

# PROSEMINAR ONLINE ALGORITHMEN

im Wintersemester 2000/2001

Prof. Dr. Rolf Klein, Dr. Elmar Langetepe, Dipl. Inform. Thomas Kamphans (Betreuer)

Vortrag vom 15.11.2000 von Jan Schmitt

Thema : Finden eines Zielpunktes in unbekannter Umgebung

## Inhaltsübersicht :

1. Vorstellung des Problems
2. Modell des Roboters und der Umgebung
3. untere Schranke für beliebige Strategien
4. die Strategie „Bug1“
5. Kompetitivität der Strategie „Bug1“

## 1. Vorstellung des Problems

Ein Roboter soll in einer unbekannten Umgebung mit Hindernissen einen fest vorgegebenen Zielpunkt finden und die Hindernisse dabei umfahren. Dies soll online geschehen, d.h. der Roboter kennt vorher weder die Form noch die Lage der Hindernisse, sondern diese Informationen werden erst auf dem Weg zum Ziel erkundet, dazu später mehr. Bekannt sind dem Roboter jedoch stets sein eigener Standpunkt und die Lage des Zielpunktes. Im

Unterschied zu vielen anderen Onlinealgorithmen sind die Objekte also hier nicht nur zeitlich, sondern auch räumlich getrennt.

## 2. Modell des Roboters und der Umgebung

Um mit dem Problem besser umgehen zu können, werden zur Vereinfachung einige Voraussetzungen gemacht, die in der Realität auch oft gegeben sind:

Die Umgebung (ohne die Hindernisse) soll eine zweidimensionale, ebene Fläche sein. Die Hindernisse haben eine beliebige Form, jedoch eine gewisse Mindesthöhe, so dass sichergestellt ist dass der Roboter sie mit seinen Sensoren (siehe unten) erfassen kann. Außerdem haben die Hindernisse einen endlichen Umfang und stetige Umrandungen. Je zwei Hindernisse sollen sich nicht berühren, d.h. ein beliebiger Punkt auf einem Hindernis gehört genau zu diesem Hindernis und liegt auf keinem anderen Hindernis. Des weiteren wird ein Koordinatensystem zugrundegelegt. Dies kann ein beliebiges Koordinatensystem sein, wie z.B. lokale Koordinaten einer Halle, Gauss-Krüger-Koordinaten, oder sonstige Weltkoordinaten die über Satellitenpeilung bestimmt werden können. In diesem Koordinatensystem sollen von Beginn an sowohl Start- als auch Zielpunkt fest vorgegeben sein.

Der Roboter selbst wird dabei als Punktförmig angenommen, d.h. insbesondere er kann immer zwischen je zwei beliebigen Hindernissen hindurchnavigieren. Ferner wird vorausgesetzt dass der Roboter stets sowohl seine eigenen Koordinaten als auch die Koordinaten des Zielpunktes kennt und über einen Speicher verfügt, mit dem er sich Positionen merken kann. Außerdem soll der Roboter so ausgestattet sein dass er aus beliebigen aktuellen Koordinaten oder Koordinaten aus dem Speicher Entfernungen berechnen kann. Er verfügt über einen Tastsensor, womit er bemerkt wenn er ein Hindernis berührt und mit Hilfe dessen er auch an einem Hindernis entlang fahren kann. Da es sich um ein „Onlineproblem“ handelt, hat der Roboter außer den Informationen im Speicher nur Informationen über das aktuelle Objekt das er grade berührt. Deshalb kann er auch, wenn er auf ein neues Hindernis stößt, zunächst nicht entscheiden ob es günstiger ist zuerst nach rechts oder nach links zu gehen. Diesbezüglich findet auch kein Lernprozess statt, der Roboter sammelt also keinerlei Erfahrungen über die Umgebung oder zieht aus solchen Rückschlüsse auf die Lage oder Form noch nicht erkundeter Hindernisse. Weiterhin muss der Roboter natürlich navigieren können. Dazu genügt es wenn er 3 Grundbefehle ausführen kann: Als erstes soll er Roboter geradlinig in Richtung Zielpunkt navigieren können, zweitens an einer Hinderniswand entlang fahren, und zwar sowohl links- als auch rechtsläufig, und drittens schließlich stoppen wenn er das Ziel erreicht hat bzw. festgestellt hat dass es keinen freien Weg zum Zielpunkt gibt.

## 3. untere Schranke für beliebige Strategien

In diesem Abschnitt wird folgendes interessante Theorem betrachtet und bewiesen :

Theorem : Zu jedem Algorithmus, bezogen auf die vorherigen Bedingungen, gibt es eine Umgebung, so dass zu jeder noch so großen Weglänge  $P$ , jedem noch so kleinen Abstand  $|ST|$  des Startpunktes  $S$  vom Zielpunkt  $T$  und jedem noch so kleinen  $\delta > 0$  gilt :

$$P \geq |ST| + \sum_i P(i) - \delta ,$$

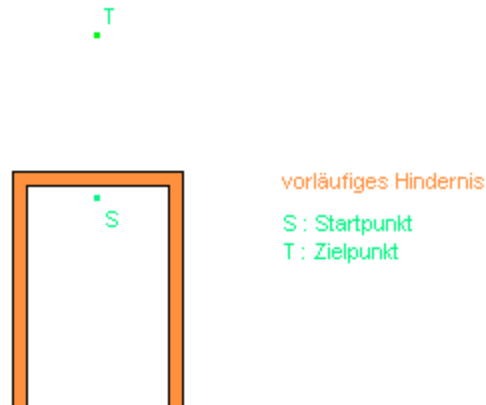
wobei  $P(i)$  den Umfang des  $i$ -ten Objekts bezeichnet.

Dabei kann man das Modell sogar insofern verallgemeinern, als dass man auch andere Bewegungen des Roboters als die oben genannten zulässt. So lässt sich das Theorem auch beweisen, wenn der Roboter sich z.B. auch vom Zielpunkt wegbewegen kann ohne gerade eine Hinderniswand zu berühren.

Beweisidee : Konstruktiver Beweis, d.h. wir versuchen, zu einem beliebigen Algorithmus unter obigen Bedingungen eine Umgebung zu konstruieren, so dass die Ungleichung aus Theorem 1 erfüllt ist.

Beweis :

Wir betrachten zunächst die „vorläufige“ Umgebung :



(Abb. 1)

In Abbildung 1 stellt S den Startpunkt und T (von englisch target) den Zielpunkt dar, das Hindernis ist in rot dargestellt. Die Wandstärke des Hindernisses wird als vernachlässigbar gering angenommen.

Zum Beweis lassen wir den Roboter nun mit einem beliebigen Algorithmus X vom Startpunkt S zum Zielpunkt T laufen, und erhalten dann die „endgültige“ Umgebung. Um diese zu konstruieren, werden aus der „vorläufigen“ Umgebung genau die Teile des Hindernisses übernommen die der Roboter unter dem Algorithmus X tatsächlich berührt hat, und alle anderen Teile werden entfernt: Jedes Teilstück, an dem der Roboter ein Stück entlanggefahren ist wird so in die endgültige Umgebung übernommen, und wenn der Roboter auf seinem Weg das Hindernis an einer Stelle nur berührt und dann sofort wieder verlassen hat, so wird an dieser Stelle ein Stück des vorläufigen Hindernisses der Breite  $\delta$  in die endgültige Umgebung übernommen. Man stellt leicht fest dass der Roboter unter Algorithmus X dann in der endgültigen Umgebung genau den gleichen Weg vom Start- zum Zielpunkt zurücklegt wie in der vorläufigen Umgebung. Denn genau die Teile des vorläufigen Hindernisses die der Roboter nicht berührt hat wurden ja entfernt, und diese haben, eben weil der Roboter sie gar nicht berührt, auch keinen Einfluss auf das Verhalten des Roboters.

Sofort ersichtlich ist zunächst dass der zurückgelegte Weg nicht kürzer sein kann als  $|ST|$ , da dies gerade die Entfernung vom Start- zum Zielpunkt ist. Zum Beweis des Theorems unterscheiden wir 3 Fälle :

1. Fall : Der Roboter hat auf seinem Weg vom Start- zum Zielpunkt unter Algorithmus X das vorläufige Hindernis gar nicht berührt. Also ist die endgültige Umgebung frei von Hindernissen und somit  $\sum_i P(i) = 0$  und der insgesamt zurückgelegte Weg  $P > |ST|$ , und somit ist die zu beweisende Ungleichung erfüllt.
2. Fall : Der Roboter ist auf seinem Weg vom Start- zum Zielpunkt unter Algorithmus X genau einmal auf das vorläufige Hindernis gestoßen. Dann enthält die endgültige Umgebung genau ein Hindernis. Da der Roboter einen Weg der Länge dieses Hindernisses auf dem Weg vom Ziel weg zurückgelegt hat, muss er, wenn er wieder in

vorläufiges Hindernis

endgültiges Hindernis

Abbildung 2

Somit ist die Ungleichung erfüllt.

- #### 4. die Strategie „Bug 1“

```

Repeat
  Repeat
    Laufe geradlinig auf Zielpkt. zu
  Until Wandkontakt
  A := Aktuelle Position ( auf Hinderniswand )
  D := Aktuelle Position ( bisher dem Zielpunkt nächster besuchter Punkt auf Hindernis )
  Repeat
    Rücke Aktuelle Position entlang Hindernis vor;
    If Aktuelle Position näher zum Zielpkt. als D
      Then D := Aktuelle Position
  Until Aktuelle Position = A;
  Gehe auf kürzestem Weg entlang Hindernis zu D
Until Zielpunkt erreicht

```

Dem ist hinzuzufügen, dass der Roboter, jedes Mal wenn er auf ein Hindernis trifft, in dem Beispielen zuerst nach links geht, da er ja nicht entscheiden kann welche der beiden

Möglichkeiten günstiger ist. Wenn der Roboter ein Hindernis auf dem Weg nur tangiert so ignoriert er dies einfach und fährt weiter. Abb. 3 zeigt ein Beispiel zur Arbeitsweise der Strategie Bug1 :

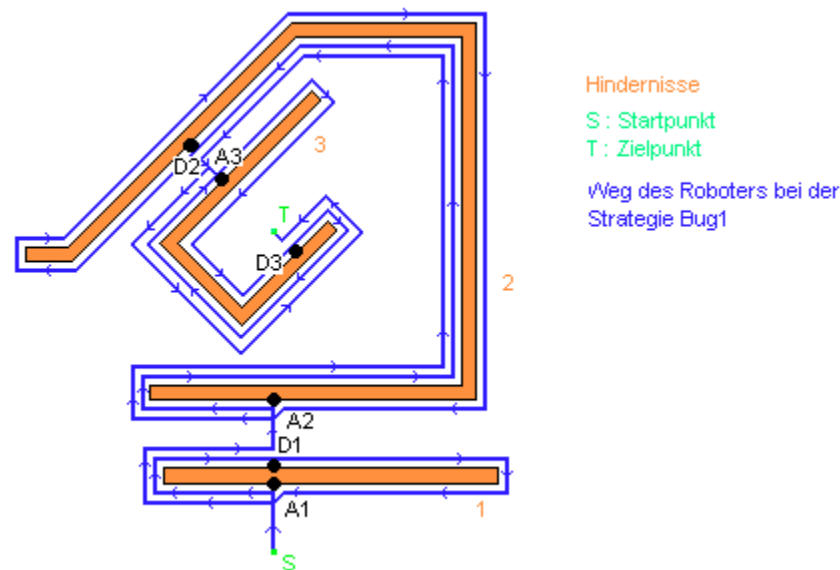


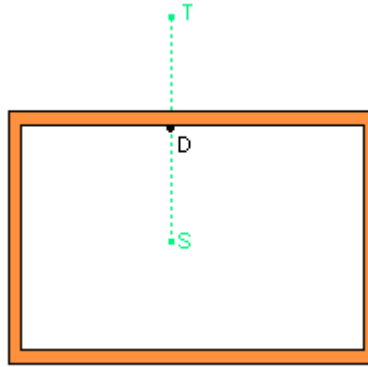
Abbildung 3

Dabei führt der Weg natürlich stets wie oben formuliert entweder geradlinig auf den Zielpunkt zu oder aber direkt an einer Hinderniswand entlang. Die Punkte A1...A3 bzw. D1...D3 in der Abbildung stellen die Punkte A und D am Ende der jeweiligen Schleifendurchläufen dar. Nun sollte die Arbeitsweise von Bug1 klar sein und wir betrachten einige Eigenschaften des Algorithmus :

- Wenn der Roboter das Ziel T vom Startpunkt S aus erreicht und dabei  $i$  Hindernisse vom Umfang  $U(i)$  umfahren hat, hat er höchstens einen Weg der Länge  $|ST| + 3/2 \sum_{i=1}^n U(i)$

zurückgelegt. Beweis : Die Summe aller vom Roboter zurückgelegten Teilstrecken von  $D(i)$  nach  $A(i+1)$  ist insgesamt keinesfalls länger als die Gesamtstrecke  $|ST|$  . Außerdem wird jedes getroffene Hindernis genau einmal und dann noch einmal höchstens zur Hälfte umrundet, was den Faktor  $1,5 = 3/2$  ergibt.

- Auf jedem Hindernis gibt es genau einen Punkt an dem dieses angefahren wird.
- Auf jedem Hindernis gibt es genau einen Punkt an dem dieses wieder verlassen wird. Existieren auf einem Hindernis mehrere dem Ziel nächste Punkte, so wird ein beliebiger ausgewählt.
- jedes Hindernis wird höchstens einmal umfahren. Wenn ein Hindernis einmal verlassen wurde, so wird dieses nie wieder berührt, da der Roboter ja von einem dem Zielpunkt nächsten Punkt auf der Hinderniswand aus ein Stück geradlinig in Richtung Zielpunkt gefahren ist.
- erreicht der Roboter einen Punkt D und kann von dort nicht weiter in Richtung Zielpunkt fahren, so ist das Ziel nicht erreichbar. Mindestens Zielpunkt oder Roboter sind komplett von einem Hindernis eingeschlossen, da der Roboter den eigentlich dem Ziel am nächsten gelegenen Punkt auf der Hinderniswand nicht erreichen kann (siehe Abbildung) ,

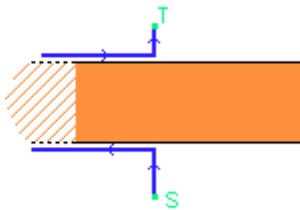


bei eingeschlossenem Zielpunkt entsprechend.

- Bug1 wird in der Praxis nicht eingesetzt, da es Verbesserungen des Algorithmus gibt (z.B. Bug2), die i.A. mit kürzeren Wegen zum Zielpunkt finden.

## 5. Kompetitivität der Strategie „Bug1“

Zur Erinnerung zunächst noch einmal die Definition der Kompetitivität : Ein Onlinealgorithmus mit Kostenfunktion  $K$  heißt  $c$ -kompetitiv ( zum optimalen Offlinealgorithmus ), wenn es eine Konstante  $c$  gibt, so dass  $K(\text{onlinealgorithmus}) \leq c K(\text{optimaler Algorithmus}) + a$ , wobei  $a$  eine von der Anfrage unabhängige Konstante ist, für alle Anfragen ( hier : Umgebungen, in denen der Roboter seinen Weg sucht ) gilt. Als Kosten werden bei diesem Algorithmus die Weglängen angenommen. Man kann leicht eine Umgebung konstruieren in der der Roboter im Onlinealgorithmus Bug1 eine beliebig größere Weglänge benötigt um vom Start- zum Zielpunkt zu finden als ein optimaler Offlinealgorithmus dem die gesamte Umgebung mitsamt Lage und Form der Hindernisse von vornherein bekannt sind :



In dieser Umgebung führt der kürzeste Weg rechts am Hindernis vorbei. Man kann das Hindernis jedoch beliebig nach links erweitern und dann benötigt Bug1 einen im Verhältnis zum kürzest möglichen Weg beliebig langen Weg um vom Startpunkt zum Zielpunkt zu finden. Also ist der Algorithmus „Bug1“ nicht kompetitiv !

Verwendete Literatur : - Rolf Klein, Algorithmische Geometrie, Addison Wesley 1997  
 - V. J. Lumelsky and A. A. Stepanov. Path-planning strategies for a point mobile automaton moving amidst unknown obstacles of arbitrary shape. Algorithmica, 1997.

