

Aufgabe 1

„Lisa rennt“- Dokumentation

37. Bundeswettbewerb Informatik 2018/19 - 2. Runde

Lukas Rost

Teilnahme-ID: 48125

29. April 2018

Inhaltsverzeichnis

1	Lösungsidee	1
1.1	Das Geometric-Shortest-Path-Problem	1
1.2	Erzeugung eines Sichtbarkeitsgraphen	2
1.3	Der Dijkstra-Algorithmus	2
1.4	Lösung des Problems ohne Hindernisse	2
1.5	Kombination der Ansätze	2
1.6	Laufzeitbetrachtung	2
2	Umsetzung	2
2.1	Allgemeine Hinweise zur Benutzung	2
2.2	Struktur des Programms	2
2.3	Implementierung der wichtigsten Algorithmen	2
3	Beispiele	2
3.1	Beispiel 1	2
3.2	Beispiel 2	2
3.3	Beispiel 3	2
3.4	Beispiel 4	2
3.5	Beispiel 5	2
3.6	Eigene Beispiele	2
4	Quellcode	2

1 Lösungsidee

1.1 Das Geometric-Shortest-Path-Problem

Das der Aufgabe zugrundeliegende Problem nennt sich *Geometric Shortest Path* (GSP), auf Deutsch auch bekannt als Problem des geometrischen kürzesten Weges. Bei diesem Problem ist ein punktförmiger Roboter (oder auch eine Schülerin namens Lisa) gegeben, der/die sich an einer Startposition p_{start} in einem kartesischen Koordinatensystem befindet. In diesem Koordinatensystem befinden sich mehrere als Polygone modellierte Hindernisse, wobei jedes einzelne Polygon durch seine Eckpunkte gegeben ist. Weiterhin ist eine Position p_{ziel} gegeben. Nun soll ein möglichst kurzer Weg von p_{start} zu p_{ziel} gefunden werden, wobei dieser nicht durch Hindernisse führen soll.¹[2]

Das hier gegebene Problem unterscheidet sich von GSP dadurch, dass keine Position p_{ziel} , sondern eine Gerade g_{ziel} (in Form eines beliebigen Punktes auf der Geraden) erreicht werden soll. Zusätzlich soll nicht unbedingt der Weg optimiert werden, sondern die Startzeit, die abhängig von der Länge des Weges und der für jeden Punkt der Geraden unterschiedlichen Zielzeit ist. [TODO labern bzgl. Problemstellung]

Das Geometric-Shortest-Path-Problem wird im Allgemeinen in zwei Schritten gelöst: Zunächst wird ein Sichtbarkeitsgraph erstellt, auf welchem dann Dijkstras Algorithmus ausgeführt wird. Zur Lösung des hier gegebenen Problems wird jedoch ein zusätzlicher Schritt benötigt, der auf der Lösung des Problems ohne Hindernisse basiert. Diese drei Algorithmen sollen in den folgenden Abschnitten vorgestellt und näher erläutert werden.

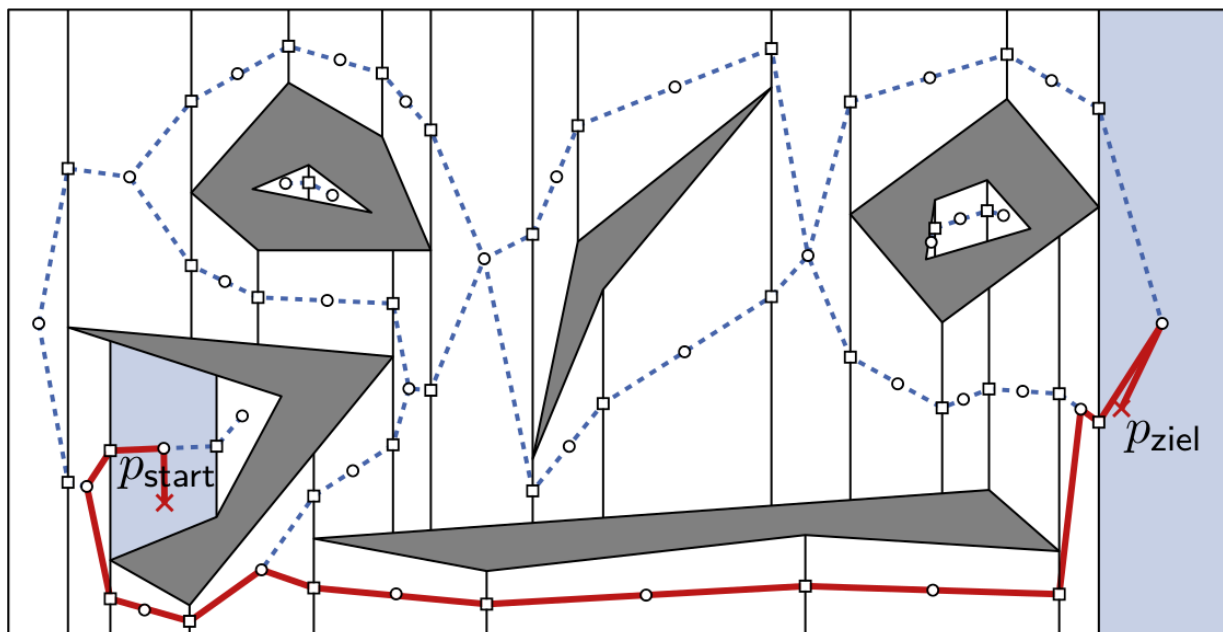


Abbildung 1: Illustration des GSP-Problems (aus [2])

¹Zumindest nicht, wenn Lisa als Zielposition nicht das Krankenhaus erreichen will.

1.2 Erzeugung eines Sichtbarkeitsgraphen

1.3 Der Dijkstra-Algorithmus

1.4 Lösung des Problems ohne Hindernisse

1.5 Kombination der Ansätze

1.6 Laufzeitbetrachtung

Literatur

- [1] Bittel, O. (HTWG Konstanz): Autonome Roboter - Wegekartenverfahren, SS 2016 (Präsentation)
- [2] Nöllenburg, Martin (KIT): Vorlesung Algorithmische Geometrie - Sichtbarkeitsgraphen, 2011 (Präsentation)
- [3] Reksten-Monsen, Christian: Distance Tables Part 1 - Defining the Problem, <https://taipanrex.github.io/2016/09/17/Distance-Tables-Part-1-Defining-the-Problem.html>
- [4] Reksten-Monsen, Christian: Distance Tables Part 2: Lee's Visibility Graph Algorithm, <https://taipanrex.github.io/2016/10/19/Distance-Tables-Part-2-Lees-Visibility-Graph-Algorithm.html>

2 Umsetzung

2.1 Allgemeine Hinweise zur Benutzung

2.2 Struktur des Programms

2.3 Implementierung der wichtigsten Algorithmen

3 Beispiele

3.1 Beispiel 1

3.2 Beispiel 2

3.3 Beispiel 3

3.4 Beispiel 4

3.5 Beispiel 5

3.6 Eigene Beispiele

4 Quellcode