

4.3.10. Plane Sweep Alg. zum Mischen von zwei Konturen A und B.

(49)

→ 4.3.10.1. Problemstellung:

Eingabe: Zwei Konturen A, B in den Ecken durch zwei Mengen von Liniensegmenten.

Ausgabe: Menge von Segmenten, die die Kanten von $A \cup B$ definieren.

→ 4.3.10.2. Definition: Ein Segment heißt sichtbar, wenn es zur Ausgabe gehört, d.h. auf Rand von $A \cup B$ liegt.

→ 4.3.10.3. Idee:

Wir modifizieren den Plane Sweep Alg. zum Linienschnitt.

Y-Struktur: Folge der von SL geschnittenen Segmenten (von unten nach oben sortiert). Wir speichern für jedes Segment in der Y-Struktur, ob es zur Zeit sichtbar oder unsichtbar ist.

Außerdem speichern wir für jedes Paar (s_1, s_2) von in Y benachbarten Segmenten, ob und von wem das Gebiet zwischen s_1 und s_2 zur Zeit überdeckt wird.

$$\text{cover}(s_1, s_2) = \begin{cases} \emptyset, & \text{nicht überdeckt} \\ \{A\}, & \text{von A} \\ \{B\}, & \text{von B} \\ \{A \cup B\}, & \text{von beiden.} \end{cases}$$

→ 4.3.10.4. Aktionen:

1. Linker Endpunkt $p \in A$ (B analog) d.h. linke Ecke von zwei Segmenten. $s_1, s_2 \in A$.

$$s' = Y.\text{succ}(s_1), s'' = Y.\text{pred}(s_2)$$

Falls $\text{cover}(s', s'') = \emptyset$ oder $\text{cover}(s', s'') = A$ dann

markiere s_1, s_2 als sichtbar

sonst unsichtbar

Falls $A \in \text{cover}(s', s'')$ (d.h. bei p beginnt ein Loch) dann:

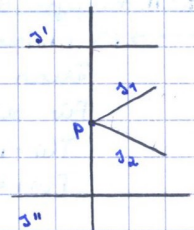
$$\text{cover}(s_1, s_2) \leftarrow \text{cover}(s', s'') \setminus \{A\}$$

sonst $\text{cover}(s_1, s_2) \leftarrow \text{cover}(s', s'') \cup \{A\}$.

$$\text{cover}(s_1, s_1) \leftarrow \text{cover}(s', s'')$$

$$\text{cover}(s_2, s_2) \leftarrow \text{cover}(s', s'')$$

$$Y \leftarrow Y \cup \{s_1, s_2\}$$

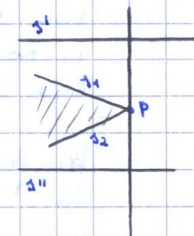


2. Rechter Endpunkt $p \in A$ (B analog)

Falls s_1 bzw. s_2 sichtbar, dann Ausgabe der betreffenden Segmente.

$$Y \leftarrow Y \setminus \{s_1, s_2\}$$

$$\text{cover}(s', s'') \leftarrow \text{cover}(s', s_1) \quad (= \text{cover}(s_2, s''))$$



3. Schnittpunkt $p = s_1 \cap s_2$

$$\text{oder } A \quad s_1 \in A, s_2 \in B$$

Falls s_1 (s_2) sichtbar \Rightarrow Ausgabe bis p (spalten!)

$$Y \leftarrow Y \setminus \{s_1, s_2\}, Y \leftarrow Y \cup \{s'_1, s'_2\}$$

in umgekehrter Reihenfolge mit negativer Sichtbarkeitsinformation

$$\text{cover}(s', s'_1) \leftarrow \text{cover}(s', s_1)$$

$$\text{cover}(s', s'_2) \leftarrow \text{cover}(s_2, s'')$$

Falls $A \in \text{cover}(s_1, s_2)$ dann

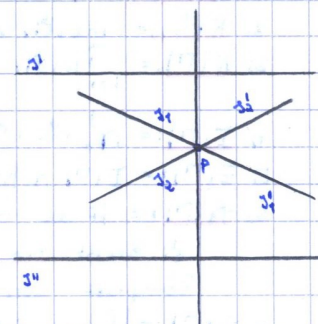
$$\text{cover}(s'_2, s'_1) \leftarrow \text{cover}(s_1, s_2) \setminus \{A\}$$

sonst $\text{cover}(s'_2, s'_1) \leftarrow \text{cover}(s_1, s_2) \cup \{A\}$

Falls $B \in \text{cover}(s_1, s_2)$ dann

$$\text{cover}(s'_2, s'_1) \leftarrow \text{cover}(s_1, s_2) \setminus \{B\}$$

sonst $\text{cover}(s'_2, s'_1) \leftarrow \text{cover}(s_1, s_2) \cup \{B\}$.



4. Durchgang $p \in A$ (B analog)

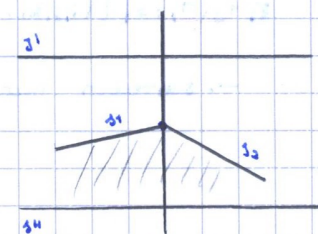
s_1 endet und s_2 beginnt

$$Y \leftarrow Y \setminus \{s_1\}, Y \leftarrow Y \cup \{s_2\}$$

Falls s_1 sichtbar \Rightarrow Ausgabereihenfolge

$\text{sichtbarkeit}(s_2) \leftarrow \text{sichtbarkeit}(s_1)$

$$\text{cover}(s', s_2) \leftarrow \text{cover}(s', s_1); \text{cover}(s_2, s'') \leftarrow \text{cover}(s_1, s'')$$



→ 4.3.10.5 Bem:

- Es gibt nur Schnittpunkte zwischen Segmenten aus verschiedenen Kurven.
- Segmente werden an Schnittpunkten gespalten (→ entweder ganz sichtbar oder ganz unsichtbar.)
- Die Sichtbarkeitsinformation ändert sich nur bei Schnittpunkten (wechselt von sichtbar zu unsichtbar oder umgekehrt).

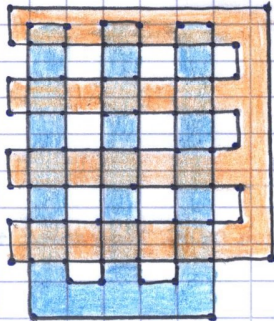
→ 4.3.10.6 Lemma:

Zwei Kurven A und B mit insgesamt n Ecken können in Zeit $O((n+s)\log n)$ berechnet werden, wobei $s = \#$ Schnittpunkte zw. Segmenten in A und B.

Beweis: siehe Segmenteschnitt.

→ 4.3.10.7 Bem: Im Allg. ist $s = O(n^2)$...

→ 4.3.10.8 Bsp:



z.z.: $T(n) = O(n \log^2 n)$.

Indann: Beh. gelte für $\frac{n}{2}$

Indschritt: dann gilt sie auch für n !

$$\begin{aligned}
 T(n) &\leq c_1 n \log n + 2T\left(\frac{n}{2}\right) \leq c_1 n \log n + 2 \cdot c_2 \frac{n}{2} \cdot \log^2\left(\frac{n}{2}\right) = \\
 &= c_1 n \log n + c_2 n (\log n - \log 2)^2 = c_1 n \log n + c_2 n (\log n - 1)^2 = \\
 &= c_1 n \log n + c_2 n \log^2 n - 2c_2 n \log n + c_2 n = \\
 &= n \log^2 n \left[c_1 \cdot \frac{1}{\log n} + c_2 - \frac{2c_2}{\log n} + \frac{c_2}{\log n} \right] = \\
 &= n \log^2 n \cdot \left[\underbrace{c_1 - 2c_2}_{\xrightarrow{n \rightarrow \infty} 0} + c_2 + \underbrace{\frac{c_2}{\log^2 n}}_{\xrightarrow{n \rightarrow \infty} 0} \right] \leq \\
 &\leq M \cdot n \log^2 n \quad \forall n \geq n_0
 \end{aligned}$$

4.3.11. Analyse der Laufzeit:

→ 4.3.11.1. Idee: Wir zeigen nun, dass bei unserer Anwendung $s = O(n)$

Dann kostet der Merschritt $O(n \log n)$

Für die Gesamtlaufzeit $T(n)$:

$T(1) = 1$

$T(n) = O(n \log n) + 2T\left(\frac{n}{2}\right) \Rightarrow T(n) = O(n \log^2 n)$

Übung 8.

Um die $O(n)$ Schranke für s zu zeigen, verwenden wir die Tatsache, dass sich die Ränder von jeweils zwei aufgetrennten Hindernissen Q_i und Q_j in höchstens zwei Punkten schneiden.

Bew. Übung.

Wir betrachten ein etwas allgemeineres Problem:

Sei $\Gamma = \{ \delta_1, \dots, \delta_m \}$ eine Menge einfacher geschlossener Kurven (= Jordankurven) mit $|\delta_i \cap \delta_j| \leq 2 \quad \forall i \neq j$. d.h. zwei Kurven schneiden sich höchstens zweimal.

→ 4.3.11.2 Definitionen:

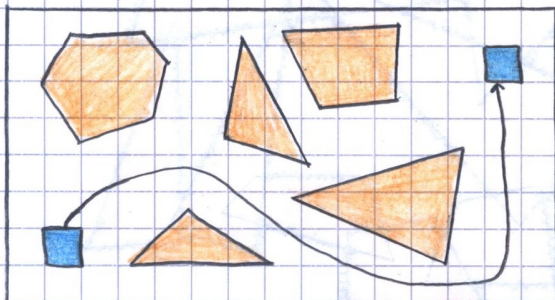
- $\text{Inn}(\delta_i)$: = Innere Regionen von δ_i
- $K(\delta_i)$: = $\text{Inn}(\delta_i) \cup \delta_i$ (also Abschluss)
- $K(\Gamma)$: = $\bigcup_{i=1}^m K(\delta_i)$
- $I(\Gamma)$: = $\bigcup_{i \neq j} (\delta_i \cap \delta_j)$ (alle Schnittpunkte)
- $E(\Gamma)$: = $I(\Gamma) \cap \text{Rand}(K(\Gamma))$
- $\tau_1(\Gamma)$: = # der redundanten Kurven in Γ , wobei: δ_i heißt redundant, falls $\delta_i \subset \bigcup_{i \neq j} K(\delta_j)$, d.h. δ_i wird komplett von anderen Kurvenflächen überdeckt. also $\Rightarrow E(\Gamma \setminus \{\delta_i\}) = E(\Gamma)$
- $\tau_2(\Gamma)$: = $|\{ (i,j) : i \neq j \wedge \delta_i \cap \delta_j \neq \emptyset \}|$
- $\tau_3(\Gamma)$: = $|\{ (i,j,k) : i,j,k \text{ paarw. versch.}, K(\delta_i) \cap K(\delta_j) \cap K(\delta_k) \neq \emptyset \}|$.

→ schreiben nun einfacher τ_i statt $\tau_i(\Gamma)$ für $i=1,2,3$.

4.3 Problem 2: R ist konv. Polygon und S Menge von konv. Polygonen.

4.3.1. Problemschilderung: S ist eine Szene von m konvexen Polygonen P_1, \dots, P_m mit $n := \sum_{i=1}^m \# \text{Ecken von } P_i$ und $P_i \cap P_j = \emptyset \quad \forall i \neq j$

R (Roboter) ist konvexes Polygon mit c Ecken ($c = \text{Konstante}$),
Bewegungen: Nur Translationen von R (keine Rotationen).



4.3.2. Ann:

- S ist in einem Rechteck eingeschlossen, z.B. Dummi-Hindernisse.
- Sei p ein beliebiger Referenzpunkt im Inneren von R , dann ist die Position von R durch die Position von p im \mathbb{R}^2 definiert.

4.3.3. Idee:

Reduziere das Problem auf das Bewegungsplanungsproblem eines plattformigen Roboters (p) in einer komplizierten Szene S' .

Dazu blähen wir alle Hindernisse P_1, \dots, P_m auf.

D.h. alle Hindernisse werden um das Maß vergrößert, auf das der Roboter verkleinert werden muss, um ein Pfad zu werden.

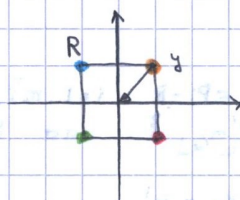
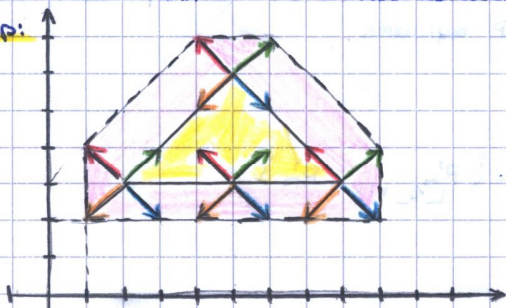
4.3.4. Konstruktion von P_i' (= aufgeblähtes Hindernis).

$P_i' = P_i - R =$ "Minkowski-Differenz".

$$= \{x - y : x \in P_i \wedge y \in R\}$$

\uparrow \mathbb{R}^2 \uparrow im Koordinatensystem des Roboters.

Bsp:



orange $(2,3) - (1,1) = (1,2)$

blau $(2,3) - (-1,1) = (3,2)$

rot $(2,3) - (1,-1) = (1,4)$

grün $(2,3) - (-1,-1) = (3,4)$

pink kommt dazu

Betrachte die Szene $S' = \{P_1', \dots, P_m'\}$ mit $P_i' = P_i - R, i=1..m$.

Dann ist die Menge der freien Platzierungen von R in S' (bzgl. des Referenzpunktes p):
 $FP = \mathbb{R}^2 \setminus \bigcup_{i=1}^m P_i'$

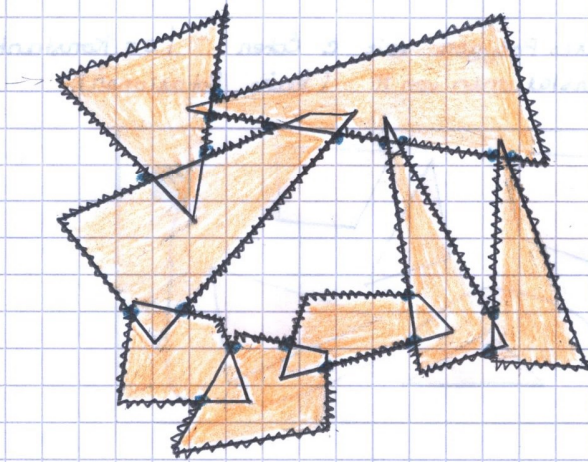
d.h. das Komplement der Vereinigung aller aufgeblähten Hindernisse.
FP ist i.a. nicht mehr zusammenhängend.

D.h. FP ist ein u.U. nicht zusammenhängendes Gebiet der Ebene, begrenzt durch Polygonzüge.

Es ist offensichtlich, dass eine Bewegung von A nach B genau dann möglich ist, wenn A und B in derselben Zusammenhangskomponente von FP liegen.

Übung 8.1.

4.3.5. Beispiel:



$n = \#$ alle Ecken
 Hier: $28 = n$
 Auf dem Rand dann: $19 < n$.

4.3.6. Wkt: der Rand oder die Kontur von FP hat Größe $O(n)$

Vermutung: $\#$ alle Ecken auf dem Rand der Vereinigung ist $O(n)$.

4.3.7. Satz: Seien P_1, \dots, P_m paarweise disjunkte, konvexe Polygone mit insgesamt n Ecken und R ein konvexes Polygon mit konstant vielen Ecken.

Dann hat $\text{Kontur} = K = \bigcup_{i=1}^m (P_i - R)$ $O(n)$ Ecken.
 Polygonfläche mit Höchern.

Bew. später.

4.3.8. Algorithmus:

Um Problem 2 zu lösen, müssen wir FP und seine Zusammenhangskomponenten berechnen. Dazu berechnen wir zunächst die Kontur (K) von FP, dh. die Menge der Kanten, die den Rand von FP definieren.

→ Divide & Conquer.

Algorithmus:

1) Berechne $S' = \{P_i' = P_i - R : i = 1..m\}$.

2) Sei $S_1 := \{P_1' \dots P_{\lfloor m/2 \rfloor}'\}$ und $S_2 := \{P_{\lfloor m/2 \rfloor + 1}' \dots P_m'\}$

Falls $|S_1| = 1 \Rightarrow K_1 = P_1'$

Falls $|S_2| = 1 \Rightarrow K_2 = P_{\lfloor m/2 \rfloor + 1}'$

sonst:

Berechne rekursiv:

$K_1 = \text{Kontur von } S_1$

$K_2 = \text{Kontur von } S_2$

3) Berechne $K = \text{Kontur von } S'$ durch Überlagerung von K_1 und K_2 .

4.3.9. Laufzeit: Schritt 1: $O(n)$

Dazu nötig, dass Minkowski-Differenz von konvexen Polygonen in linearer Zeit.

→ Übung

Rest: $T(n) = 2 \cdot T(n/2) + \text{Zeit für Mischen}$

↑ Vereinigung von zwei Konturen K_1 und K_2 .

