieren.
  - Auf diese Weise werden nur relevante Zustände aktualisiert.

- D* Lite
  - Basiert nicht auf dem D*-Algorithmus.
  - Implementiert aber das gleiche Verhalten, jedoch in weniger Codezeilen und auf einfachere Weise.
  - Grundlage ist das Lifelong Planning A* .

Quellen: The D*Algorithm for Real-Time Planning of Optimal Traverses (cmu.edu), D* - Wikipedia

Ostbayerische
Technische Hochschule
Amberg-Weiden

VIELEN DANK!