

Entwurf von Algorithmen zur Lösung von Objektverschiebungsproblemen

Kolloquium

Maximilian Mühlfeld

23. Oktober 2014

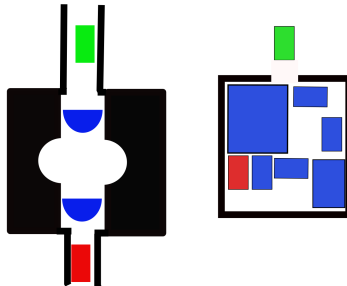
Inhaltsverzeichnis

- 1 Einleitung
- 2 Lösungsansätze und Grundlagen
- 3 Kantenrepräsentation mit Vektoren
- 4 Kantenrepräsentation mit Funktionen
- 5 Resultat und Diskussion

Inhalt

- 1 Einleitung
- 2 Lösungsansätze und Grundlagen
- 3 Kantenrepräsentation mit Vektoren
- 4 Kantenrepräsentation mit Funktionen
- 5 Resultat und Diskussion

Ziel



Lösen von Verschiebungsproblemen bei gegebener Startkonfiguration und Zielposition eines Objektes.

Motivation: Industrieroboter

Industrieroboter werden meist nur in statischer Umgebung genutzt und eingerichtet mit absolut getrennten Arbeitsbereichen, da die Konfiguration der Roboterbahn somit simpel bleibt.

Idee:

- Mehrere Roboterarme parallel im selben Arbeitsraum.
- Endeffektorwegplanung dynamisch mithilfe von Objekterkennung.

Motivation: Automatisiertes Rätselerstellen

Automatisches Testen von Rätseln auf Lösbarkeit.

- Ermöglicht das Erstellen komplett zufälliger Rätsel (z.b. im Rahmen eines Spieles).
- Erlaubt bei Erweiterung um mehrere Zielobjekte das Lösen von Puzzles mit vorgegebenem Endzustand.

Inhalt

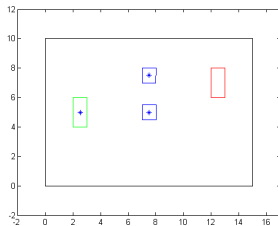
- 1 Einleitung
- 2 Lösungsansätze und Grundlagen**
- 3 Kantenrepräsentation mit Vektoren
- 4 Kantenrepräsentation mit Funktionen
- 5 Resultat und Diskussion

Ideen

- Suche auf einem komplett definiertem Raster.
- Schrittweise suchen nach einer besseren Konfiguration bis zur Zielkonfiguration mit/ohne Raster.
- Einteilen des Raumes in Zellen und diese als Suchraum nutzen.

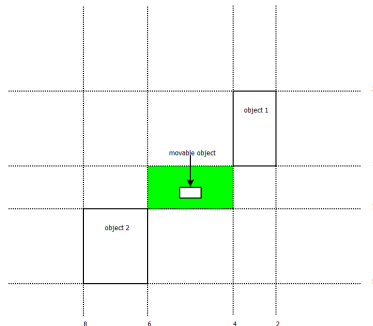
Abstraktion: Objekte

- Objekte werden als konvex angenommen.
- Jedes Objekt ist durch einen Angelpunkt und der geordneten Liste seiner Eckpunkte definiert.
- Je nach gewählter Repräsentation kann der Kantenzug des Objektes vorausberechnet mitgeliefert werden.



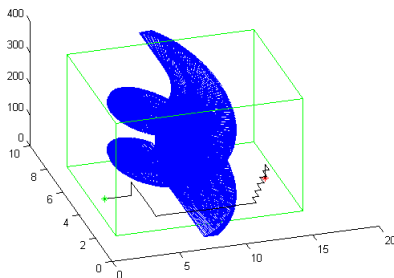
Abstraktion: Zellen

- Zellen sind durch die Verlängerung der Kanten der Objekte begrenzt.
- Jede Zelle ist definiert durch ihre Lage zu ALLEN Kanten der Objekte.

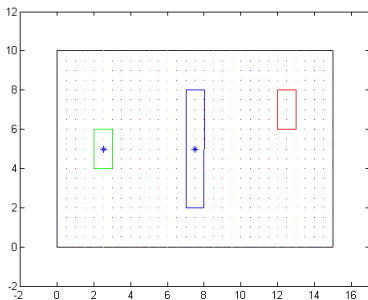


Abstraktion:Konfiguration

- Angelpunkte aller Objekte zusammen mit deren Drehwinkeln ergibt die momentane Konfiguration.



Definiertes Raster



Definiertes Raster

Für ein komplett definiertes Raster aller Konfigurationen muss jede mögliche Kombination berechnet werden.

Mit nur zwei Objekten (ein Hauptobjekt und ein bewegliches Hindernis) auf einem 10x10 Raster folgt für die Anzahl an Kombinationen:

$$\begin{aligned} N_{conf} &= (10 * 10 * 360)^2 \\ &= 1296000000 \\ &= 1.296 \cdot 10^9 \end{aligned}$$

TODO: tolle Beispiele auf Zettel wieviel Speicher/Zeit benötigt

Schrittweise aufbauen des Suchraumes mit Raster

Anstelle das komplette Raster zu erstellen, werden immer nur die Knoten erstellt, die für den nächsten Schritt benötigt werden.

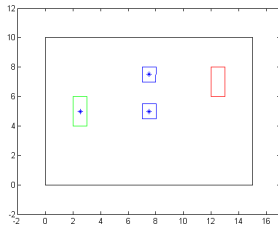
- Suchalgorithmen wie A^* /Dijkstra erlauben das generieren der Randknoten zur Laufzeit.
- Verringert die Menge an benötigten Knoten enorm im Vergleich zum komplett definiertem Raster.

Schrittweise aufbauen des Suchraumes ohne Raster

Wir entfernen das Raster und erhalten dadurch eine unendliche Menge an möglichen Konfigurationen.

- Genaueres Bewegen der Objekte möglich durch exaktere Positionierung.
- Die Anzahl der Knoten auf dem Pfad Ergebnispfad ist geringer

Schrittweise aufbauen des Suchraumes ohne Raster

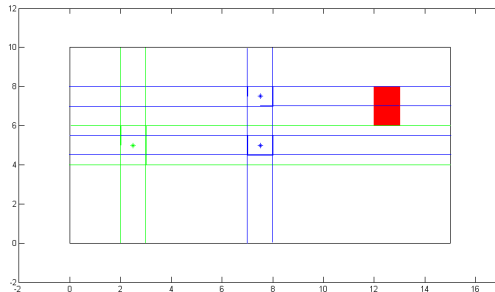


Einteilung des Raumes in Zellen

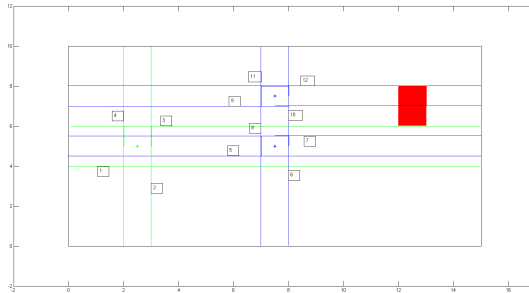
Wir fügen eine Einteilung des Raumes in Zellen hinzu, welche zwar die Menge an Konfiguration reduziert, jedoch nicht die Genauigkeit.

- Zellen erlauben das Zusammenfassen von vielen Konfigurationen zu einem Punkt im Konfigurationsraum.
- Die Zahl der Knoten die erstellt werden ist abhängig von der Zahl der Objekte, und nicht mehr von der gewünschten Genauigkeit.

Schrittweise aufbauen des Suchraumes mit Zellen



Schrittweise aufbauen des Suchraumes mit Zellen



Inhalt

- 1 Einleitung
- 2 Lösungsansätze und Grundlagen
- 3 Kantenrepräsentation mit Vektoren**
- 4 Kantenrepräsentation mit Funktionen
- 5 Resultat und Diskussion

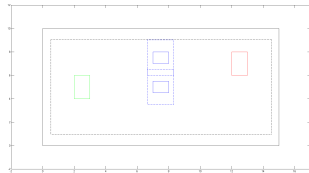
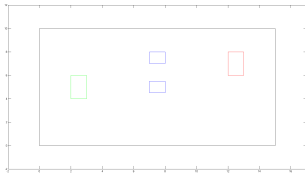
Kanten als Vektoren

Da die Objekte aus sortierten Listen O_E der Ecken bestehen, sind die Vektoren der Kanten O_K einfach zu berechnen aus:

$$O_K[i] = O_E[i + 1] - O_E[i]$$

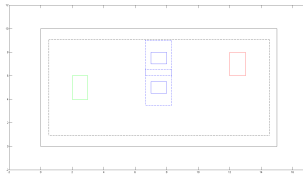
Kollisionsberechnung mittels Konvexer Hülle I

Aufstellen der Konvexen Hülle aus den Ecken des momentan bewegten Objektes und allen anderen.



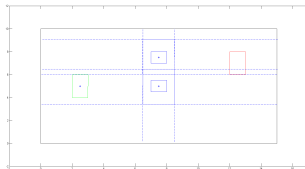
Kollisionsberechnung mittels Konvexer Hülle II

Der Vektor bestehend aus der Objektmitte als Aufpunkt und dem Richtungsvektor der Bewegung wird nun auf Schnittpunkte mit den Berechneten Vektoren der Konvexen Hüllen überprüft.



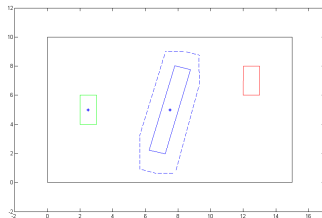
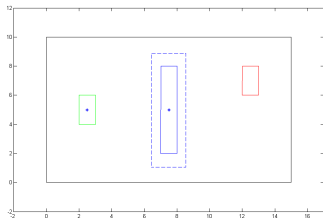
Darstellung der Zellen

Zellen sind die Verlängerung der Kantenvektoren der Konvexen Hülle.



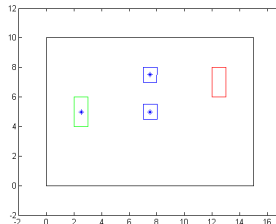
Auswirkungen von Rotation

Rotationen erhöhen die Anzahl an Kanten der Konvexen Hüllen.



Lösungsweg am Beispiel

Lösung eines Rätsels mit Vektorrepräsentation und Zelleinteilung



Inhalt

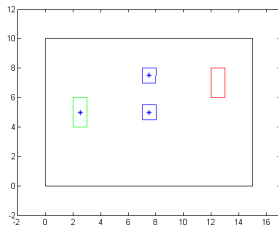
- 1 Einleitung
- 2 Lösungsansätze und Grundlagen
- 3 Kantenrepräsentation mit Vektoren
- 4 Kantenrepräsentation mit Funktionen**
- 5 Resultat und Diskussion

Kanten als Funktionen

Bei der Erstellung der Objekte werden nicht Eckpunkte, sondern Funktionen und Definitionsbereiche festgelegt.

Kollisionsberechnung mittels Funktionen

Kollision ist nun durch das finden von Schnittpunkten zwischen den Funktionen möglich. Durch Definitionsbereich werden viele Rechnungen stark verkürzt.

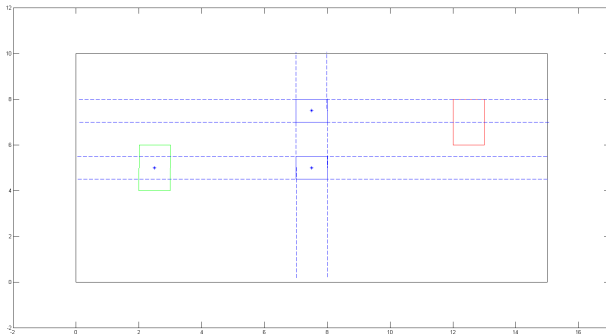


Auswirkungen von Rotation

Rotationen könnten dazu führen, dass eine vertikale Kante mit einer Steigung von unendlich erzeugt wird.

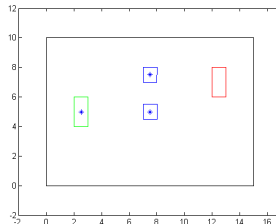
Darstellung von Zellen

Zellen sind durch die Lage des Objektes zu den Funktionen
Definiert.



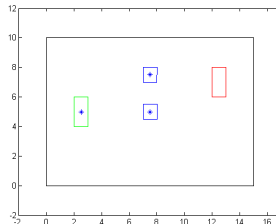
Lösungsweg am Beispiel

Lösung eines Rätsels mit Funktionsrepräsentation ohne Zellen



Lösungsweg am Beispiel

Lösung eines Rätsels mit Funktionsrepräsentation mit Zellen



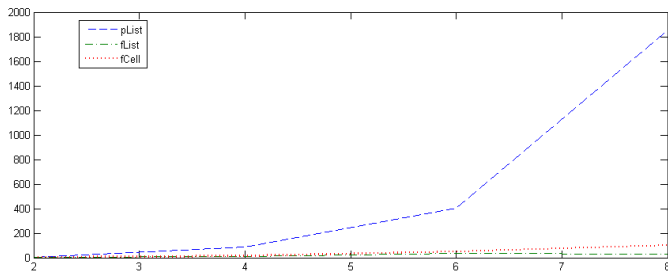
Inhalt

- 1 Einleitung
- 2 Lösungsansätze und Grundlagen
- 3 Kantenrepräsentation mit Vektoren
- 4 Kantenrepräsentation mit Funktionen
- 5 Resultat und Diskussion**

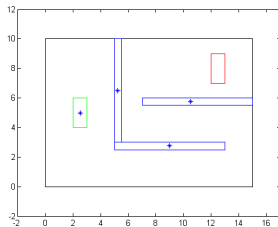
Zeiten der verschiedenen implementierten Algorithmen:

Algorithmus	2 Objekte	4 Objekte	6 Objekte	8 Objekte
Vector+Zellen	4.9305	89.3621	403.9329	1855.4
Funktion-Zellen	0.6747	7.7242	33.7001	32.1049
Funktion+Zellen	0.1472	13.5623	50.3155	100.63497

Benötigte Zeit gegen Objekte im Rätsel



"Schlechte" Rätsel:



6 Objekte			
Algorithm	schnellste Zeit	längste Zeit	mittlere Zeit
Funktion-Zellen	34234	35599	34917
Funktion+Zellen	24.036	27.8225	25.001

Ausblick:

- Zulassen nicht-konvexer Objekte.
- Rotation Abschnittsweise als kontinuierliche Funktion definieren.
- Krumme Oberflächen exakt durch Funktionen Repräsentieren.

Fragen?

Vielen Dank für Ihre Aufmerksamkeit.