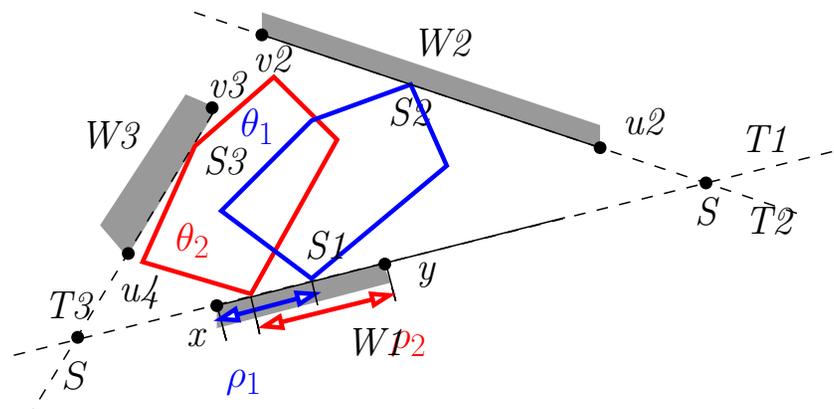


Offline Bewegungsplanung: Kritische Platzierungen

Elmar Langetepe
University of Bonn

Besonderheiten der Arrangements $A(O_i)$!

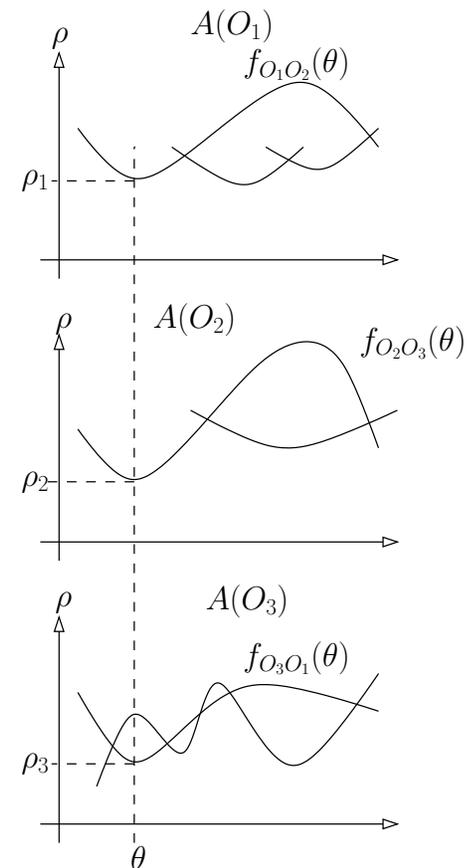
- Ein O_i kann zwei verschiedene Arrangements besitzen
- Je nachdem in welche Richtung wir schieben müssen
- Arrangements $A_L(O_i)$ und $A_R(O_i)$
- Aufwand verdoppelt
- Definitionsbereich kann zerfallen!



Kandidaten Krit. Platz. Def. 2.37

T^+ Platzierungen (x, y, θ) : ■

- a Ecken untere Konturen der $O(mn)$ Arrangements $A(O_i)$ ■
- b Zyklische Tripel von Kontaktpaaren (O_1, O_2, O_3) : ■
 - O_1 beschränkt O_3 , O_3 beschränkt O_2 und O_2 beschränkt O_1 . ■
 - Tripel liefert ≤ 4 Platzierungen ■ Platzierungen durch (ρ_1, θ) (oder $(\rho_2, \theta), (\rho_3, \theta)$) festgelegt. ■
 - Nullstellen einer Kurve 4. Grades. ■



T^+ berechnen **Theorem 2.38**

T^+ kann in Zeit $O(mn\lambda_6(mn) \log(mn))$ berechnet werden!■

■ Beweis!!■

Vorsicht: T^+ kann noch verbotene Platzierungen enthalten!!!

Bislang gar nicht überprüft!!■

T^+ berechnen **Alg. 2.8** Arrangements aufbauen!

- Kontaktpaar O_i ■
- $\Pi_{O_i O_k} := \{ \theta | O_k \text{ beschränkt } O_i \text{ bei Winkel } \theta \}$.■
- $O(nm)$ mal $O(nm)$ ■
- Bilde Funktionen■
- Bilde Arrangements $A(O_i)$ ■
- `KritPlatz.html`■
- $O((nm)^2)$ ■

T^+ berechnen: **Alg. 2.8** Untere Konturen $UK(A(O_i))$

- $O(nm)$ Arrangements $A(O_i)$ ■
- Funktionen schneiden sich maximal 4 mal ■
- Divide and Conquer ■
- Merge: Zwei Konturen, Sweep $O(2\lambda_6((mn)/2))$: ■
Komplexität gesamte Kontur $O(\lambda_6(mn))!!$ ■
- Eine Kontur in $O(\lambda_6(mn) \log(mn))$ ■
- Alle unteren Konturen in $O(mn\lambda_6(mn) \log(mn))$ ■

T^+ berechnen: **Alg. 2.8** Untere Konturen $UK(A(O_i))$

- Merge: Zwei Konturen mergen, Sweep, $O(\lambda_6(mn))$ ■
- Divide and Conquer abschätzen ■
- $T(mn) = 2T\left(\frac{mn}{2}\right) + C \times \lambda_6(mn)$ ■
- $T(mn) \leq \dots \leq 2^k T(1) + C \times \sum_{i=0}^{\log(nm)} 2^i \lambda_6\left(\frac{mn}{2^i}\right)$ ■
- $2\lambda_s(n) \leq \lambda_s(2n)$, $2^i \lambda_s\left(\frac{n}{2^i}\right) \leq \lambda_s(n)$ ■
- $T(mn) \leq C \times \lambda_6(mn) \log(mn)$ ■

T^+ berechnen: **Alg. 2.8** Platzierungen!

- Fall 1: Knoten der unteren Konturen: $O(mn\lambda_6(mn))$ viele ■
- Fall 2: Zyklische Tripel! Alle Konturen betrachten ■
- Sortieren aller Knoten: Sortieren von sortierten Mengen ■
- $O(mn)$ Arrangements mit je $\lambda_6(mn)$ sortierten Knoten ■
- Durch simultanen Sweep: $O(mn\lambda_6(mn)\log(mn))$ ■
- Intervalle I_k sukzessive abarbeiten ■

T^+ berechnen: Alg. 2.8 Intervall I_0 !

- Beginne bei I_0 : Für jedes vorhandene O_i teste Zyklus!■
- Falls Zyklus vorhanden: Berechne max. 4 zugehörige Krit. Platzierungen■
- Berichte die Elemente aus W , deren Winkel in I_0 liegt■
- Füge andere Platzierungen in Warteschlange W nach Winkeln ein■
- Aufwand: $O(mn)$ O_i s, Ermitteln und Einfügen: $O(mn \log mn)$ ■

T^+ berechnen: Alg. 2.8 I_k zu I_{k+1} !

- Endpunkt von $A(O_1)$ markiert Übergang
- Annahme: Von $f_{O_1O_2}$ nach $f_{O_1O_4}$
- O_1 hatte über I_k einen Zyklus mit O_2 : entferne aus W die zu (O_1, O_2, O_s) gehörenden kritischen Platzierungen, da diese nicht mehr aktuell sind (Untere Kontur wechselt!) Jeweils $O(\log mn)$!
- Teste O_1, O_4 einen Zyklus (O_1, O_4, O_t) über I_{k+1} . Falls ja, berechne wie oben die kritischen Platzierungen.
Einfügen! $O(\log mn)$ ■
- Berichte alle in W gespeicherten Platzierungen $(\theta, (O_i, O_k, O_\ell))$ mit $\theta \in I_{k+1}$, entfernen aus W . Jeweils $O(\log mn)$! ■
- Aufwand: $O(mn\lambda_6(nm))$ Intervalle, W hat max. $O(mn\lambda_6(mn))$ Einträge, insgesamt: $O(mn\lambda_6(mn) \log(mn))$ ■

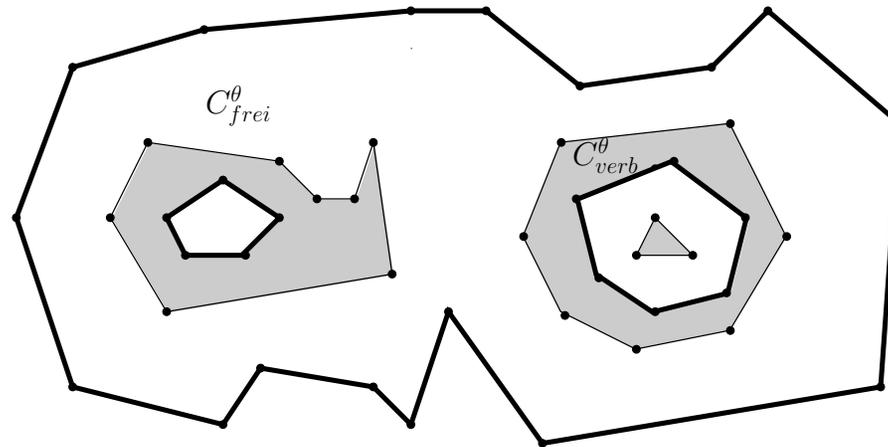
T^+ berechnen **Theorem 2.38**

T^+ kann in Zeit $O(mn\lambda_6(mn) \log(mn))$ berechnet werden! ■

■

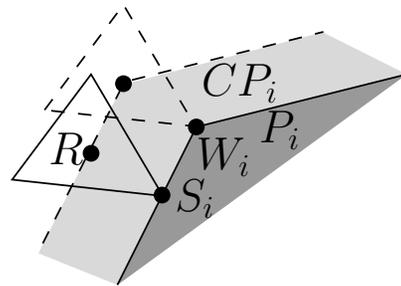
2.3.2 Kantengraph und Knotengraph

- T^+ berechnet, wichtigste Berechnung
- Idee: Bewegung durch 3D in C_{frei} und Änderungen des Konfigurationsraumes beobachten
- Startrotation θ : C_{frei}^θ
- C_{frei}^θ : Komplexität: $O(mn)$, Berechnung: $O(mn \log^2(mn))$
- Rand um Szene: Alle C_{frei}^θ beschränkt

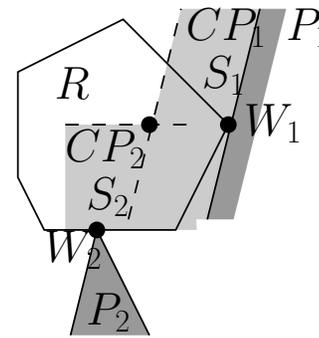


Knoten in C_{frei}^θ betrachten

- Konkave oder konvexe Knoten in pol. Szene
- Entsteht durch beteiligte Kontaktpaare und Referenzpunkt
- Ecke/Ecke oder zwei Ecke/Kante Kontaktpaare, Name!
- Änderungen über θ ermitteln



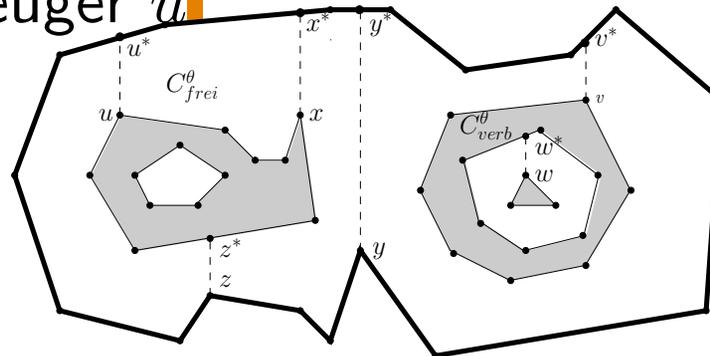
(i) konvex



(ii) konkav

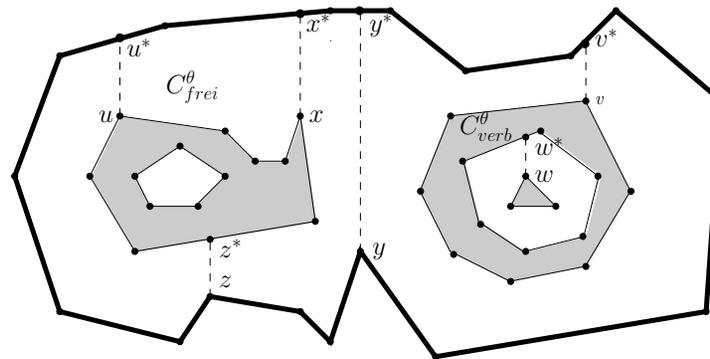
Knotengraph V^θ : Def.:2.39

- Graph, repräsentiert die Struktur von C_{frei}^θ ■
- Knoten der Polygone von C_{frei}^θ ■
- Lokal Y -max. Knoten u , ■ darüberl. Knoten u^* ,
- Kanten uu^* und Kanten zwischen adjazenten Knoten ■
- Beschriftung der Knoten und Kanten nach Entstehung:
 - Konvex: Ecke/Ecke Paar (W_i, S_i) ■
 - Konkav: Ecke/Kante Paare $(W_i, S_i) (W_j, S_j)$ ■
 - Knoten u^* : Erzeuger u ■



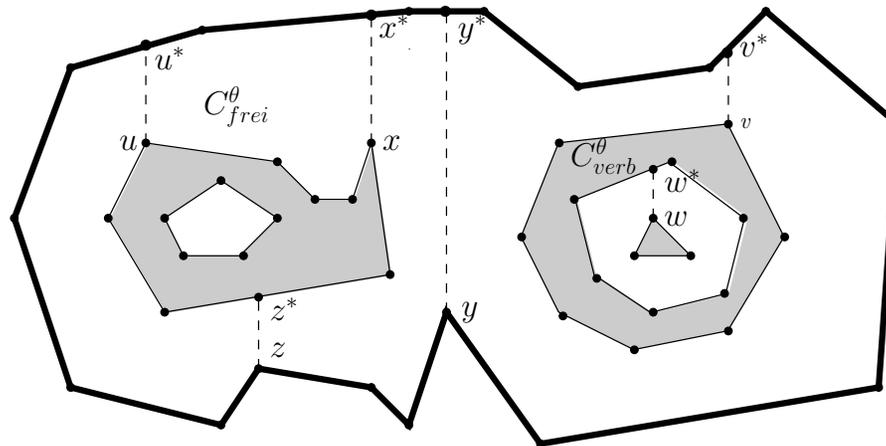
Eigenschaften V^θ : **Bem.:2.40**

- i) V^θ planarer Graph mit $O(mn)$ Knoten und Kanten.■
- ii) Jeder Knoten hat Grad ≤ 3 .■
- iii) V^θ in Zeit $O(mn \log^2(mn))$ berechnen.■
- iv) Zu einer Beschriftung findet man den Knoten/die Kante in Zeit $O(\log(mn))$.■



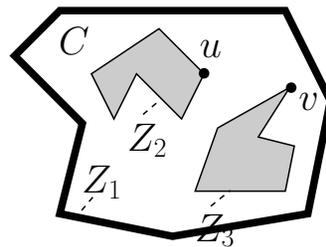
Eigenschaften V^θ : Lem.:2.41

- u, v Knoten von V^θ ■
- u, v liegen in derselben Zusammenhangskomponente von V^θ
 $\iff u, v$ liegen auf dem Rand derselben
Zusammenhangskomponente von C_{frei}^θ ■
- Beweis: \implies einfach (Tafel) ■



Lem.:2.41: " \Leftarrow "

- u, v liegen auf dem Rand derselben Zusammenhangskomponente
- von C_{frei}^θ
- Mehrere Komponenten Z_1, Z_2, \dots, Z_i
- u, v auf einem $Z_i \Rightarrow$ fertig
- Sonst: Verbindung u nach v über äußere Komp. Z_1
- Beweis: V_u^θ (V_v^θ) hat Elemente aus Z_1 (Tafel)

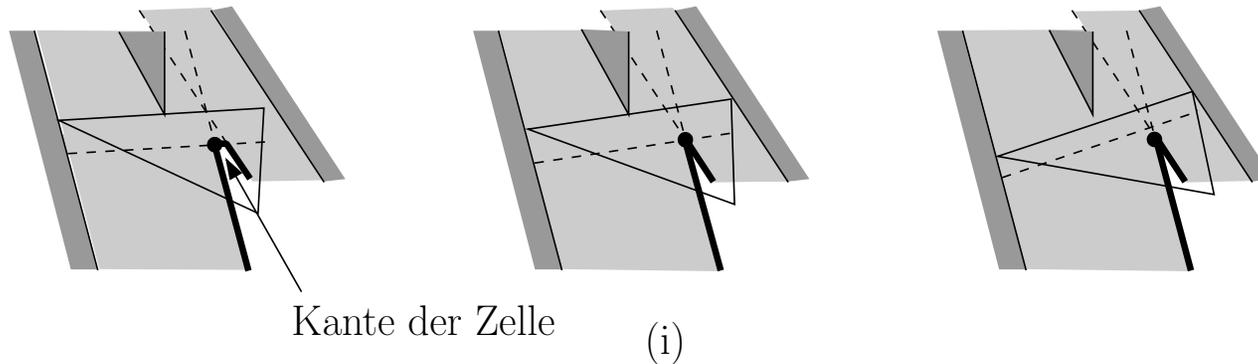


Änderungen von V^θ

- Stetige Veränderung des Winkel θ ■
- Strukturelle Veränderung von V^θ ■
- Definition: **Kritische Orientierungen**■
- Alle Arten systematisch betrachten■
- Insgesamt: $O(mn\lambda_6(mn))$ viele Änderungen von 0 bis 2π ■
- In $O(mn\lambda_6(mn) \log mn)$ Durchführen■

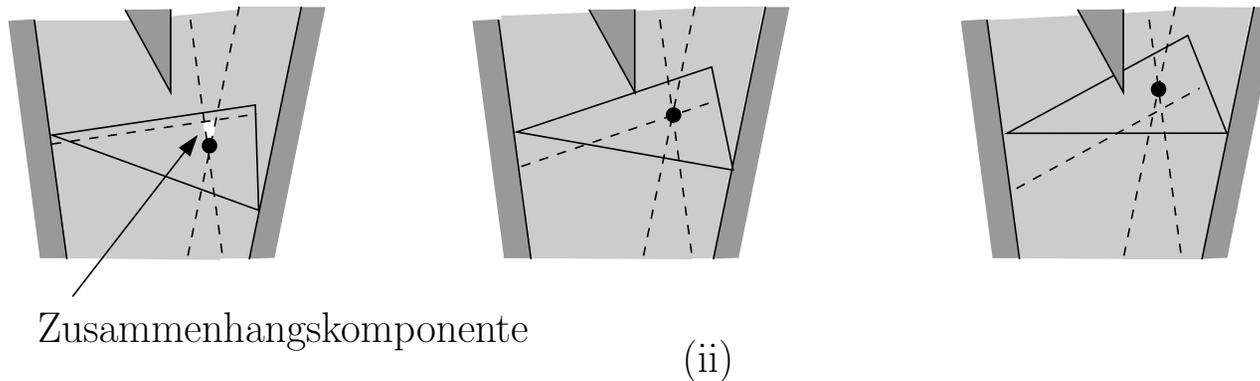
Kritische Orientierungen i)

- Kante von ∂C_{frei} verschwindet beim Übergang
- Drei Ecke/Kante Kontaktpaare
- Kritische Platzierung: $O(mn\lambda_6(mn))$ viele



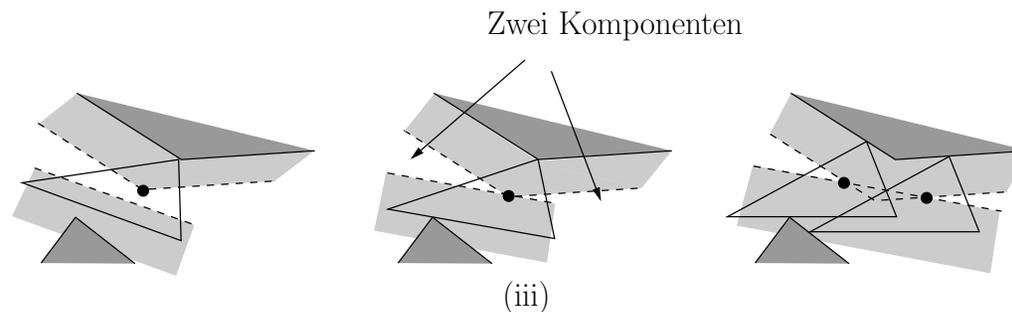
Kritische Orientierungen ii)

- Komponente von C_{frei} verschwindet beim Übergang
- Drei Ecke/Kante Kontaktpaare
- Kritische Platzierung: $O(mn\lambda_6(mn))$ viele



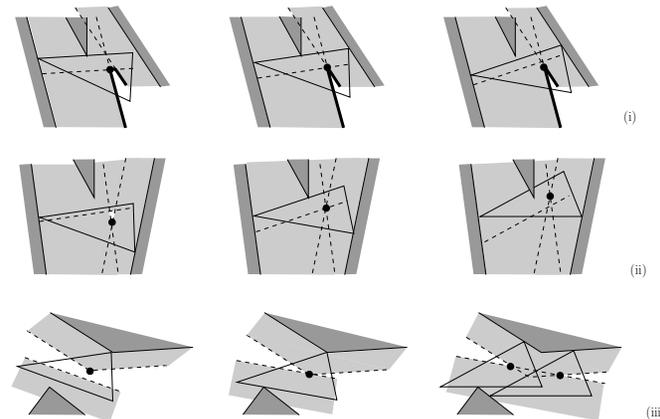
Kritische Orientierungen iii)

- Zusammenhang von C_{frei} verschwindet beim Übergang durch **konvexe Ecke**
- Ein Ecke/Ecke Kontaktpaar, ein Ecke/Kante Kontaktpaar
- Kritische Platzierung: $O(mn\lambda_6(mn))$ viele (Übungsaufgabe!!)



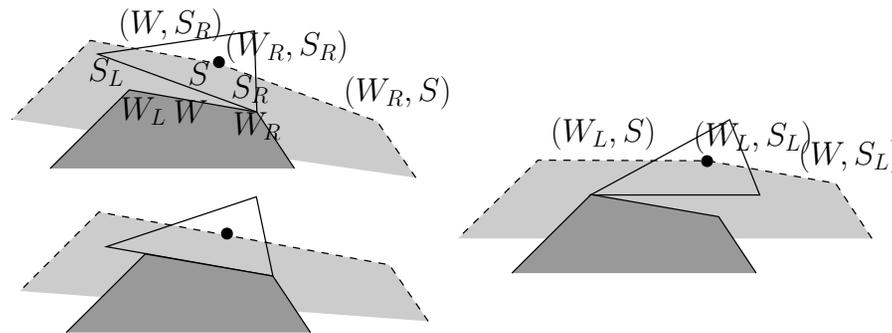
Kritische Orientierungen i)-iii)

- Kritische Platzierungen
- $O(mn\lambda_6(mn))$ viele (Kandidaten)
- Aufwand Aktualisierung von V^θ : jeweils $O(\log mn)$
- Komplexität V^θ : $O(mn)$
- Über Benennung darauf zugreifen
- Kandidaten T^+ : $O(mn\lambda_6(mn) \log(mn))$ berechn. und sort.
- Leider evtl. nicht alle korrekt



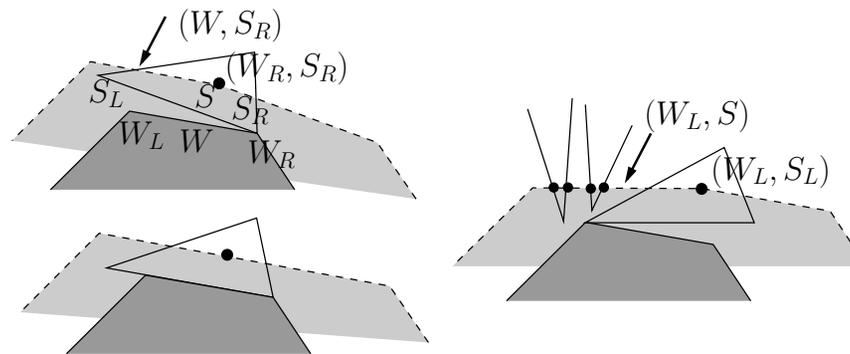
Kritische Orientierungen iv)

- Knoten und Kanten von ∂C_{frei} wechselt Beschriftung beim Übergang, zwei benachbarte Kanten werden kollinear
- Ecke/Ecke Kontaktpaar und angrenzendes Ecke/Kante Kontaktpaar
- Kante von R und Kante von P_i parallel
- $O(mn)$ mal kann das vorkommen



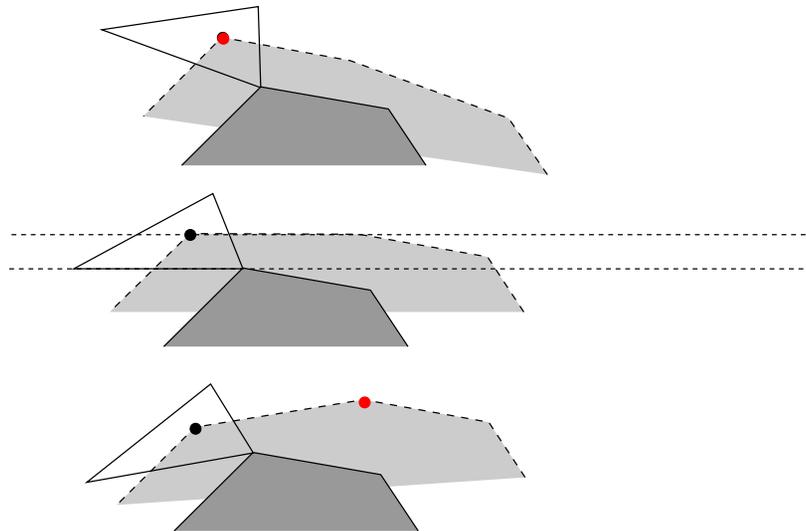
Kritische Orientierungen iv)

- Aktualisierung der Beschriftung: Von (W_R, S_R) nach (W_L, S_L) ■
- Bis zu $O(mn)$ Knoten in V^θ sind mit (W_L, S) entstanden ■
- Konkave Ecken mit zwei Kontakten ■
- Jeweils $O(\log mn)$ für Aktualisierung ■
- $O(mn)$ mal kann das vorkommen: $O((mn)^2 \log mn)$ ■



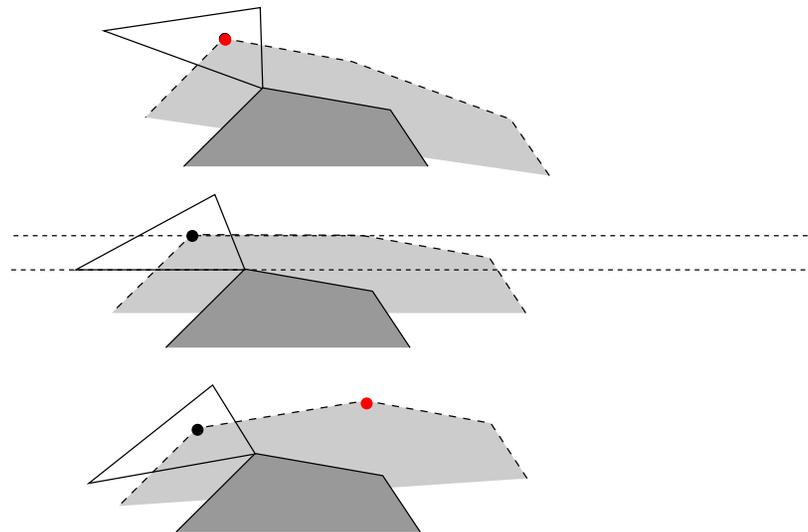
Kritische Orientierungen v)

- Lokales Y Maximum ändert sich
- Ecke/Kante Kontaktpaar und waagerechte Kante in C_{frei}



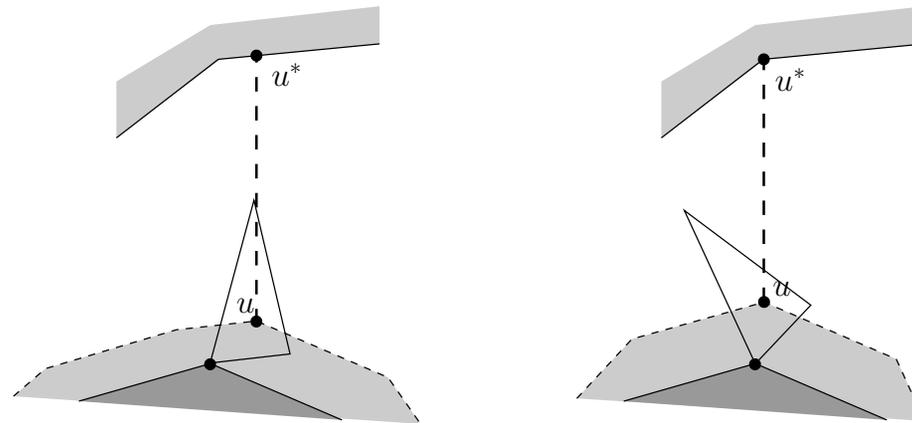
Kritische Orientierungen v)

- Für jede Kante von R kann das passieren: m Orient. ■
- Für jeden Knoten aller P_i Änderung möglich ■
- Knoten u, u^* löschen ■, Knoten v, v^* einfügen ■
- Ein voller Sweep über V^θ in $O(mn \log mn)$ ■
- Bei m Orientierungen: $O(m^2 n \log mn)$ ■



Kritische Orientierungen vi)

- Wechsel einer Ecke u^* ■
- Ecke/Ecke Kontaktpaar erzeugt konvexen Knoten ■
- Aktualisierung/neue Beschriftung: $O(\log mn)$ ■
- Vorkommen: $O(mn\lambda_6(mn))$, ■ Technik wie Krit.Platz ■



Kritische Orientierungen i)-vi)

- Insgesamt nicht mehr als $O(mn\lambda_6(mn))$ Kandidaten ■
- Alle Aktualisierungen in $O(mn\lambda_6(mn) \log mn)$ ■
- Kandidaten für i)-iv) in $O(mn\lambda_6(mn) \log mn)$ berechnen ■
- Kandidaten für i) und ii): Alg. 2.8 ■
- iii) und iv) Übungsaufgabe/Alg.2.9 ■
- Kandidaten für v)-vi): auch in $O(mn\lambda_6(mn) \log mn)$ berechnen ■
- Alle Kandidaten in $O(mn\lambda_6(mn) \log mn)$ berechnen ■
- **Alle Kandidaten in $O(mn\lambda_6(mn) \log mn)$ bearbeiten** ■
- Alle Kandidaten in Winkel-Reihenfolge bearbeiten! ■
- Problem: Sind die auch tatsächlich bereits frei? ■