Technische Universität Braunschweig

Einführung in algorithmische Geometrie

Arne Schmidt

Vorstellung

Interessen

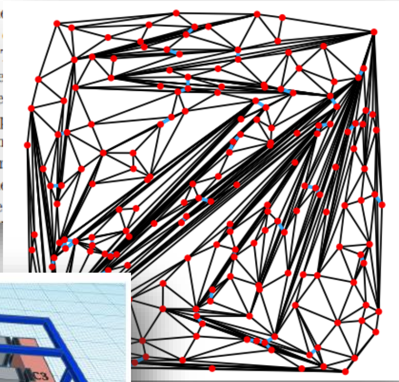
- Geometrische Optimierung,
- Programmierbare Materie,
- Komplexitätstheorie

#GernePerDu

Computing MaxMin Edge Length Triangulations

Sándor P. Fekete * Winfried Hellmann* Michael Hemmer* Arne Schmidt* Julian Troegel*

Abstract



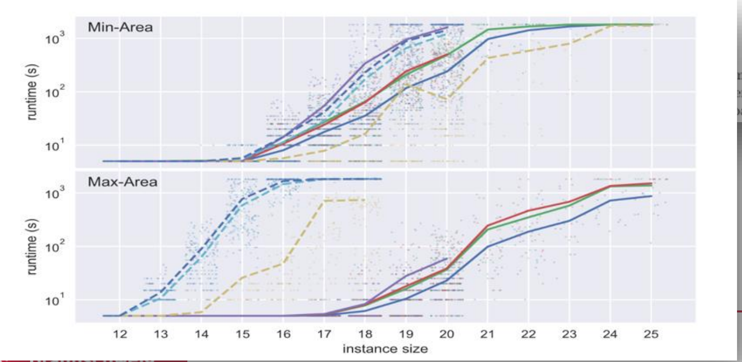
Particle-Based Assembly Using Precise Global Control*

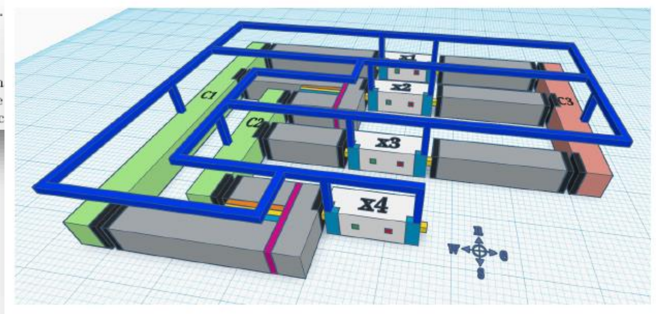
Keller[1][0000–0001–9988–953X], Christian Rieck[1][0000–0003–0846–5163], Scheffer[2][0000–0002–3471–2706], and Arne Schmidt[1][0000–0001–8950–3963]

[1] Department of Computer Science, TU Braunschweig, Germany.
{jkeller, rieck, aschmidt}@ibr.cs.tu-bs.de
[2] Department of Computer Science, University of Münster, Germany.
christian.

Computing Area-Optimal Simple Polygonizations

SÁNDOR P. FEKETE, ANDREAS HAAS, PHILLIP KELDENICH, MICHAEL PERK, and ARNE SCHMIDT, Department of Computer Science, TU Braunschweig





Braunschweig

Organisation

Arne Schmidt

Kai Kobbe

Vorlesung + Gr. Übung

- Grundlagen

- Vertiefungen

Kl. Übung

- Besprechung Hausaufgaben

Fragen

Inhaltlich oder allgemeiner Ablauf	Vorlesung / große Übung
Zu Übungsblättern oder Korrektur	Kleine Übung, Kai Kobbe (k.kobbe@tu-braunschweig.de)
Individuelle Fragen	**Immer per Mail** an aschmidt@ibr.cs.tu-bs.de (Nicht über StudIP)
Sprechstunde	Montags, 09:45 Uhr im Raum IZ 333 (Am besten vorher per Mail ankündigen)

Semesterplan (vorläufig)

Datum (VL)	VL Nummer	VL / Ü Inhalt	Hausaufgabe (Ausgabe)	Hausaufgabe (Abgabe, Do. 13:15)	Hausaufgabe (Besprechung)
10. April	0	Einleitung			
17. April	1	Range Queries			
24. April	2 + Ü	KD-Trees			
01. Mai	-		HA1		
08. Mai	3 + Ü	Sweep Line Algorithmen			
15. Mai	4 + Ü	Overlays	HA2	HA1	
22. Mai	5	Punktlokalisierung			HA1
29. Mai	-		HA3	HA2	
05. Juni	6	Trapezoidal Maps			HA2
12. Juni	Exkursionswoche				
19. Juni	7 + Ü	Minkowski Summen	HA4	HA3	
26. Juni	8	PSPACE			HA3
03. Juli	9 + Ü	Art Gallery		HA4	
10. Juli	10	Sichtbarkeitspolygone			HA4
17. Juli	11 + Ü	Zusammenfassung			

Hausaufgaben

- 20 Punkte pro Blatt
- 4 Blätter
- 50% aller Punkte (40 Punkte) für die Studienleistung
- Einzelabgaben
- Auf Papier

Klausur – Vorläufig

Datum:
31.07.25

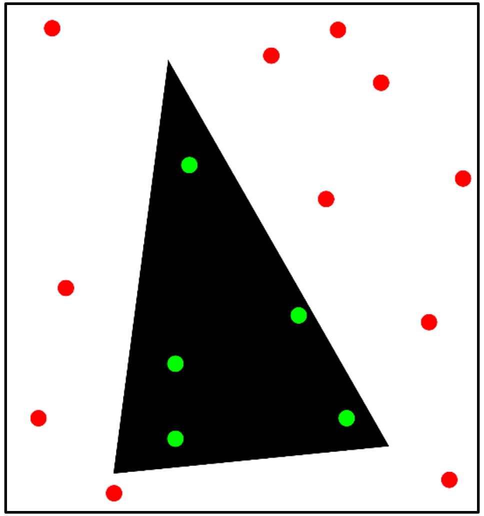
Uhrzeit:
13 Uhr – 15 Uhr

Ort:
ZI 24.2

Inhalt:
Hauptsächlich Vorlesung

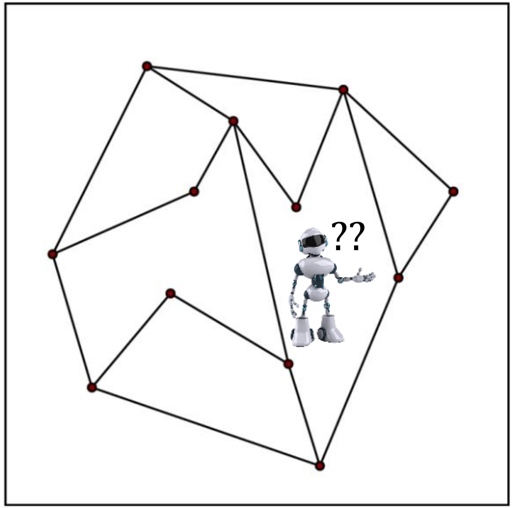
Kapitel 1 - Einführung

Probleme



Welche Punkte liegen in dem Bereich?

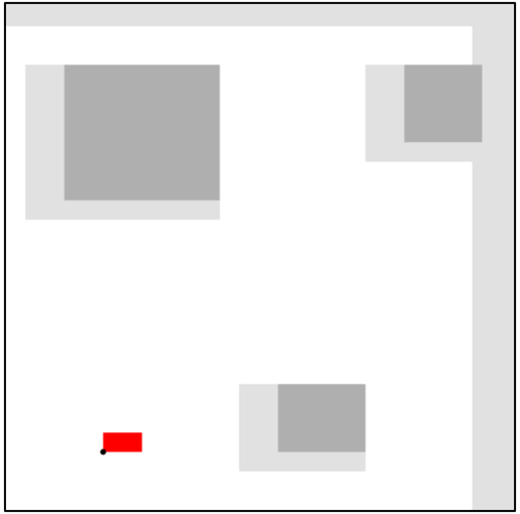
→ Range Query



In welchem Bereich liegt ein Punkt?

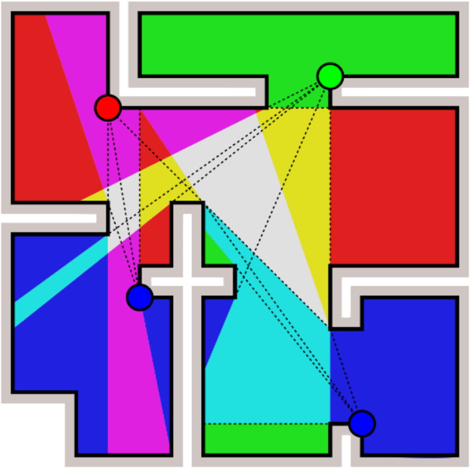
→ Punktlokalisierung

Probleme



Wohin und wie kann das rote Objekt bewegt werden?

→ Robot Motion Planning



Wie viele Wächter werden zur Abdeckung benötigt?

→ Art Gallery Problem

Kapitel 1.1 – Primitive

Primitive

Einige Operatoren werden immer wieder benötigt.

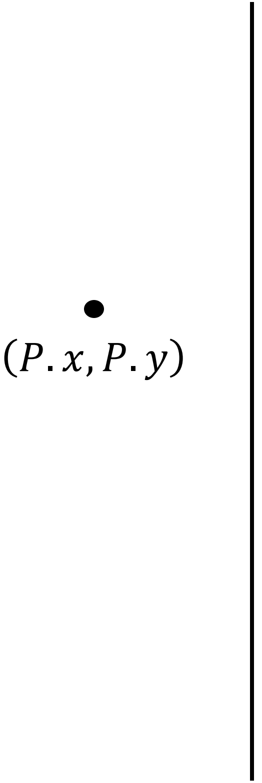
Wie können wir testen, ob ein Punkt „links“ von einer Linie liegt?

Wie können wir testen, ob zwei Segmente sich schneiden?

Wie testen wir, ob ein Punkt innerhalb einer Fläche liegt?

Kapitel 1.1.1 – Punkte und Linien

Punkt-Gerade-Test



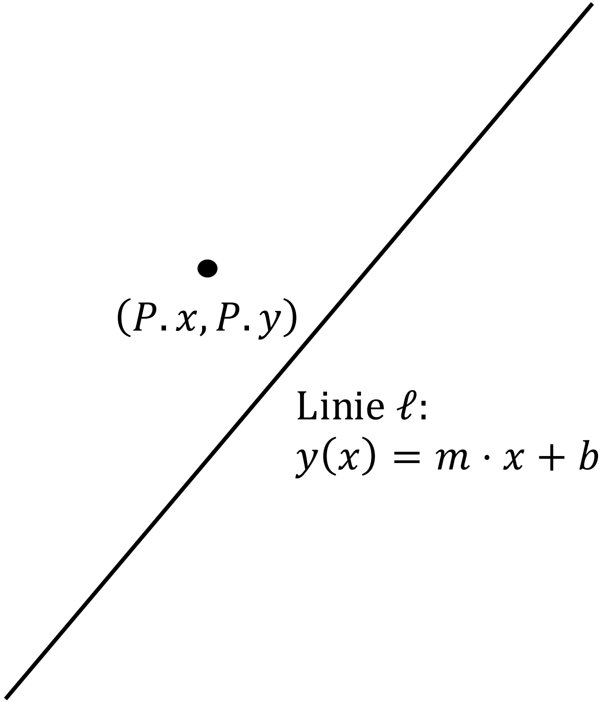
Linie ℓ:
$\ell.x = \text{const}$

Wie testet man, ob der Punkt P links/auf/recht von ℓ liegt?

- $P.x < \ell.x \Rightarrow P$ liegt links von ℓ
- $P.x = \ell.x \Rightarrow P$ liegt auf ℓ
- $P.x > \ell.x \Rightarrow P$ liegt rechts von ℓ

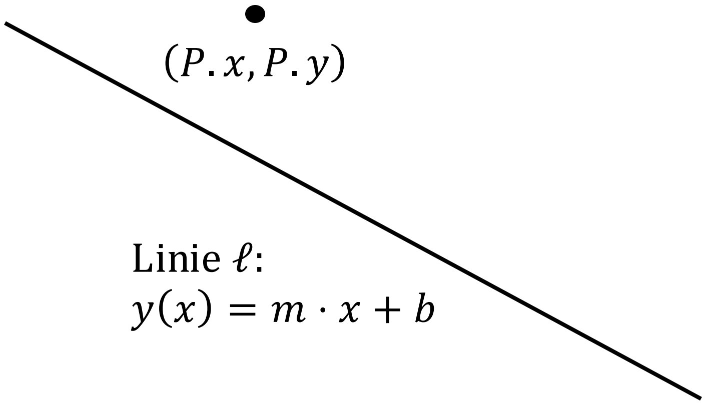
Punkt-Gerade-Test

Wie testet man, ob der Punkt P links/auf/recht von ℓ liegt?



Punkt-Gerade-Test

Wie testet man, ob der Punkt P links/auf/recht von ℓ liegt?



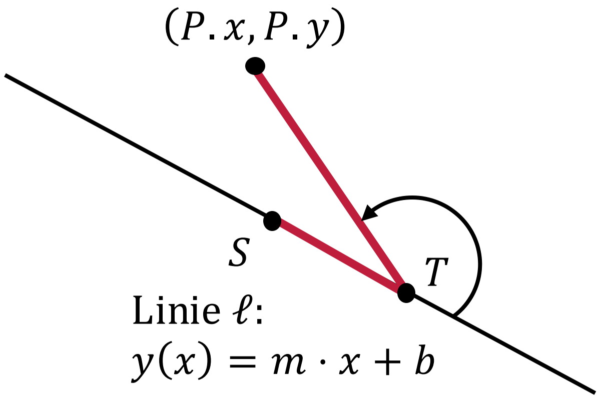
Punkt-Gerade-Test

Wie testet man, ob der Punkt P links/auf/recht von ℓ liegt?

$(P.x, P.y)$

Linie ℓ:
$y(x) = m \cdot x + b$

Punkt-Gerade-Test



Wie testet man, ob der Punkt P links/auf/recht von ℓ liegt?

Definition 1.1:
Ein *Punkt* P ist ein Element von $\mathbb{R}^2$.
Ein *Segment* (Strecke) ST ist die Linie von einem Punkt S zu einem Punkt T.

Zwei Segmente ST und TP (kurz STP) bilden einen *links-Knick*, wenn der gebildete Winkel gegen den Uhrzeigersinn verläuft. Ansonsten bilden sie einen *rechts-Knick*.

Seien S und T zwei Punkte auf einer Linie ℓ. Dann gilt:

- STP bildet einen links-Knick, dann liegt P links von ℓ.
- STP bildet einen rechts-Knick, dann liegt P rechts von ℓ.
- S, T, P sind *kolinear*, wenn P auf ℓ liegt.

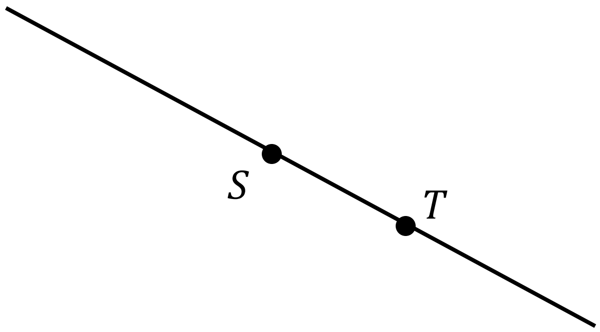
Punkt-Segment-Test

Das heißt, wir können uns auf Segmente und Punkte konzentrieren.

Betrachten wir den einfachen Fall: P liegt auf der Geraden durch S und T.

Wie beschreibt man die Gerade durch S und T?

Gerade durch zwei Punkte



Definition 1.2:
Jeder Punkt auf der Geraden durch S und T lässt sich beschreiben durch:

$$P(\lambda) = S + \lambda(T - S)$$

Bzw.

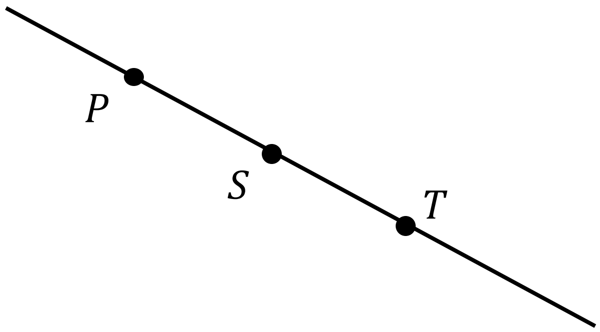
$$P(\lambda) = (1 - \lambda)S + \lambda T$$

Das ist eine ***Linearkombination***.
Ist $\lambda \in [0,1]$, spricht man von einer ***Konvexkombination***.

Lemma 1.3:
Ein Punkt P liegt auf der Geraden durch S und T, wenn ein $\lambda \in \mathbb{R}$ mit $P = (1 - \lambda)S + \lambda T$ existiert.

Gerade durch zwei Punkte

Lemma 1.3:
Ein Punkt P liegt auf der Geraden durch S und T, wenn ein $\lambda \in \mathbb{R}$ mit $P = (1-\lambda)S + \lambda T$ existiert.



Damit erhalten wir für einen Punkt P folgende Gleichungen:

$$P.x = (1-\lambda)S.x + \lambda T.x$$
$$P.y = (1-\lambda)S.y + \lambda T.y$$

Umgestellt nach λ:

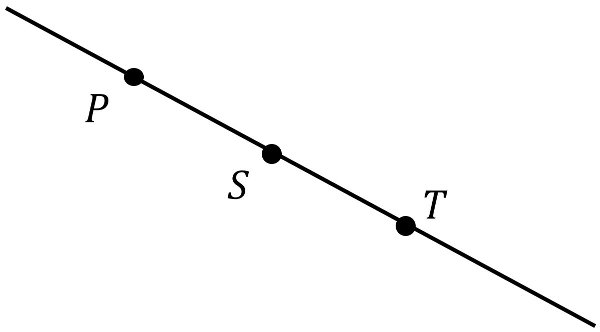
$$\lambda = \frac{P.x - S.x}{T.x - S.x}$$

$$\lambda = \frac{P.y - S.y}{T.y - S.y}$$

Also muss gelten:

$$(P.y - S.y)(T.x - S.x) - (P.x - S.x)(T.y - S.y) = 0$$

Gerade durch zwei Punkte



Lemma 1.3:
Ein Punkt P liegt auf der Geraden durch S und T, wenn ein $\lambda \in \mathbb{R}$ mit $P = (1-\lambda)S + \lambda T$ existiert.

Also muss gelten:

$$(P.y - S.y)(T.x - S.x) - (P.x - S.x)(T.y - S.y) = 0$$

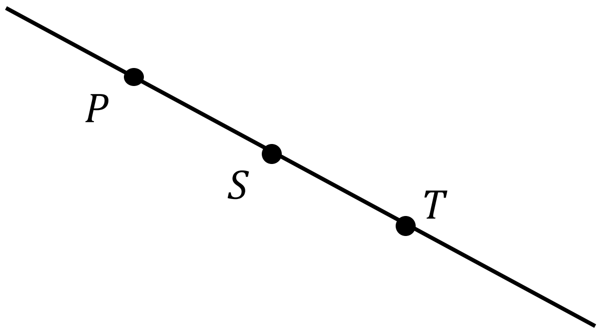
Das ist eine Determinante!

$$\det\begin{pmatrix} T.x - S.x & T.y - S.y \\ P.x - S.x & P.y - S.y \end{pmatrix}$$

Etwas anders dargstellt:

$$\det\begin{pmatrix} 1 & S.x & S.y \\ 0 & T.x - S.x & T.y - S.y \\ 0 & P.x - S.x & P.y - S.y \end{pmatrix}$$

Gerade durch zwei Punkte



Lemma 1.3:
Ein Punkt P liegt auf der Geraden durch S und T, wenn ein $\lambda \in \mathbb{R}$ mit $P = (1 - \lambda)S + \lambda T$ existiert.

Lineare Algebra liefert:

$$\det\begin{pmatrix} 1 & S.x & S.y \\ 0 & T.x - S.x & T.y - S.y \\ 0 & P.x - S.x & P.y - S.y \end{pmatrix} = \det\begin{pmatrix} 1 & S.x & S.y \\ 1 & T.x & T.y \\ 1 & P.x & P.y \end{pmatrix}$$

Korollar 1.4:
Ein Punkt P liegt auf der Geraden durch S und T, wenn

$$\det\begin{pmatrix} 1 & S.x & S.y \\ 1 & T.x & T.y \\ 1 & P.x & P.y \end{pmatrix} = 0.$$

Beispiele

$P_1 = (-1, 2)$

$P_2 = (2, 2)$

$T = (1,1)$

$P_3 = (3, 1)$

$S = (0,0)$

$$\det\begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 1 \\ 1 & -1 & 2 \end{pmatrix} = 2 + 1 = 3$$

$$\det\begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 1 \\ 1 & 2 & 2 \end{pmatrix} = 2 - 2 = 0$$

$$\det\begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 1 \\ 1 & 3 & 1 \end{pmatrix} = 1 - 3 = -2$$

Punkte und Segmente

Definition 1.5:
Seien $P, S, T \in \mathbb{R}^2$. Wir definieren

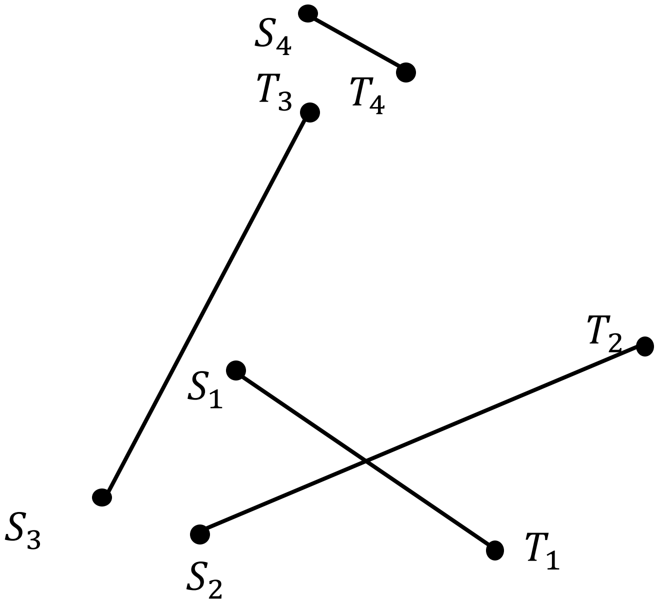
$$ccw(S, T, P) := \det \begin{pmatrix} 1 & S.x & S.y \\ 1 & T.x & T.y \\ 1 & P.x & P.y \end{pmatrix}$$

Theorem 1.6:
Seien $P, S, T \in \mathbb{R}^2$. Dann ist STP

- ein links-Knick, wenn $ccw(S, T, P) > 0$
- ein rechts-Knick, wenn $ccw(S, T, P) < 0$
- kolinear, wenn $ccw(S, T, P) = 0$

Kapitel 1.1.2 – Schneiden von Segmenten

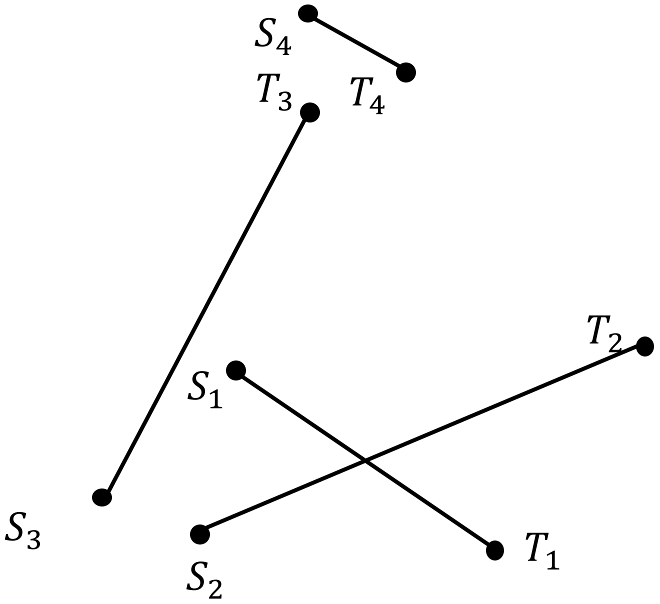
Segmente



Beobachtung 1:
Wenn sich zwei Segmente schneiden, dann müssen die Endpunkte der jeweils anderen Linie auf unterschiedlichen Seiten liegen!

Beobachtung 2:
Es reicht nicht, nur eine Linie zu prüfen!

Segmente



Theorem 1.7:
Zwei Segmente S_1T_1 und S_2T_2 schneiden sich genau dann, wenn:

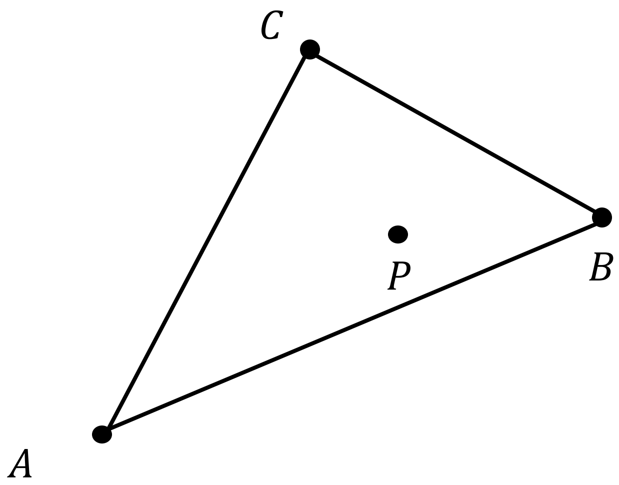
1. $ccw(S_1, T_1, S_2) \cdot ccw(S_1, T_1, T_2) \leq 0$ und
2. $ccw(S_2, T_2, S_1) \cdot ccw(S_2, T_2, T_1) \leq 0$

Etwas stärkere Bedinung fordert strikt kleiner 0, falls kolineare Punkte auftreten.

Auch zu berücksichtigen:
Degenerierte Segmente! ($S = T$)

Kapitel 1.1.3 – Container-Test

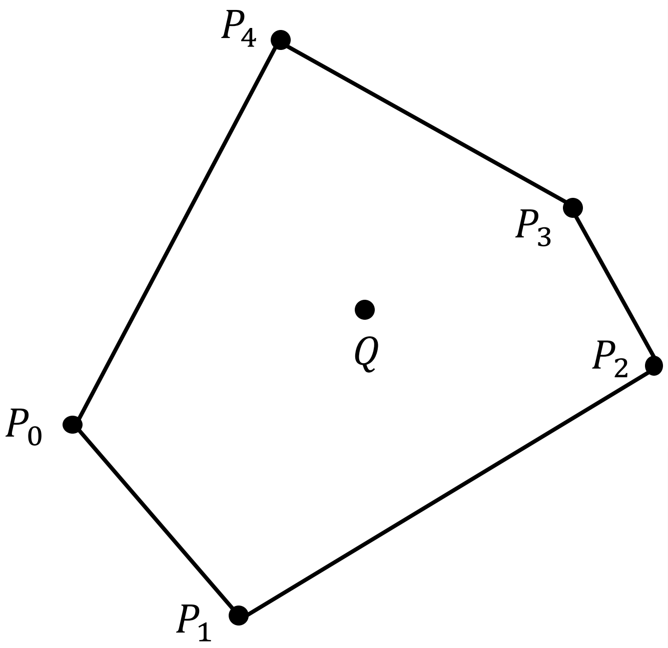
Dreiecke



Lemma 1.8:
Sei ABC ein Dreieck und P ein Punkt. P liegt genau dann innerhalb von ABC, wenn

1. $ccw(A, B, P) \geq 0$,
2. $ccw(B, C, P) \geq 0$ und
3. $ccw(C, A, P) \geq 0$

Konvexe Polygone



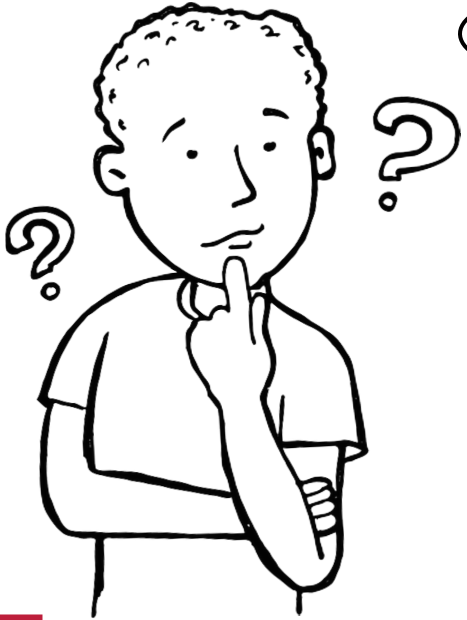
Definition 1.9 (Polygone):
Sei $P := (P_0, \ldots, P_{n-1})$ eine Folge von n Punkten. Dann heißt P *einfaches Polygon*, wenn für jedes Paar P_i, P_j mit $j \notin \{i-1,\ i,\ i+1\}$ die Segmente P_iP_{i+1} und P_jP_{j+1} sich nicht schneiden.

P heißt *konvexes* Polygon, wenn jedes Tripel P_i, P_{i+1}, P_{i+2} ein links-Knick bildet (jeder Innenwinkel ist als maximal 180°).

Achtung: Die Indizes sind immer mod n zu verstehen.

Lemma 1.10:
Sei $P := (P_0, \ldots, P_{n-1})$ ein konvexes Polygon und Q ein Punkt. Q liegt genau dann innerhalb von P, wenn für alle Paare P_i, P_{i+1} gilt: $ccw(P_i, P_{i+1}, Q) \geq 0$

Nächstes Kapitel



Wie funktioniert
das bei beliebigen
Formen?

Was passiert, wenn
ich mehrere Punkte
zu testen habe?

Was kann man tun,
wenn ganz viele Flächen
zur Verfügung stehen?