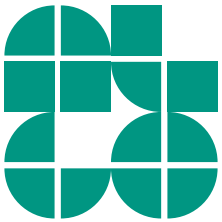
Vorlesung Algorithmische Geometrie

Geradenarrangements und Dualität

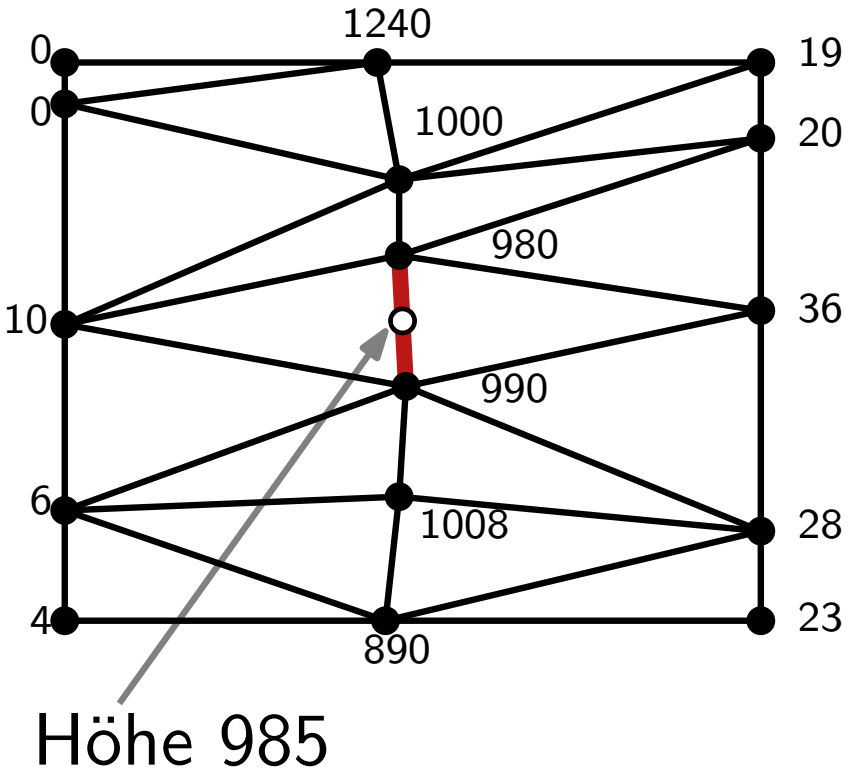
LEHRSTUHL FÜR ALGORITHMIK I · INSTITUT FÜR THEORETISCHE INFORMATIK · FAKULTÄT FÜR INFORMATIK

Martin Nöllenburg

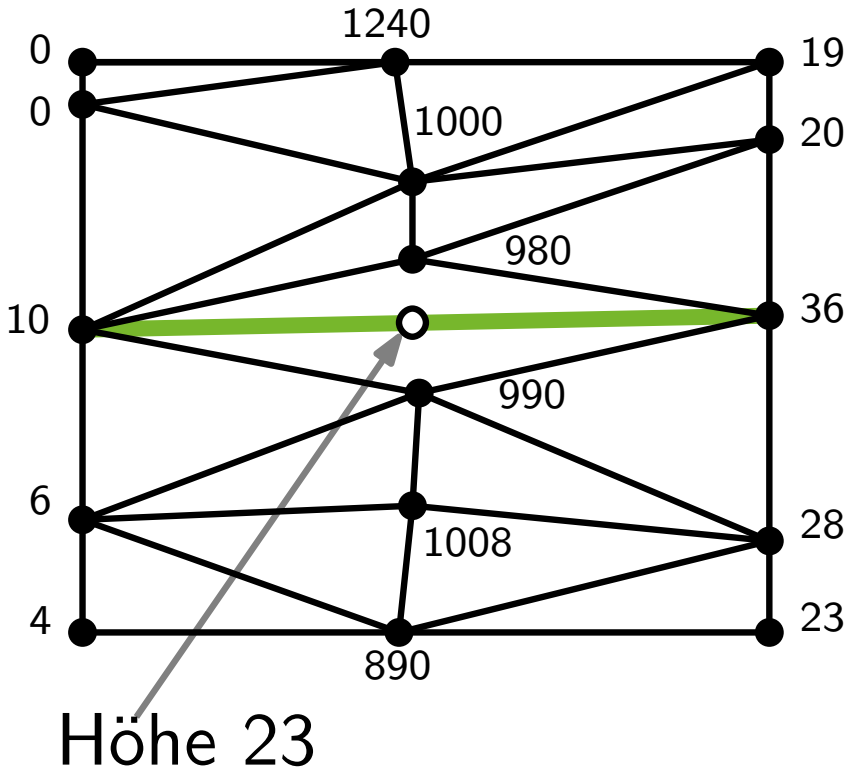
14.06.2011



Erinnerung: Höheninterpolation



Höhe 985



Höhe 23

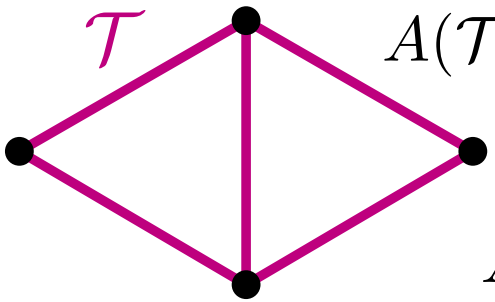
Ziel: Vermeide zu schmale Dreiecke, d.h. maximiere die kleinsten Dreieckswinkel!

Winkeloptimale Triangulierungen

Def.: Sei $P \subset \mathbb{R}^2$ eine Punktemenge, $\mathcal{T}$ eine Triangulierung von P und m die Anzahl der Dreiecke. Dann ist $A(\mathcal{T}) = (\alpha_1, \ldots, \alpha_{3m})$ der **Winkelvektor** von $\mathcal{T}$ mit den sortierten Dreieckswinkeln $\alpha_1 \leq \cdots \leq \alpha_{3m}$.

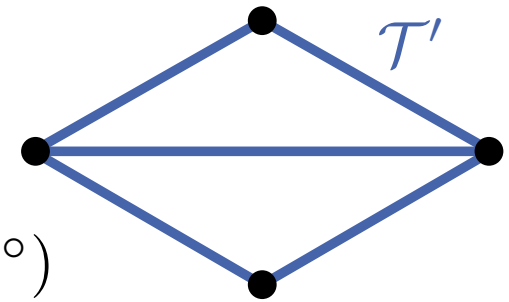
Für zwei Triangulierungen $\mathcal{T}$ und $\mathcal{T}'$ von P definiere die Ordnung $A(\mathcal{T}) > A(\mathcal{T}')$ als lexikographische Ordnung.

$\mathcal{T}$ heißt **winkeloptimal**, falls $A(\mathcal{T}) \geq A(\mathcal{T}')$ für alle Triangulierungen $\mathcal{T}'$ von P.



$$A(\mathcal{T}) = (60°, 60°, 60°, 60°, 60°, 60°)$$

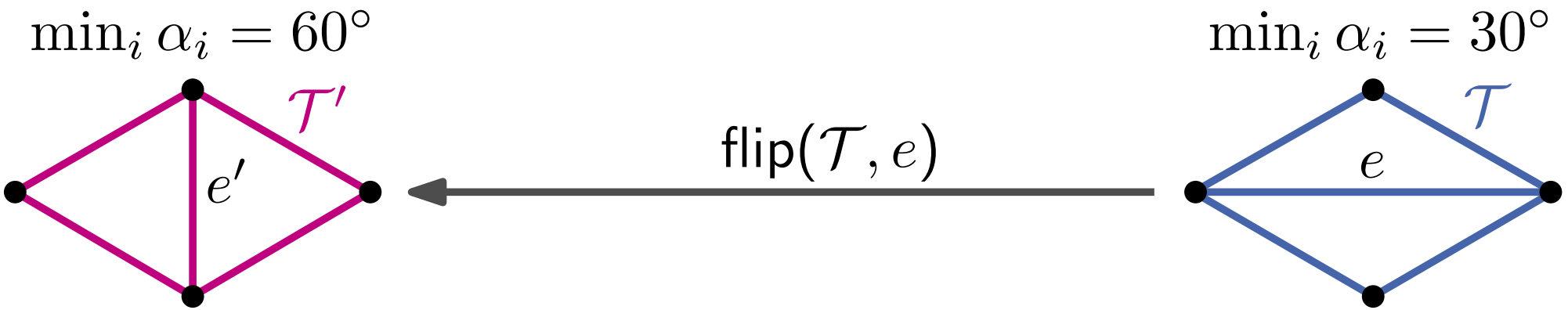
$$A(\mathcal{T}') = (30°, 30°, 30°, 30°, 120°, 120°)$$



Kantenflips

Def.: Sei $\mathcal{T}$ eine Triangulierung. Eine Kante e von $\mathcal{T}$ heißt **unzulässig**, wenn der kleinste Winkel der zu e inzidenten Dreiecke durch einen Kantenflip größer wird.

Beob.: Sei e eine unzulässige Kante von $\mathcal{T}$ und $\mathcal{T}' = \text{flip}(\mathcal{T}, e)$. Dann gilt $A(\mathcal{T}') > A(\mathcal{T})$.



Zwischenbilanz

Es gilt: Jede winkeloptimale Triangulierung ist zulässig.

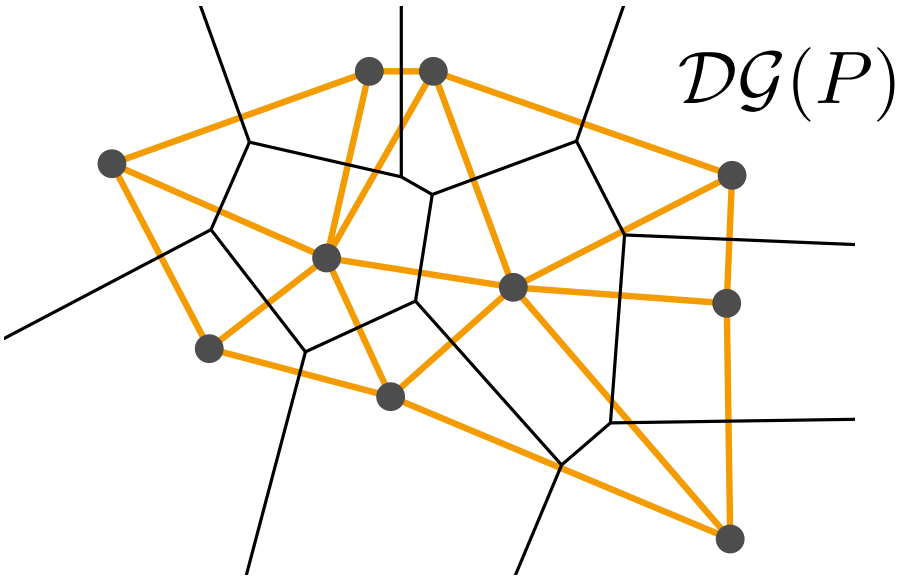
Aber ist jede zulässige Triangulierung auch winkeloptimal?

Die Delaunay-Triangulierung

Sei $\mathrm{Vor}(P)$ das Voronoi-Diagramm einer Punktmenge P.

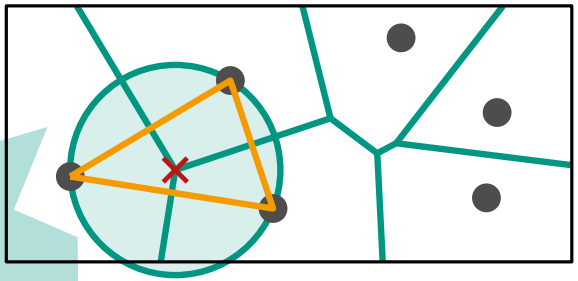
Def.: Der Graph $\mathcal{G} = (P, E)$ mit
$E = \{pq \mid \mathcal{V}(p) \text{ und } \mathcal{V}(q) \text{ sind benachbart}\}$
heißt **Dualgraph** von $\mathrm{Vor}(P)$.

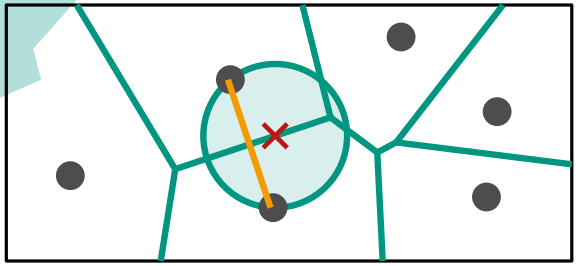
Def.: Die geradlinige Zeichnung von $\mathcal{G}$ heißt **Delaunay-Graph** $\mathcal{DG}(P)$.



Charakterisierung

Satz über Voronoi-Diagramme:





- Ein Punkt q ist ein Voronoi-Knoten
  $\Leftrightarrow |C_P(q) \cap P| \geq 3$,
- der Bisektor $b(p_i, p_j)$ definiert eine Voronoi-Kante
  $\Leftrightarrow \exists q \in b(p_i, p_j)$ mit $C_P(q) \cap P = \{p_i, p_j\}$.

Satz 4: Sei P eine Menge von Punkten.

- Punkte p, q, r sind Knoten der gleichen Facette in $\mathcal{DG}(P)$ $\Leftrightarrow$ Kreis durch p, q, r ist leer
- Kante pq ist in $\mathcal{DG}(P)$
  $\Leftrightarrow$ es gibt einen leeren Kreis $C_{p,q}$ durch p und q

Satz 5: Sei P Punktmenge und $\mathcal{T}$ eine Triangulierung von P.
$\mathcal{T}$ ist Delaunay-Triangulierung
$\Leftrightarrow$ Umkreis jedes Dreiecks ist im Inneren leer.

Zulässigkeit und Delaunay-Triangulierungen

Satz 6: Sei P Punktmenge und $\mathcal{T}$ eine Triangulierung von P. $\mathcal{T}$ ist zulässig $\Leftrightarrow$ $\mathcal{T}$ ist Delaunay-Triangulierung.

Beweisskizze:

„$\Leftarrow$“ klar; nutze

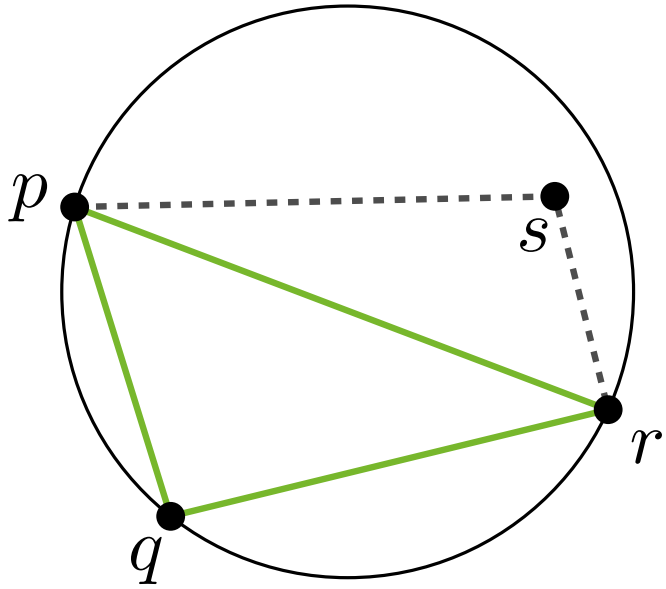
Lemma 1: Seien Δprq und Δpqs zwei Dreiecke in $\mathcal{T}$ und C der Umkreis von Δprq. Dann ist $\overline{pq}$ unzulässig genau dann wenn $s \in \text{int}(C)$.

Zulässigkeit und Delaunay-Triangulierungen

Satz 6: Sei P Punktmenge und $\mathcal{T}$ eine Triangulierung von P.
$\mathcal{T}$ ist zulässig $\Leftrightarrow$ $\mathcal{T}$ ist Delaunay-Triangulierung.

Beweisskizze:

„$\Rightarrow$“

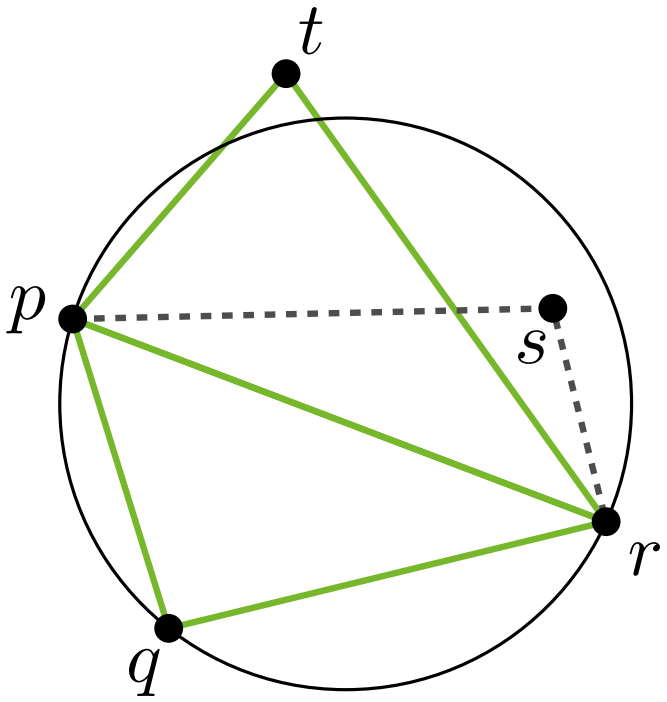


Zulässigkeit und Delaunay-Triangulierungen

Satz 6: Sei P Punktmenge und $\mathcal{T}$ eine Triangulierung von P.
$\mathcal{T}$ ist zulässig $\Leftrightarrow$ $\mathcal{T}$ ist Delaunay-Triangulierung.

Beweisskizze:

„$\Rightarrow$“

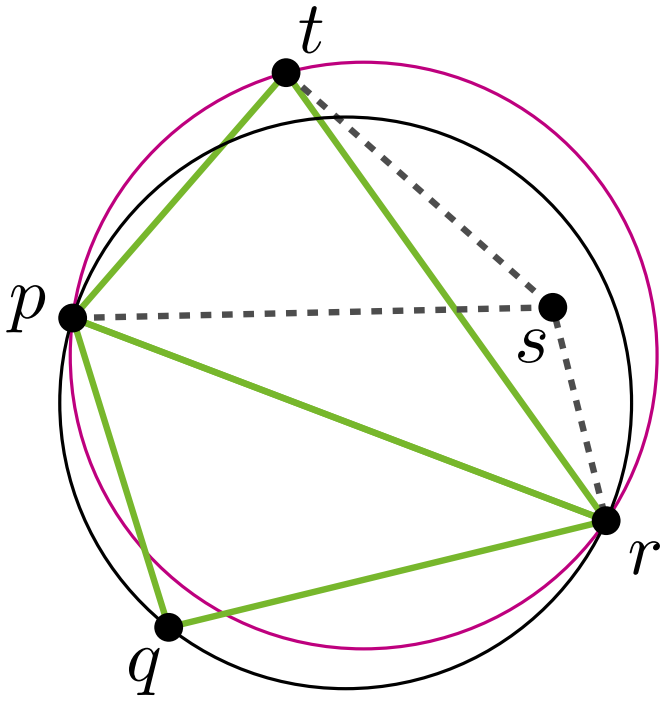


Zulässigkeit und Delaunay-Triangulierungen

Satz 6: Sei P Punktmenge und $\mathcal{T}$ eine Triangulierung von P.
$\mathcal{T}$ ist zulässig $\Leftrightarrow$ $\mathcal{T}$ ist Delaunay-Triangulierung.

Beweisskizze:

„$\Rightarrow$“



Zulässigkeit und Delaunay-Triangulierungen

Satz 6: Sei P Punktmenge und $\mathcal{T}$ eine Triangulierung von P. $\mathcal{T}$ ist zulässig $\Leftrightarrow$ $\mathcal{T}$ ist Delaunay-Triangulierung.

Beob.: Ist P in allgemeiner Lage ist $\mathcal{DG}(P)$ eindeutig

Zulässigkeit und Delaunay-Triangulierungen

Satz 6: Sei P Punktmenge und $\mathcal{T}$ eine Triangulierung von P.
$\mathcal{T}$ ist zulässig $\Leftrightarrow$ $\mathcal{T}$ ist Delaunay-Triangulierung.

Beob.: Ist P in allgemeiner Lage ist $\mathcal{DG}(P)$ eindeutig
$\Rightarrow$ zulässige Triangulierung ist eindeutig

Zulässigkeit und Delaunay-Triangulierungen

Satz 6: Sei P Punktmenge und $\mathcal{T}$ eine Triangulierung von P.
$\mathcal{T}$ ist zulässig $\Leftrightarrow$ $\mathcal{T}$ ist Delaunay-Triangulierung.

Beob.: Ist P in allgemeiner Lage ist $\mathcal{DG}(P)$ eindeutig
$\Rightarrow$ zulässige Triangulierung ist eindeutig
wissen: $\mathcal{T}$ winkeloptimal $\Rightarrow$ $\mathcal{T}$ zulässig

Zulässigkeit und Delaunay-Triangulierungen

Satz 6: Sei P Punktmenge und $\mathcal{T}$ eine Triangulierung von P. $\mathcal{T}$ ist zulässig $\Leftrightarrow$ $\mathcal{T}$ ist Delaunay-Triangulierung.

Beob.: Ist P in allgemeiner Lage ist $\mathcal{DG}(P)$ eindeutig

$\Rightarrow$ zulässige Triangulierung ist eindeutig

wissen: $\mathcal{T}$ winkeloptimal $\Rightarrow$ $\mathcal{T}$ zulässig

$\Rightarrow$ $\mathcal{DG}(P)$ winkeloptimal!

Zulässigkeit und Delaunay-Triangulierungen

Satz 6: Sei P Punktmenge und $\mathcal{T}$ eine Triangulierung von P. $\mathcal{T}$ ist zulässig $\Leftrightarrow$ $\mathcal{T}$ ist Delaunay-Triangulierung.

Beob.: Ist P in allgemeiner Lage ist $\mathcal{DG}(P)$ eindeutig

$\Rightarrow$ zulässige Triangulierung ist eindeutig

wissen: $\mathcal{T}$ winkeloptimal $\Rightarrow$ $\mathcal{T}$ zulässig

$\Rightarrow$ $\mathcal{DG}(P)$ winkeloptimal!

Ist P *nicht* in allgemeiner Lage, so ist zumindest für *jede* Triangulierung einer „großen“ Facette von $\mathcal{DG}(P)$ der *minimale* Winkel gleich (Übung!).

Zusammenfassung

Satz 7: Für n beliebige Punkte kann in $O(n \log n)$ Zeit eine Delaunay-Triangulierung berechnet werden.
(Voronoi-Diag. + Triangulierung „großer“ Facetten)

Zusammenfassung

Satz 7: Für n beliebige Punkte kann in $O(n \log n)$ Zeit eine Delaunay-Triangulierung berechnet werden.
(Voronoi-Diag. + Triangulierung „großer“ Facetten)

Korollar: Für n Punkte in allgemeiner Lage kann in $O(n \log n)$ Zeit eine winkeloptimale Triangulierung berechnet werden.
Sind die Punkte nicht in allgemeiner Lage, lässt sich zumindest eine Triangulierung mit maximalem kleinsten Winkel in $O(n \log n)$ Zeit berechnen.

Zusammenfassung

Satz 7: Für n beliebige Punkte kann in $O(n \log n)$ Zeit eine Delaunay-Triangulierung berechnet werden.
(Voronoi-Diag. + Triangulierung „großer“ Facetten)

Korollar: Für n Punkte in allgemeiner Lage kann in $O(n \log n)$ Zeit eine winkeloptimale Triangulierung berechnet werden.
Sind die Punkte nicht in allgemeiner Lage, lässt sich zumindest eine Triangulierung mit maximalem kleinsten Winkel in $O(n \log n)$ Zeit berechnen.

Ausblick: Auch im allgemeinen Fall kann die winkeloptimale Triangulierung in $O(n \log n)$ Zeit berechnet werden.
[Mount, Saalfeld '88]

Diskussion

Gibt es alternative Ansätze für Triangulierungen zur Höheninterpolation?

Diskussion

Gibt es alternative Ansätze für Triangulierungen zur Höheninterpolation?

Statt der *datenunabhängigen* Delaunay-Triangulierung gibt es auch Ansätze für *datenabhängige* Triangulierungen, die jedoch von $\mathcal{DG}(P)$ ausgehen und dann Kantenflips durchführen. Rippa (1990) zeigte dass $\mathcal{DG}(P)$ die *roughness* unabhängig von den Höheninformationen minimiert.

Diskussion

Gibt es alternative Ansätze für Triangulierungen zur Höheninterpolation?

Statt der *datenunabhängigen* Delaunay-Triangulierung gibt es auch Ansätze für *datenabhängige* Triangulierungen, die jedoch von $\mathcal{DG}(P)$ ausgehen und dann Kantenflips durchführen. Rippa (1990) zeigte dass $\mathcal{DG}(P)$ die *roughness* unabhängig von den Höheninformationen minimiert.

Hat $\mathcal{DG}(P)$ weitere interessante Eigenschaften?

Diskussion

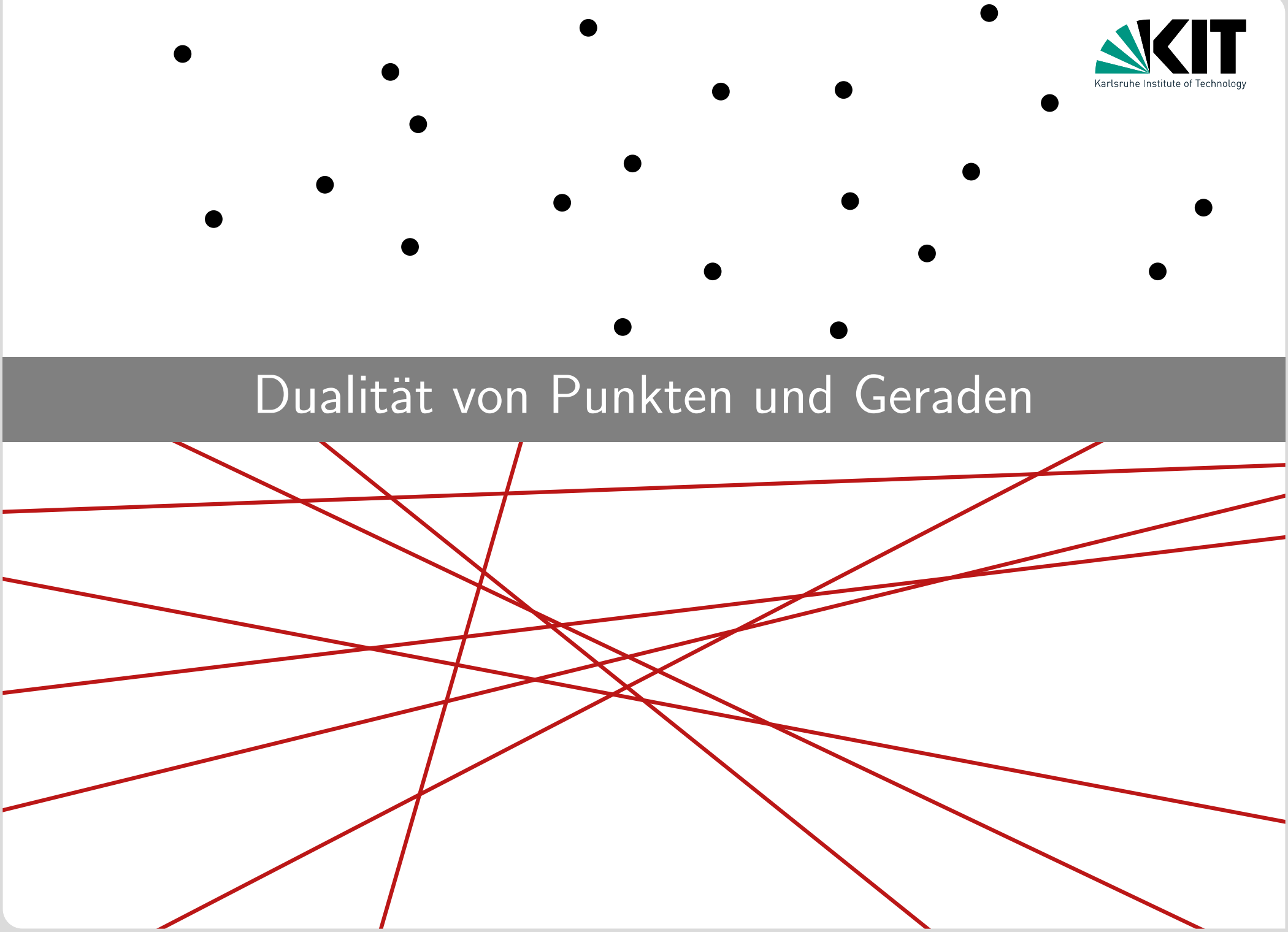
Gibt es alternative Ansätze für Triangulierungen zur Höheninterpolation?

Statt der *datenunabhängigen* Delaunay-Triangulierung gibt es auch Ansätze für *datenabhängige* Triangulierungen, die jedoch von $\mathcal{DG}(P)$ ausgehen und dann Kantenflips durchführen. Rippa (1990) zeigte dass $\mathcal{DG}(P)$ die *roughness* unabhängig von den Höheninformationen minimiert.

Hat $\mathcal{DG}(P)$ weitere interessante Eigenschaften?

Ja, der Delaunay-Graph enthält die Kanten anderer interessanter Graphen auf P (s. Übungsblatt). Es gilt z.B.

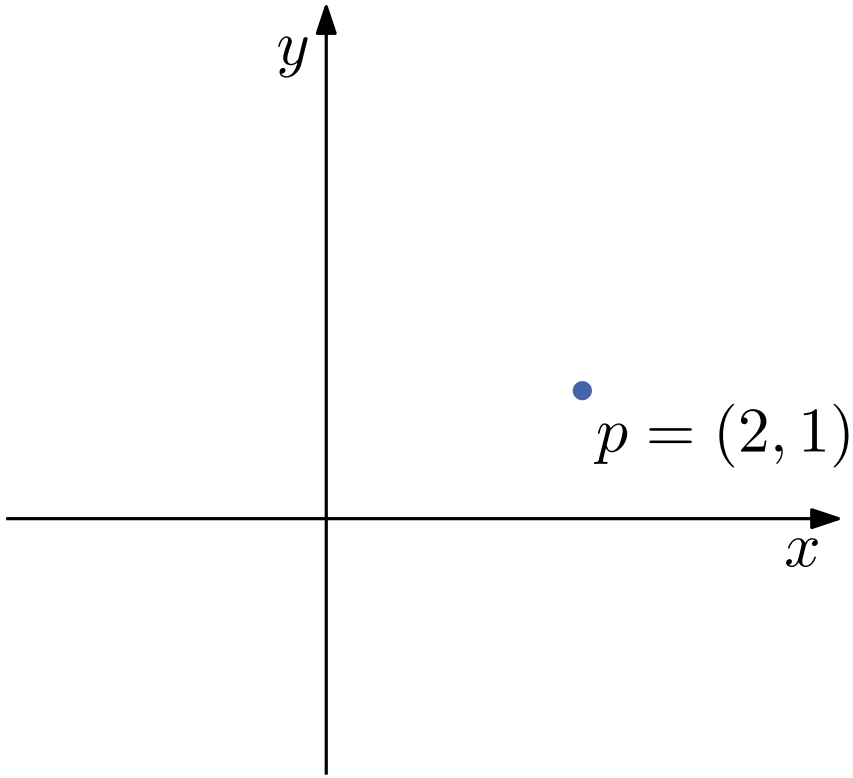
$$\text{EMST}(P) \subseteq \text{Gabriel-Graph}(P) \subseteq \mathcal{DG}(P)$$



Dualität von Punkten und Geraden

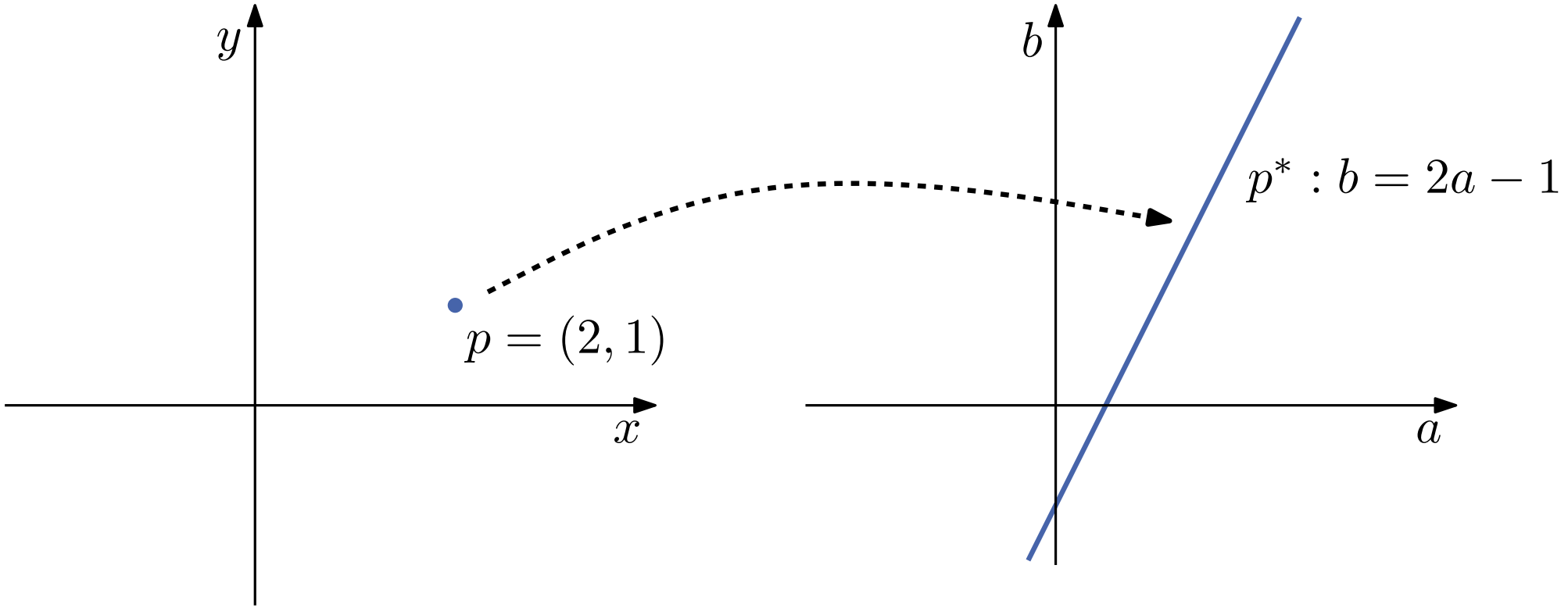
Dualitätsabbildung

Beob.: Bisher haben wir Dualität für planare Graphen gesehen. Auch Punkte und Geraden können **dual** zueinander sein.



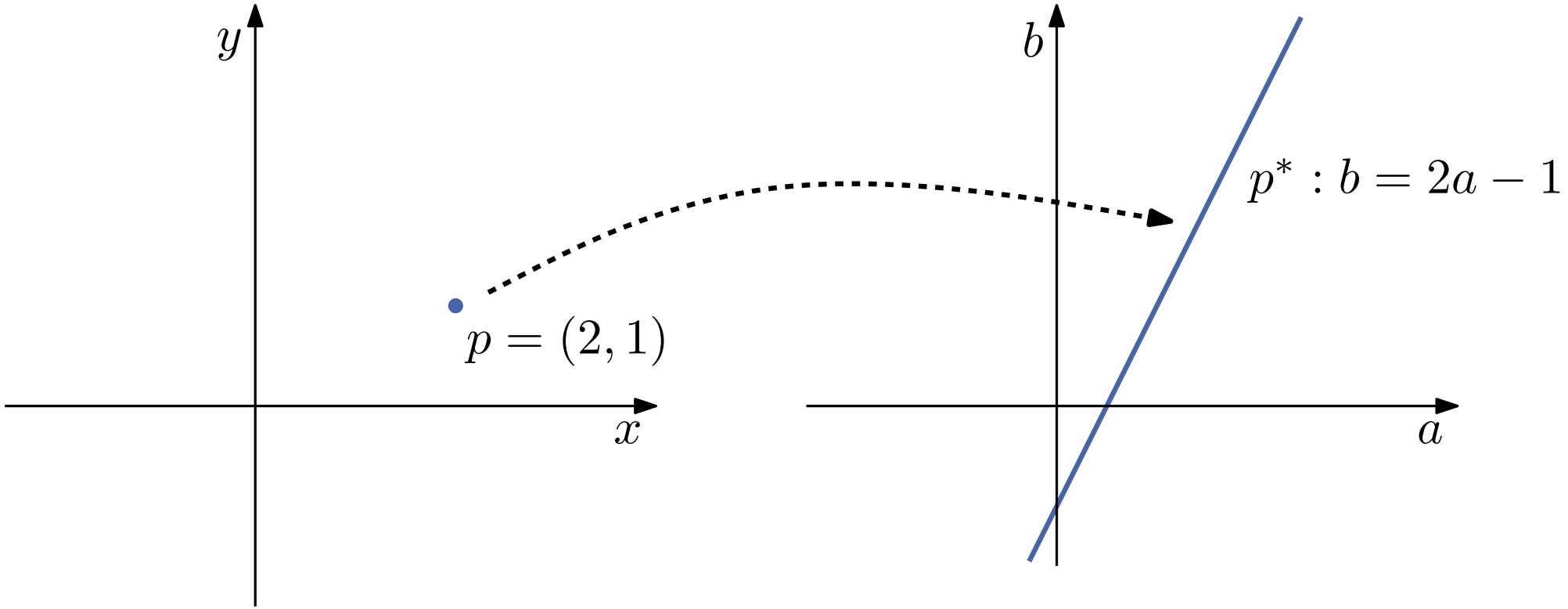
Dualitätsabbildung

Beob.: Bisher haben wir Dualität für planare Graphen gesehen.
Auch Punkte und Geraden können **dual** zueinander sein.



Dualitätsabbildung

Beob.: Bisher haben wir Dualität für planare Graphen gesehen.
Auch Punkte und Geraden können **dual** zueinander sein.

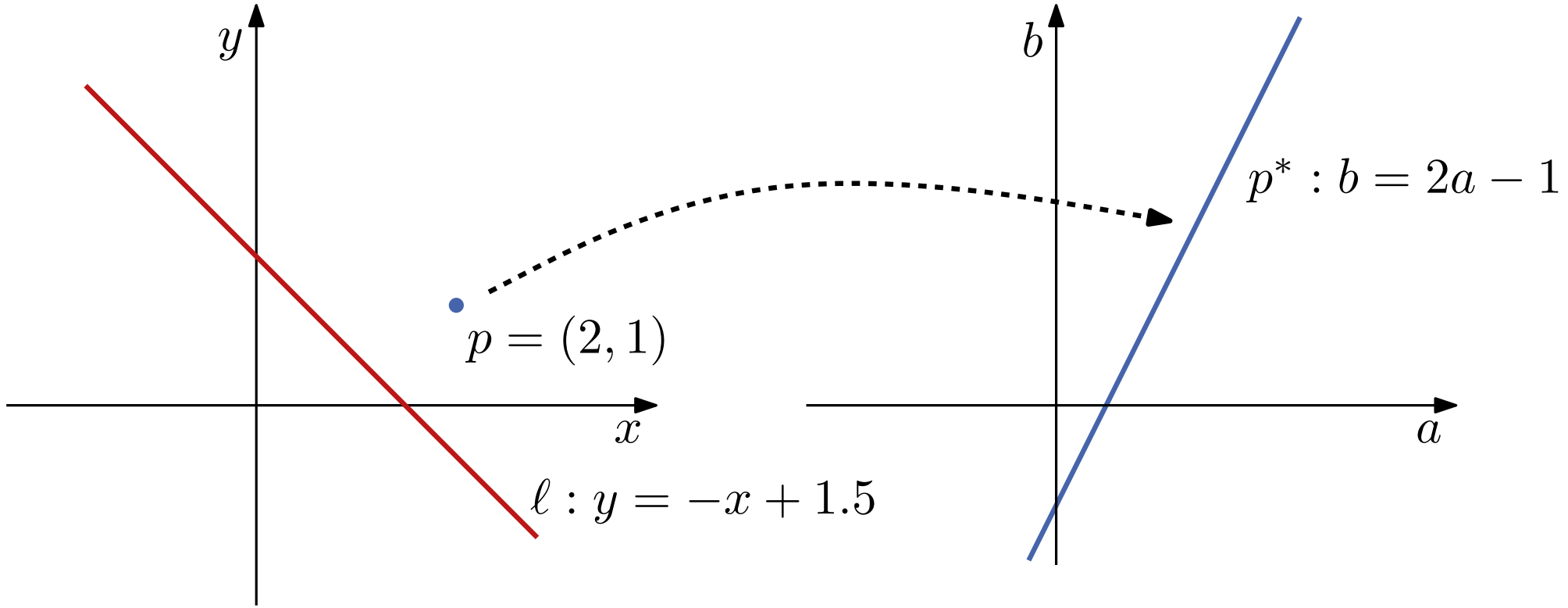


Def.: Die Dualitätsabbildung $(\cdot)^*$ ist definiert durch

$$p = (p_x, p_y) \quad \mapsto \quad p^* : b = p_x a - p_y$$

Dualitätsabbildung

Beob.: Bisher haben wir Dualität für planare Graphen gesehen.
Auch Punkte und Geraden können **dual** zueinander sein.

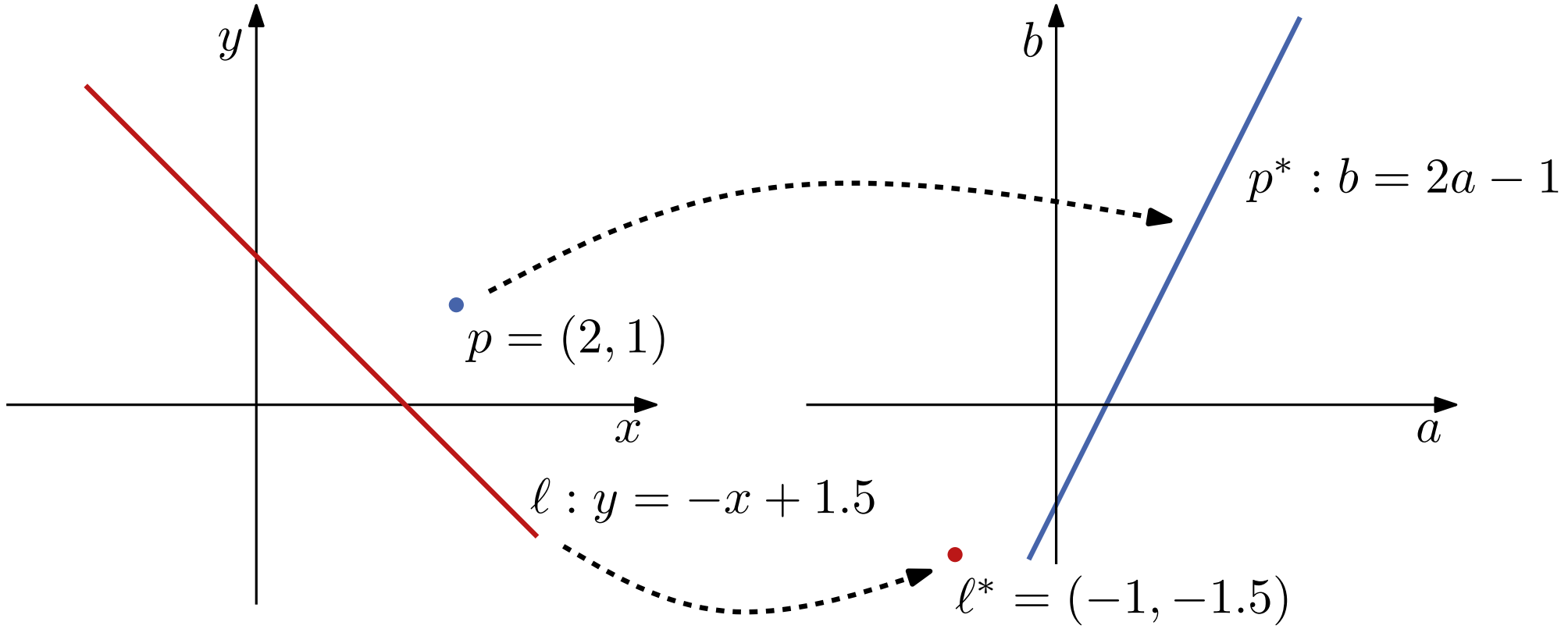


Def.: Die Dualitätsabbildung $(\cdot)^*$ ist definiert durch

$$p = (p_x, p_y) \quad \mapsto \quad p^* : b = p_x a - p_y$$

Dualitätsabbildung

Beob.: Bisher haben wir Dualität für planare Graphen gesehen.
Auch Punkte und Geraden können **dual** zueinander sein.

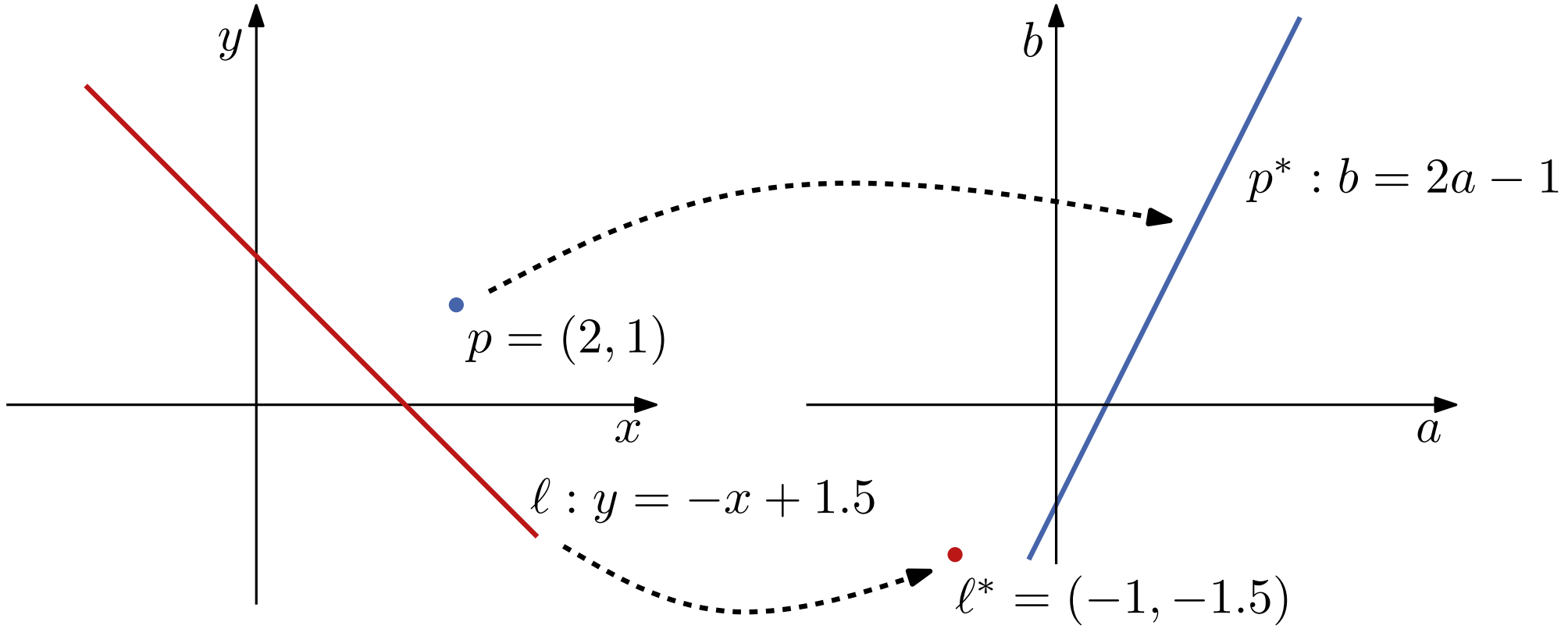


Def.: Die Dualitätsabbildung $(\cdot)^*$ ist definiert durch

$$p = (p_x, p_y) \quad \mapsto \quad p^* : b = p_x a - p_y$$

Dualitätsabbildung

Beob.: Bisher haben wir Dualität für planare Graphen gesehen.
Auch Punkte und Geraden können **dual** zueinander sein.



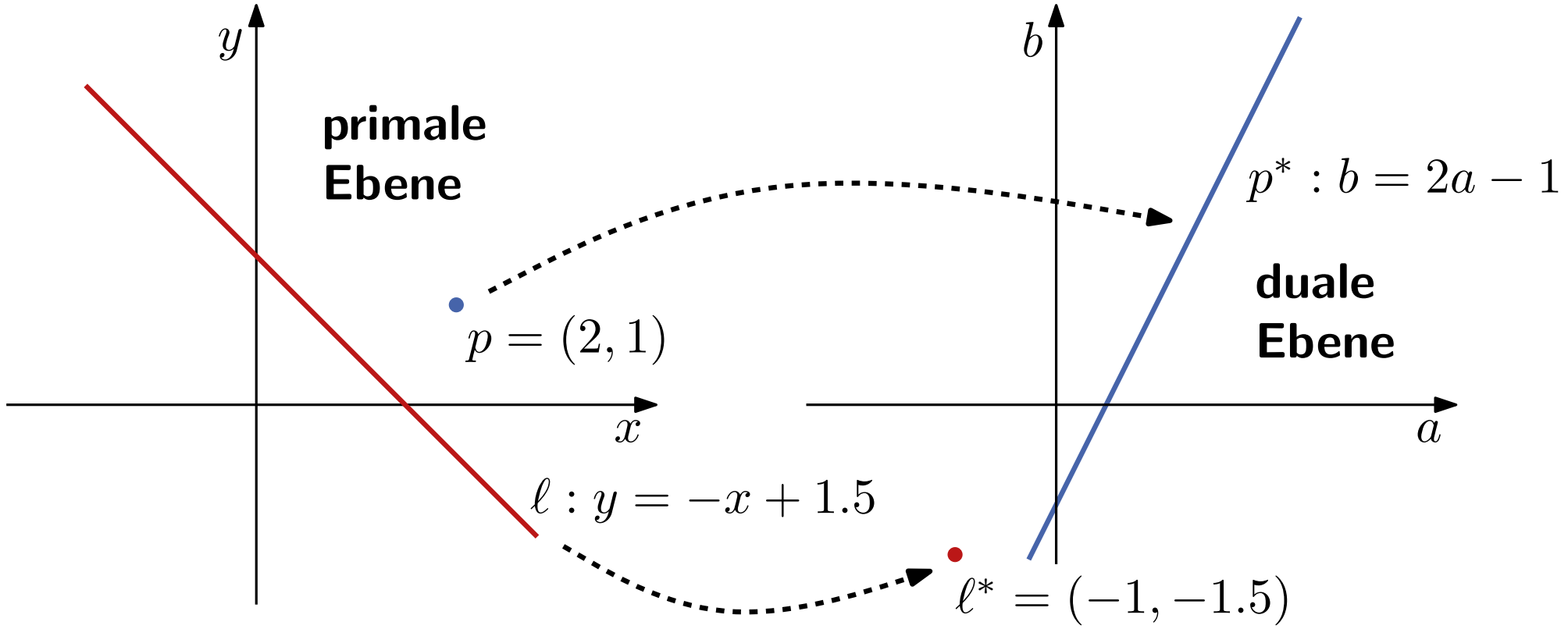
Def.: Die Dualitätsabbildung $(\cdot)^*$ ist definiert durch

$$p = (p_x, p_y) \quad \mapsto \quad p^* : b = p_x a - p_y$$

$$\ell : y = mx + c \quad \mapsto \quad \ell^* = (m, -c)$$

Dualitätsabbildung

Beob.: Bisher haben wir Dualität für planare Graphen gesehen.
Auch Punkte und Geraden können **dual** zueinander sein.



Def.: Die Dualitätsabbildung $(\cdot)^*$ ist definiert durch

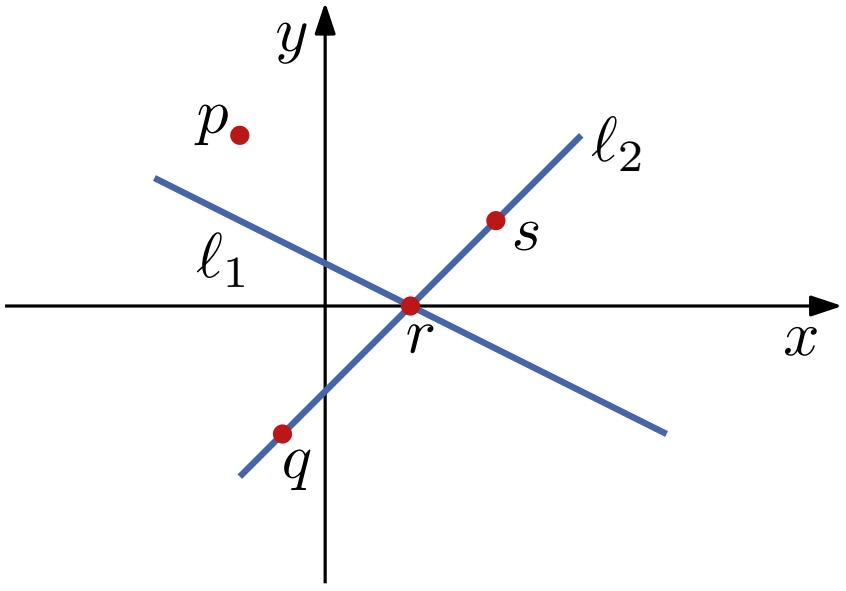
$$p = (p_x, p_y) \quad \mapsto \quad p^* : b = p_x a - p_y$$

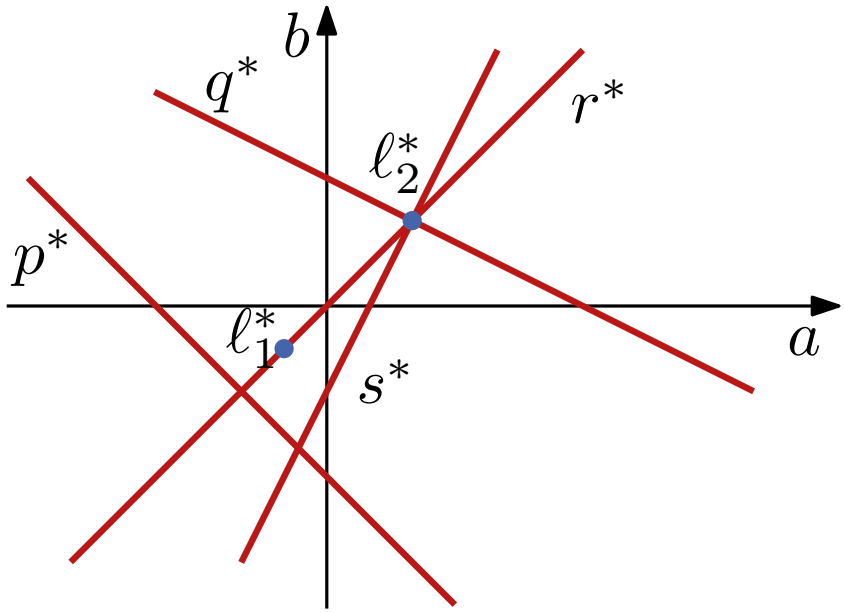
$$\ell : y = mx + c \quad \mapsto \quad \ell^* = (m, -c)$$

Eigenschaften

Lemma 1: Es gelten die folgenden Eigenschaften

- $(p^*)^* = p$ und $(\ell^*)^* = \ell$
- p liegt unter/auf/über $\ell \Leftrightarrow p^*$ läuft über/auf/unter ℓ^*
- ℓ_1 und ℓ_2 schneiden sich in p
  $\Leftrightarrow p^*$ geht durch ℓ_1^* und ℓ_2^*
- p_1, p_2, p_3 kollinear
  $\Leftrightarrow p_1^*, p_2^*, p_3^*$ schneiden sich in gemeinsamem Punkt





Eigenschaften

Lemma 1: Es gelten die folgenden Eigenschaften

- $(p^*)^* = p$ und $(\ell^*)^* = \ell$
- p liegt unter/auf/über $\ell \Leftrightarrow p^*$ läuft über/auf/unter ℓ^*
- ℓ_1 und ℓ_2 schneiden sich in p
  $\Leftrightarrow p^*$ geht durch ℓ_1^* und ℓ_2^*
- p_1, p_2, p_3 kollinear
  $\Leftrightarrow p_1^*, p_2^*, p_3^*$ schneiden sich in gemeinsamem Punkt

Wie sieht das duale Objekt zu einer Strecke $s = \overline{pq}$ aus?

Welche duale Beziehung gilt für eine Gerade ℓ, die s schneidet?

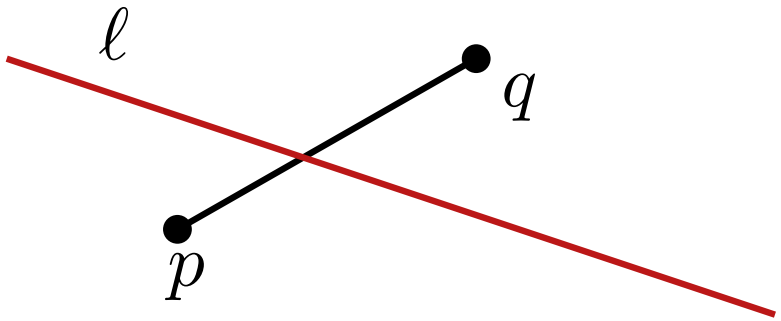
Eigenschaften

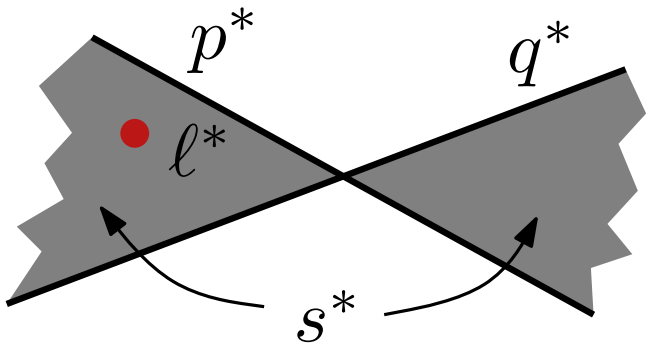
Lemma 1: Es gelten die folgenden Eigenschaften

- $(p^*)^* = p$ und $(\ell^*)^* = \ell$
- p liegt unter/auf/über $\ell \Leftrightarrow p^*$ läuft über/auf/unter ℓ^*
- ℓ_1 und ℓ_2 schneiden sich in p
  $\Leftrightarrow p^*$ geht durch ℓ_1^* und ℓ_2^*
- p_1, p_2, p_3 kollinear
  $\Leftrightarrow p_1^*, p_2^*, p_3^*$ schneiden sich in gemeinsamem Punkt

Wie sieht das duale Objekt zu einer Strecke $s = \overline{pq}$ aus?

Welche duale Beziehung gilt für eine Gerade ℓ, die s schneidet?





Anwendungen der Dualität

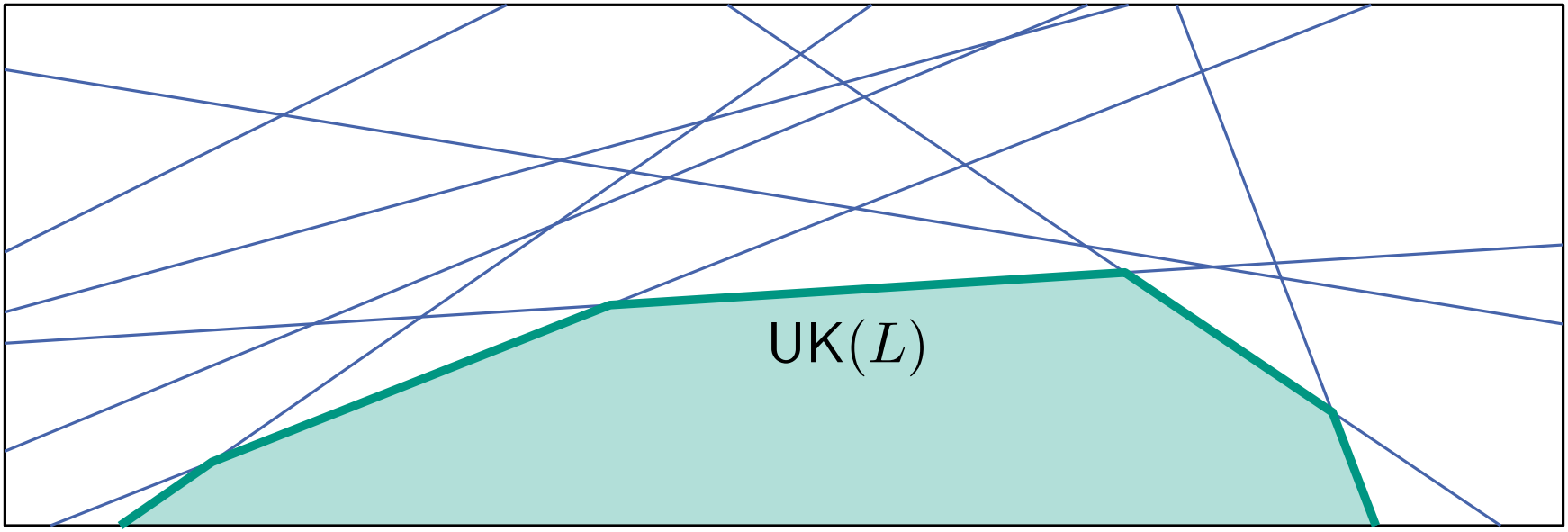
Karlsruhe Institute of Technology

Dualität macht geometrische Probleme weder leichter noch schwerer; sie bietet einen anderen (hilfreichen) Blickwinkel!

Wir werden zwei Beispiele sehen:

- untere/obere Konturen von Geradenarrangements
- kleinstes Dreieck in einer Punktmenge

Untere Kontur

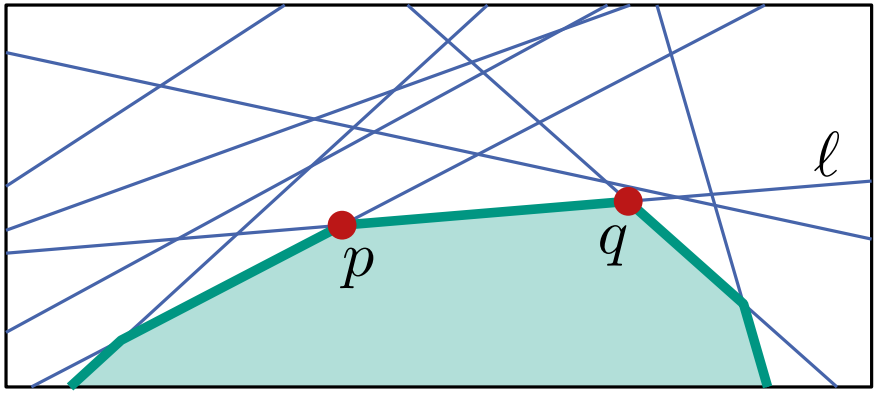


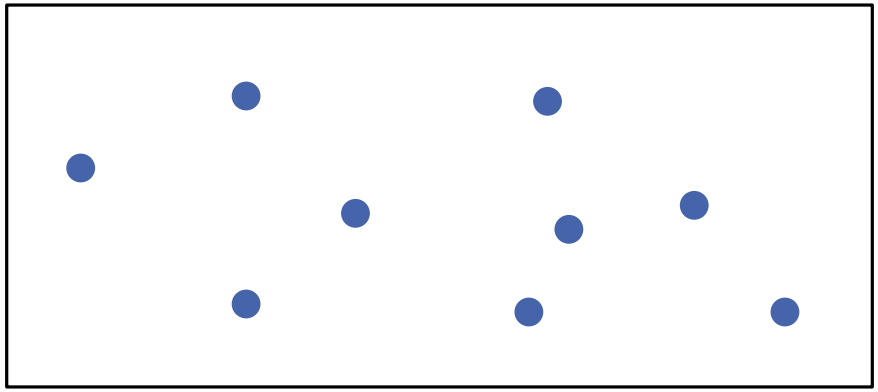
Def.: Für eine Menge L von Geraden ist die untere Kontur UK(L) von L die Menge aller Punkte aus $\cup_{\ell \in L} \ell$, die unterhalb aller Geraden aus L liegen.

Möglichkeiten zur Berechnung der unteren Kontur:

- Algorithmus `IntersectHalfplanes` aus 4. VL
- Betrachte das duale Problem für $L^* = \{\ell^* \mid \ell \in L\}$

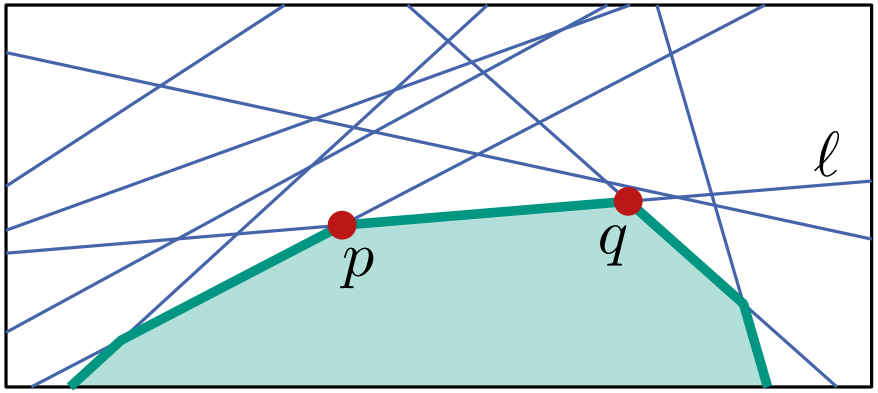
Kontur und Dualität

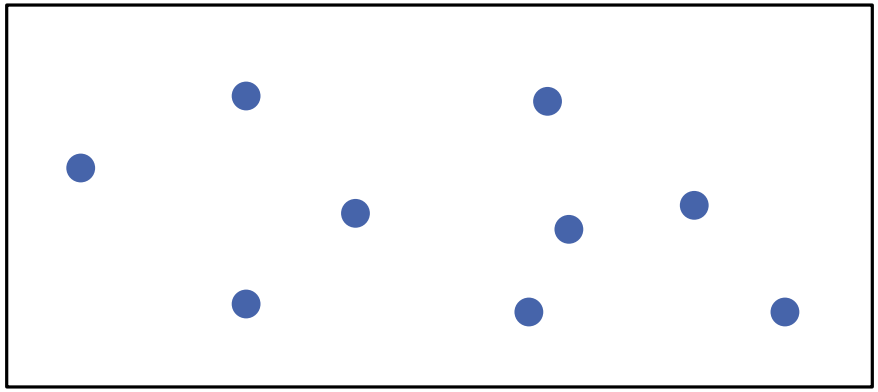




Wann erscheint ℓ als Strecke $\overline{pq}$ auf UK(L)?

Kontur und Dualität

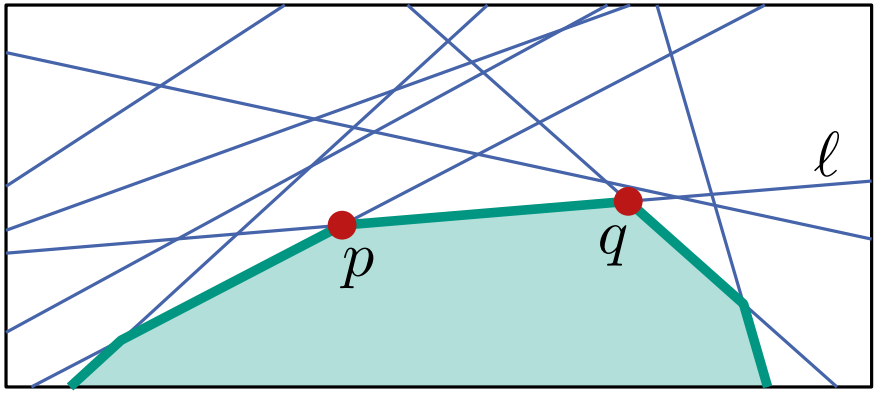


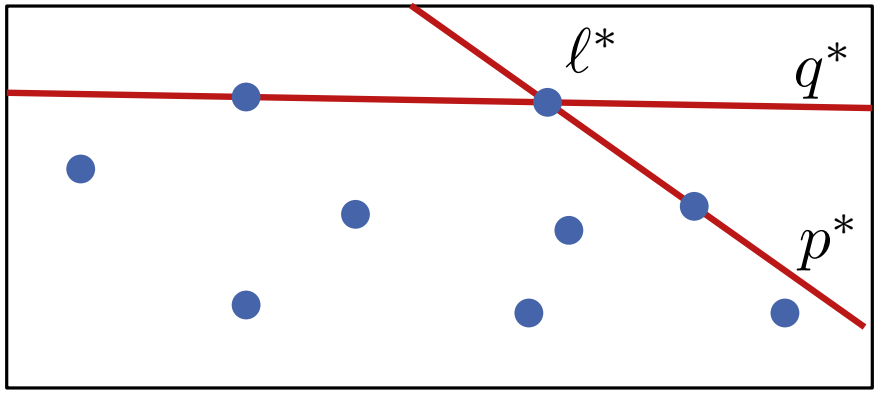


Wann erscheint ℓ als Strecke $\overline{pq}$ auf UK(L)?

- p und q liegen unterhalb aller Geraden in L

Kontur und Dualität

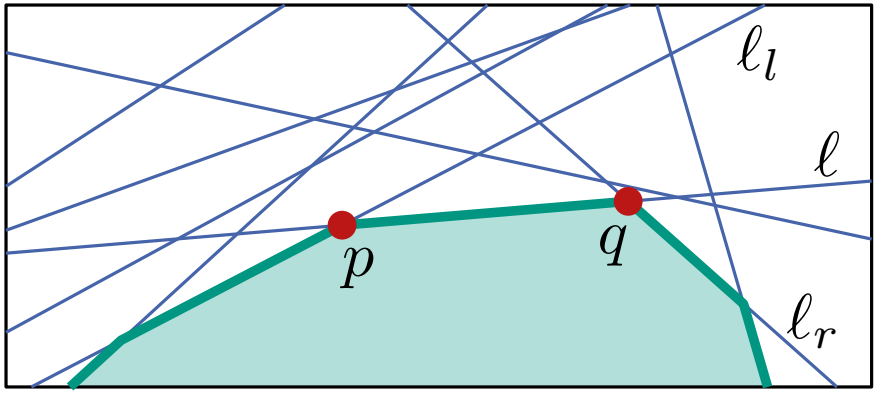


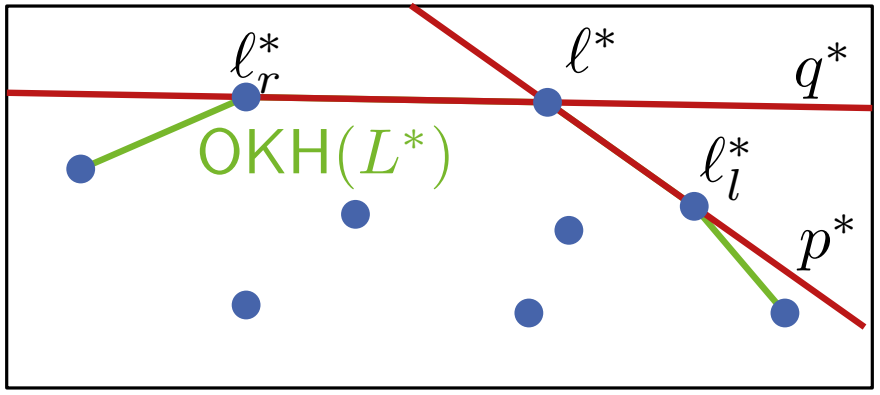


Wann erscheint ℓ als Strecke $\overline{pq}$ auf UK(L)?

- p und q liegen unterhalb aller Geraden in L
- p^* und q^* liegen oberhalb aller Punkte aus L^*

Kontur und Dualität

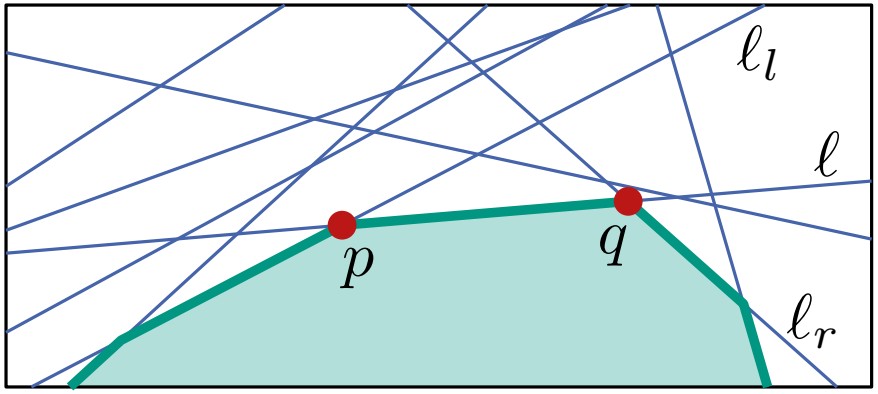


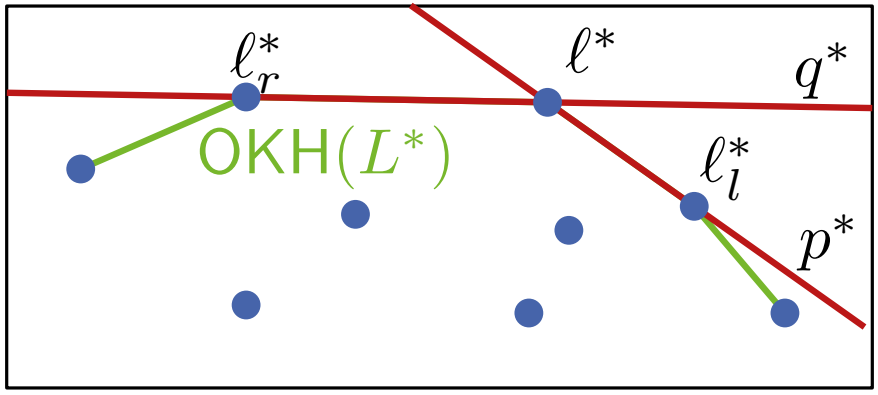


Wann erscheint ℓ als Strecke $\overline{pq}$ auf UK(L)?

- p und q liegen unterhalb aller Geraden in L
- p^* und q^* liegen oberhalb aller Punkte aus L^*
  $\Rightarrow$ liegen benachbart auf oberer konvexer Hülle OKH(L^*)
- Schnittpunkt von p^* und q^* ist ℓ^*, Knoten von OKH(L^*)

Kontur und Dualität





Wann erscheint ℓ als Strecke $\overline{pq}$ auf UK(L)?

- p und q liegen unterhalb aller Geraden in L
- p^* und q^* liegen oberhalb aller Punkte aus L^*
  $\Rightarrow$ liegen benachbart auf oberer konvexer Hülle OKH(L^*)
- Schnittpunkt von p^* und q^* ist ℓ^*, Knoten von OKH(L^*)

Lemma 2: Die Geraden auf UK(L) von links nach rechts entsprechen den Knoten auf OKH(L^*) von rechts nach links.

Berechnung der Kontur

- Algorithmus zur Berechnung der oberen konvexen Hülle mit Laufzeit $O(n \log n)$ (s. erste VL zu konvexer Hülle)

Berechnung der Kontur

- Algorithmus zur Berechnung der oberen konvexen Hülle mit Laufzeit $O(n \log n)$ (s. erste VL zu konvexer Hülle)
- Primale Geraden zu Punkten auf OKH(L^*) in umgekehrter Reihenfolge bilden UK(L)

Berechnung der Kontur

- Algorithmus zur Berechnung der oberen konvexen Hülle mit Laufzeit $O(n \log n)$ (s. erste VL zu konvexer Hülle)
- Primale Geraden zu Punkten auf OKH(L^*) in umgekehrter Reihenfolge bilden UK(L)
- Analog: obere Kontur von $L \mathrel{\hat{=}}$ untere konvexe Hülle von L^*

Berechnung der Kontur

- Algorithmus zur Berechnung der oberen konvexen Hülle mit Laufzeit $O(n \log n)$ (s. erste VL zu konvexer Hülle)
- Primale Geraden zu Punkten auf OKH(L^*) in umgekehrter Reihenfolge bilden UK(L)
- Analog: obere Kontur von $L \mathrel{\hat{=}}$ untere konvexe Hülle von L^*

Wie lässt sich das nutzen um den Schnitt von n Halbebenen zu berechnen?

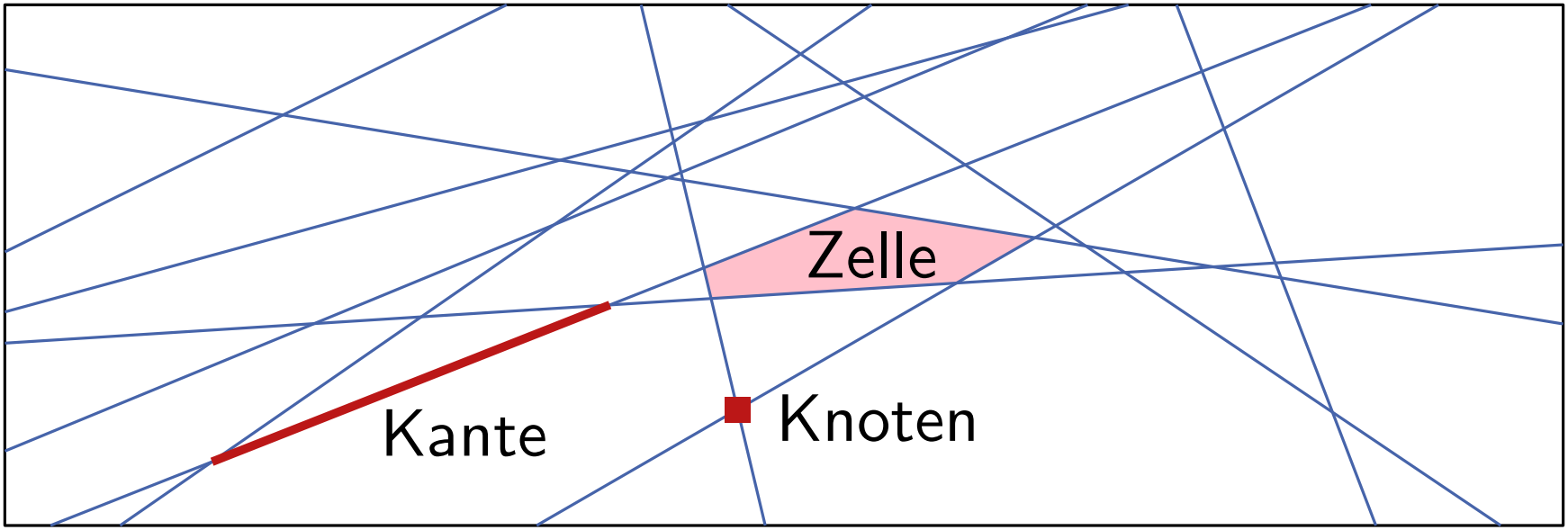
Berechnung der Kontur

- Algorithmus zur Berechnung der oberen konvexen Hülle mit Laufzeit $O(n \log n)$ (s. erste VL zu konvexer Hülle)
- Primale Geraden zu Punkten auf OKH(L^*) in umgekehrter Reihenfolge bilden UK(L)
- Analog: obere Kontur von $L \mathrel{\hat{=}}$ untere konvexe Hülle von L^*

Wie lässt sich das nutzen um den Schnitt von n Halbebenen zu berechnen?

- untere Kontur der von oben begrenzenden Geraden
- obere Kontur der von unten begrenzenden Geraden
- Schnitt der beiden konvexen Regionen (s. 4. VL)
- insgesamt $O(n \log n)$ Laufzeit

Geradenarrangements



Def.: Eine Menge L von Geraden definiert eine Unterteilung $\mathcal{A}(L)$ der Ebene (das **Geradenarrangement**) in Knoten, Kanten und Zellen (tlws. unbeschränkt).
$\mathcal{A}(L)$ heißt **einfach**, wenn keine drei Geraden durch einen Punkt gehen und keine zwei Geraden parallel sind.

Komplexität von $\mathcal{A}(L)$

Die kombinatorische Komplexität von $\mathcal{A}(L)$ ist die Zahl der Knoten, Kanten und Zellen. Es gilt:

Satz 1: Sei $\mathcal{A}(L)$ ein einfaches Arrangement von n Geraden. Dann hat $\mathcal{A}(L)$ $\binom{n}{2}$ Knoten, n^2 Kanten und $\binom{n}{2} + n + 1$ Zellen.

Komplexität von $\mathcal{A}(L)$

Die kombinatorische Komplexität von $\mathcal{A}(L)$ ist die Zahl der Knoten, Kanten und Zellen. Es gilt:

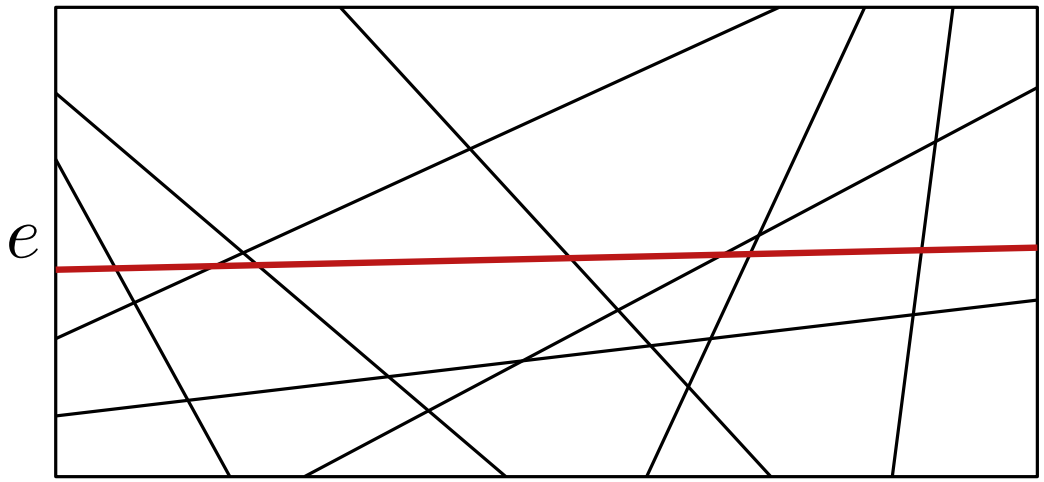
Satz 1: Sei $\mathcal{A}(L)$ ein einfaches Arrangement von n Geraden. Dann hat $\mathcal{A}(L)$ $\binom{n}{2}$ Knoten, n^2 Kanten und $\binom{n}{2} + n + 1$ Zellen.

Datenstruktur für $\mathcal{A}(L)$:

- füge bounding box aller Knoten hinzu (s. Übung)
  $\rightarrow$ planar eingebetteter Graph G
- doppelt-verkettete Kantenliste für G

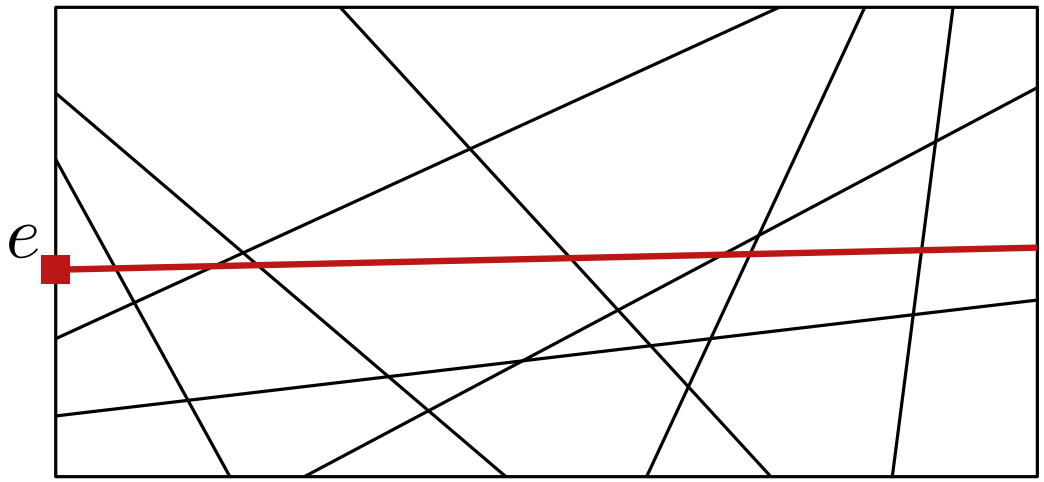
Konstruktion von $\mathcal{A}(L)$

Input: Geraden $L = \{\ell_1, \ldots, \ell_n\}$
Output: DCEL $\mathcal{D}$ für $\mathcal{A}(L)$
initialisiere $\mathcal{D}$ für bounding box B der Knoten von $\mathcal{A}(L)$
for $i \leftarrow 1$ **to** n **do**
    finde linkesten Schnittpunkt von ℓ_i mit Kante e von B
    $f \leftarrow$ innere Zelle inzident zu e
    while $f \neq$ äußere Zelle **do**
        zerteile f, update $\mathcal{D}$ und setze f auf nächste Zelle



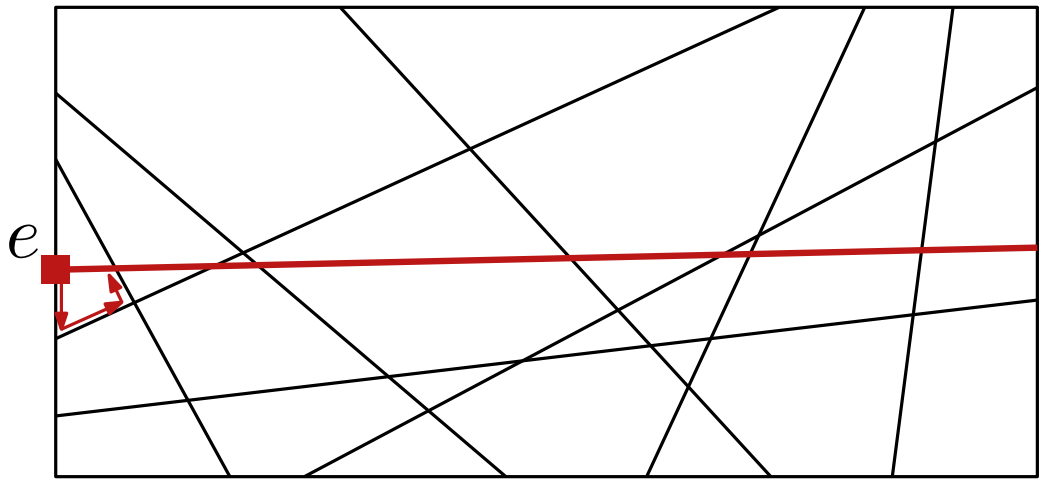
Konstruktion von $\mathcal{A}(L)$

Input: Geraden $L = \{\ell_1, \ldots, \ell_n\}$
Output: DCEL $\mathcal{D}$ für $\mathcal{A}(L)$
initialisiere $\mathcal{D}$ für bounding box B der Knoten von $\mathcal{A}(L)$
for $i \leftarrow 1$ **to** n **do**
    finde linkesten Schnittpunkt von ℓ_i mit Kante e von B
    $f \leftarrow$ innere Zelle inzident zu e
    while $f \neq$ äußere Zelle **do**
        zerteile f, update $\mathcal{D}$ und setze f auf nächste Zelle



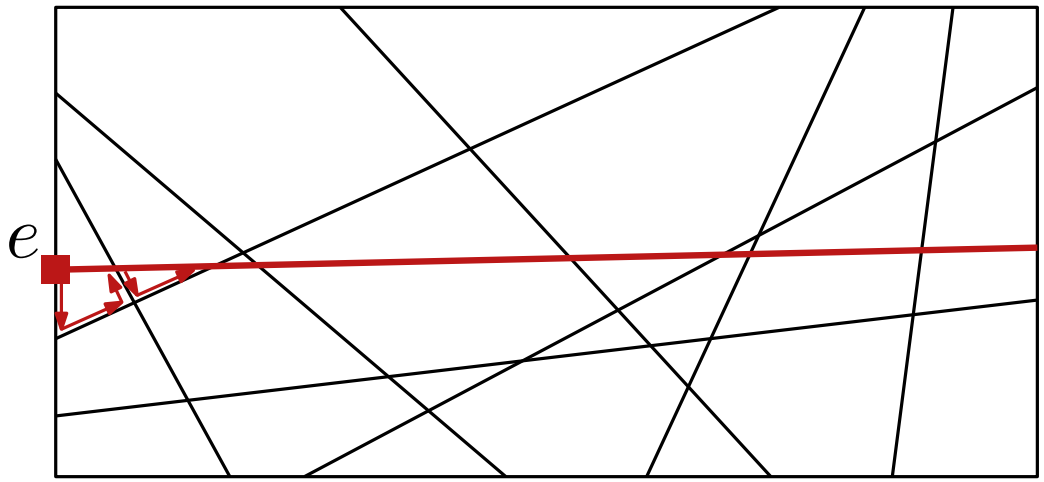
Konstruktion von $\mathcal{A}(L)$

Input: Geraden $L = \{\ell_1, \ldots, \ell_n\}$
Output: DCEL $\mathcal{D}$ für $\mathcal{A}(L)$
initialisiere $\mathcal{D}$ für bounding box B der Knoten von $\mathcal{A}(L)$
for $i \leftarrow 1$ **to** n **do**
    finde linkesten Schnittpunkt von ℓ_i mit Kante e von B
    $f \leftarrow$ innere Zelle inzident zu e
    while $f \neq$ äußere Zelle **do**
        zerteile f, update $\mathcal{D}$ und setze f auf nächste Zelle



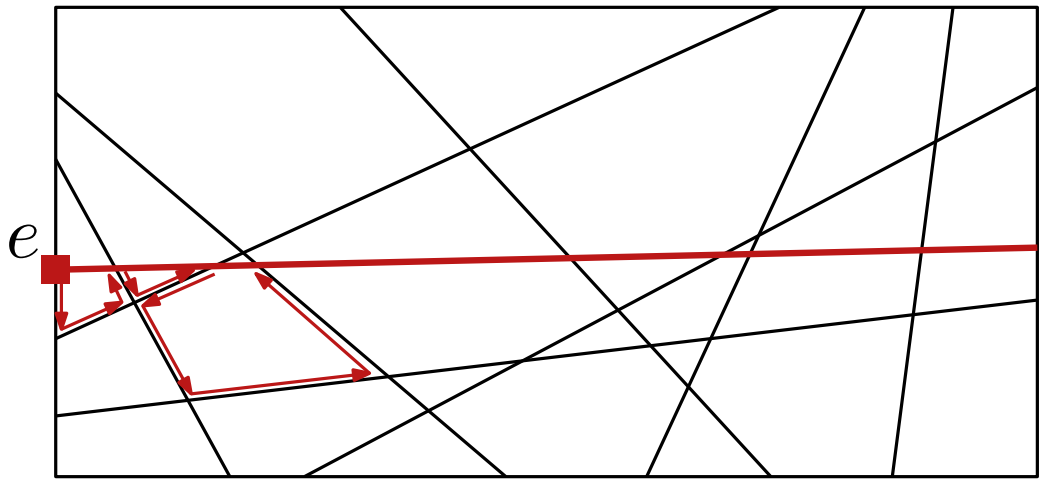
Konstruktion von $\mathcal{A}(L)$

Input: Geraden $L = \{\ell_1, \ldots, \ell_n\}$
Output: DCEL $\mathcal{D}$ für $\mathcal{A}(L)$
initialisiere $\mathcal{D}$ für bounding box B der Knoten von $\mathcal{A}(L)$
for $i \leftarrow 1$ **to** n **do**
    finde linkesten Schnittpunkt von ℓ_i mit Kante e von B
    $f \leftarrow$ innere Zelle inzident zu e
    while $f \neq$ äußere Zelle **do**
        zerteile f, update $\mathcal{D}$ und setze f auf nächste Zelle



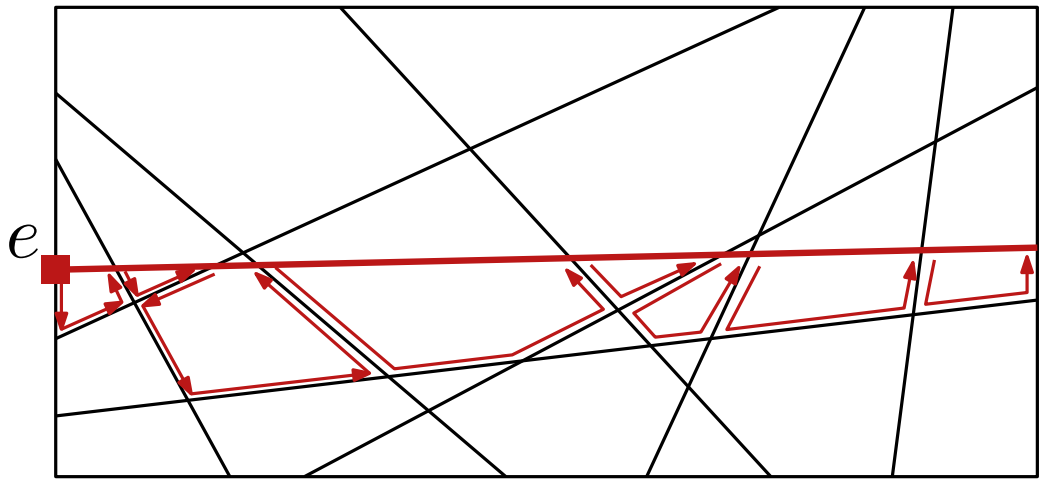
Konstruktion von $\mathcal{A}(L)$

Input: Geraden $L = \{\ell_1, \ldots, \ell_n\}$
Output: DCEL $\mathcal{D}$ für $\mathcal{A}(L)$
initialisiere $\mathcal{D}$ für bounding box B der Knoten von $\mathcal{A}(L)$
for $i \leftarrow 1$ **to** n **do**
    finde linkesten Schnittpunkt von ℓ_i mit Kante e von B
    $f \leftarrow$ innere Zelle inzident zu e
    while $f \neq$ äußere Zelle **do**
        zerteile f, update $\mathcal{D}$ und setze f auf nächste Zelle



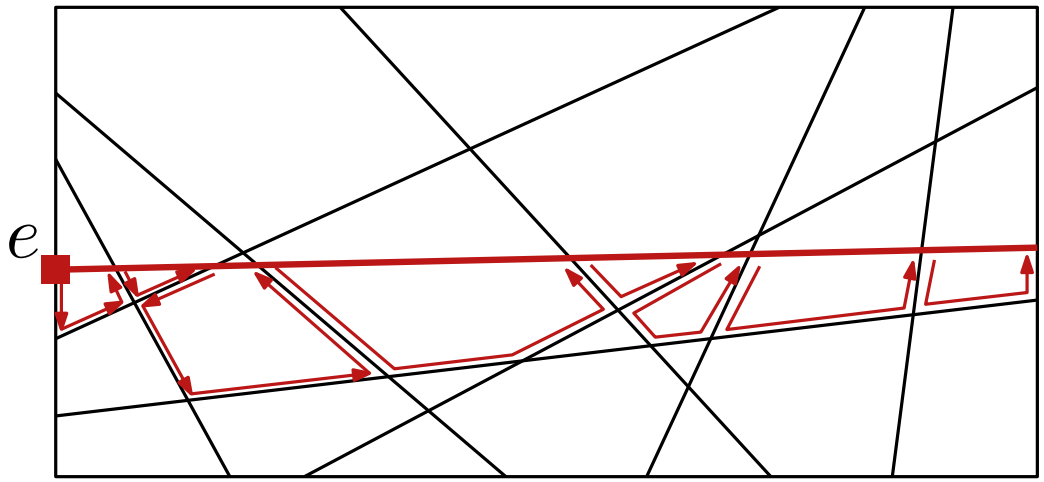
Konstruktion von $\mathcal{A}(L)$

Input: Geraden $L = \{\ell_1, \ldots, \ell_n\}$
Output: DCEL $\mathcal{D}$ für $\mathcal{A}(L)$
initialisiere $\mathcal{D}$ für bounding box B der Knoten von $\mathcal{A}(L)$
for $i \leftarrow 1$ **to** n **do**
    finde linkesten Schnittpunkt von ℓ_i mit Kante e von B
    $f \leftarrow$ innere Zelle inzident zu e
    while $f \neq$ äußere Zelle **do**
        zerteile f, update $\mathcal{D}$ und setze f auf nächste Zelle



Konstruktion von $\mathcal{A}(L)$

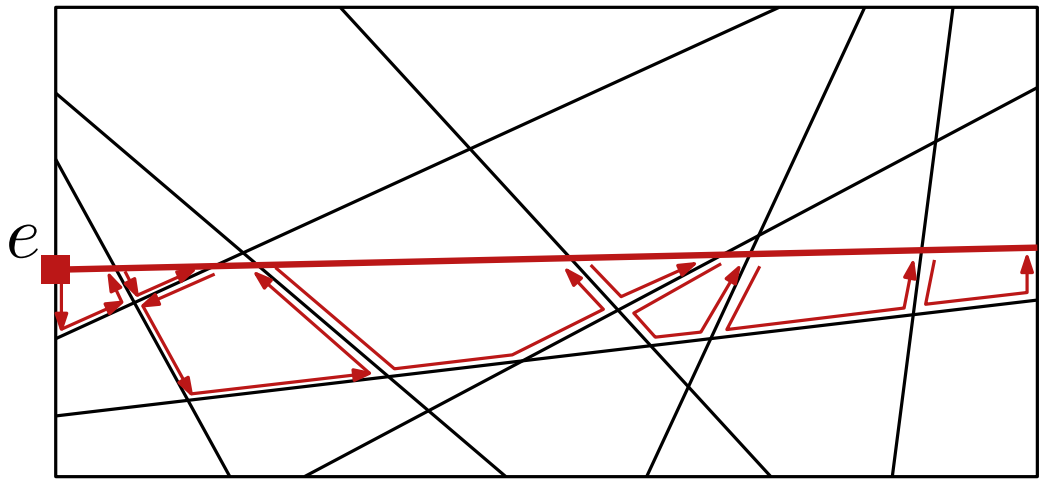
Input: Geraden $L = \{\ell_1, \ldots, \ell_n\}$
Output: DCEL $\mathcal{D}$ für $\mathcal{A}(L)$
initialisiere $\mathcal{D}$ für bounding box B der Knoten von $\mathcal{A}(L)$
for $i \leftarrow 1$ **to** n **do**
    finde linkesten Schnittpunkt von ℓ_i mit Kante e von B
    $f \leftarrow$ innere Zelle inzident zu e
    while $f \neq$ äußere Zelle **do**
        zerteile f, update $\mathcal{D}$ und setze f auf nächste Zelle



Laufzeit?

Konstruktion von $\mathcal{A}(L)$

Input: Geraden $L = \{\ell_1, \ldots, \ell_n\}$
Output: DCEL $\mathcal{D}$ für $\mathcal{A}(L)$
initialisiere $\mathcal{D}$ für bounding box B der Knoten von $\mathcal{A}(L)$
for $i \leftarrow 1$ **to** n **do**
    finde linkesten Schnittpunkt von ℓ_i mit Kante e von B
    $f \leftarrow$ innere Zelle inzident zu e
    while $f \neq$ äußere Zelle **do**
        zerteile f, update $\mathcal{D}$ und setze f auf nächste Zelle

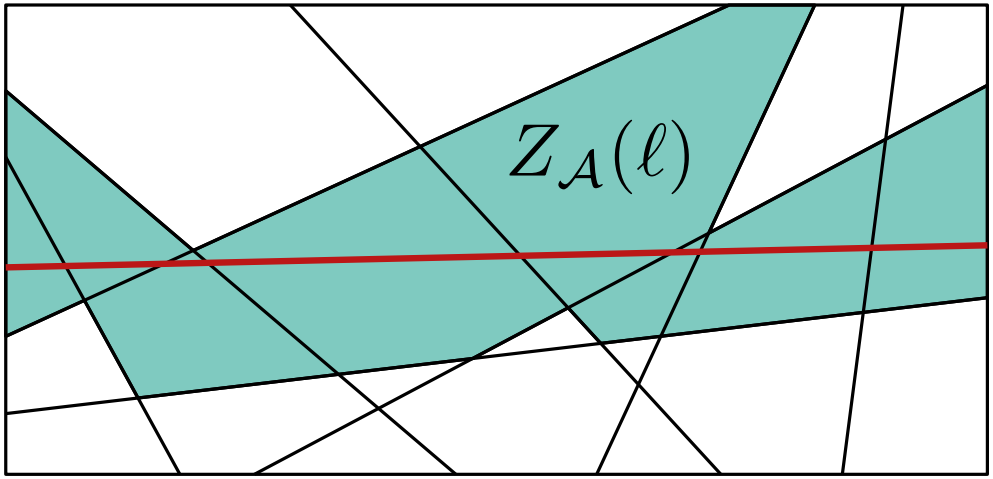


Laufzeit?

- bounding box: $O(n^2)$
- Startpunkt ℓ_i: $O(i)$
- **while**-Schleife: $O(|\text{roter Pfad}|)$

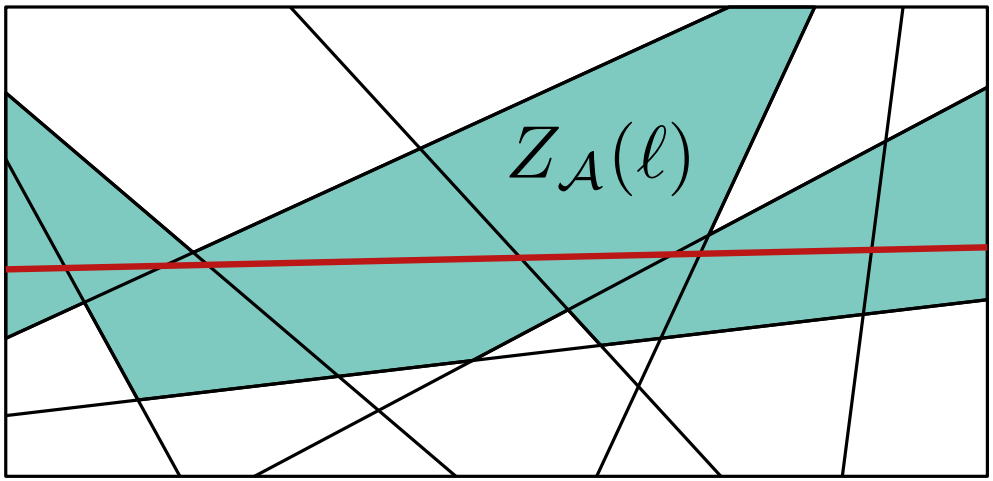
Zonensatz

Def.: Für ein Arrangement $\mathcal{A}(L)$ und eine Gerade $\ell \notin L$ ist die **Zone** $Z_{\mathcal{A}}(\ell)$ die Menge aller Zellen von $\mathcal{A}(L)$, deren Abschluss ℓ schneidet.



Zonensatz

Def.: Für ein Arrangement $\mathcal{A}(L)$ und eine Gerade $\ell \notin L$ ist die **Zone** $Z_{\mathcal{A}}(\ell)$ die Menge aller Zellen von $\mathcal{A}(L)$, deren Abschluss ℓ schneidet.

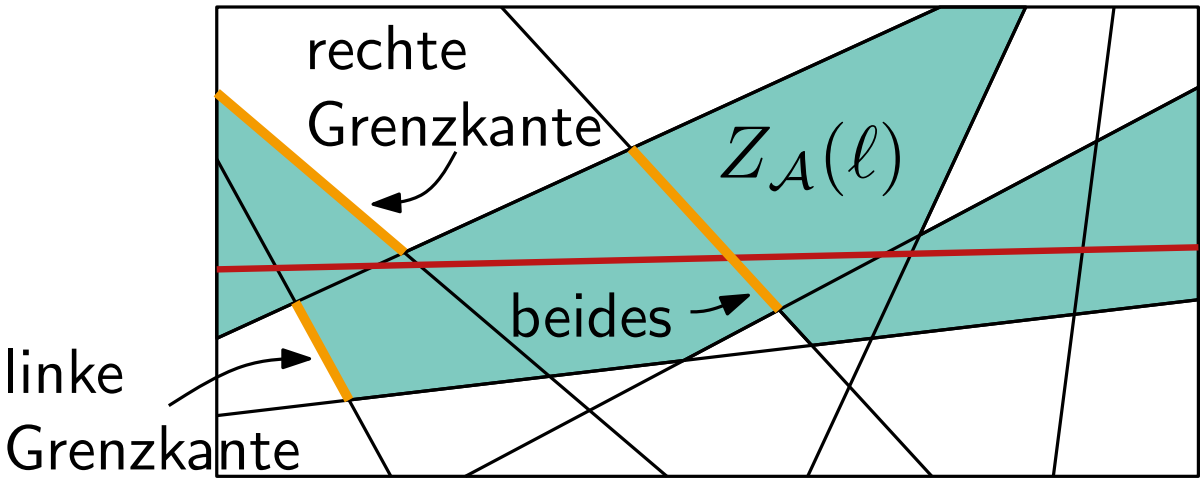


Wie viele Zellen enthält $Z_{\mathcal{A}}(\ell)$?

Wie viele Kanten enthält $Z_{\mathcal{A}}(\ell)$?

Zonensatz

Def.: Für ein Arrangement $\mathcal{A}(L)$ und eine Gerade $\ell \notin L$ ist die **Zone** $Z_{\mathcal{A}}(\ell)$ die Menge aller Zellen von $\mathcal{A}(L)$, deren Abschluss ℓ schneidet.



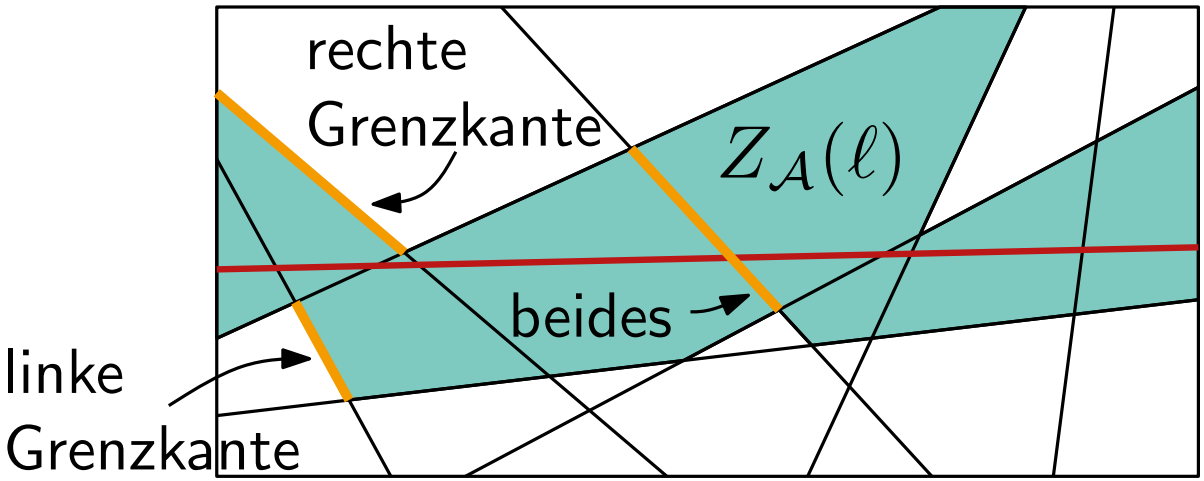
Wie viele Zellen enthält $Z_{\mathcal{A}}(\ell)$?

Wie viele Kanten enthält $Z_{\mathcal{A}}(\ell)$?

Satz 2: Für ein einfaches Arrangement $\mathcal{A}(L)$ von n Geraden in der Ebene und eine Gerade $\ell \notin L$ besteht $Z_{\mathcal{A}}(\ell)$ aus höchstens $6n$ Kanten.

Zonensatz

Def.: Für ein Arrangement $\mathcal{A}(L)$ und eine Gerade $\ell \notin L$ ist die **Zone** $Z_{\mathcal{A}}(\ell)$ die Menge aller Zellen von $\mathcal{A}(L)$, deren Abschluss ℓ schneidet.



Wie viele Zellen enthält $Z_{\mathcal{A}}(\ell)$?

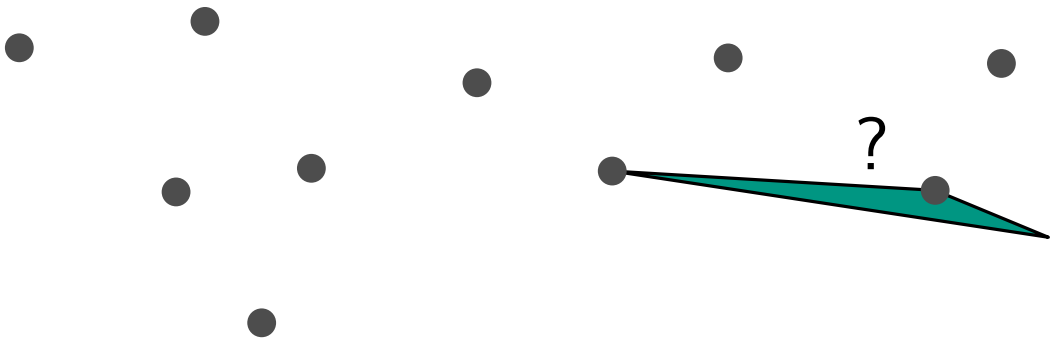
Wie viele Kanten enthält $Z_{\mathcal{A}}(\ell)$?

Satz 2: Für ein einfaches Arrangement $\mathcal{A}(L)$ von n Geraden in der Ebene und eine Gerade $\ell \notin L$ besteht $Z_{\mathcal{A}}(\ell)$ aus höchstens $6n$ Kanten.

Satz 3: Das Arrangement $\mathcal{A}(L)$ einer Menge von n Geraden kann in $O(n^2)$ Zeit und Platz konstruiert werden.

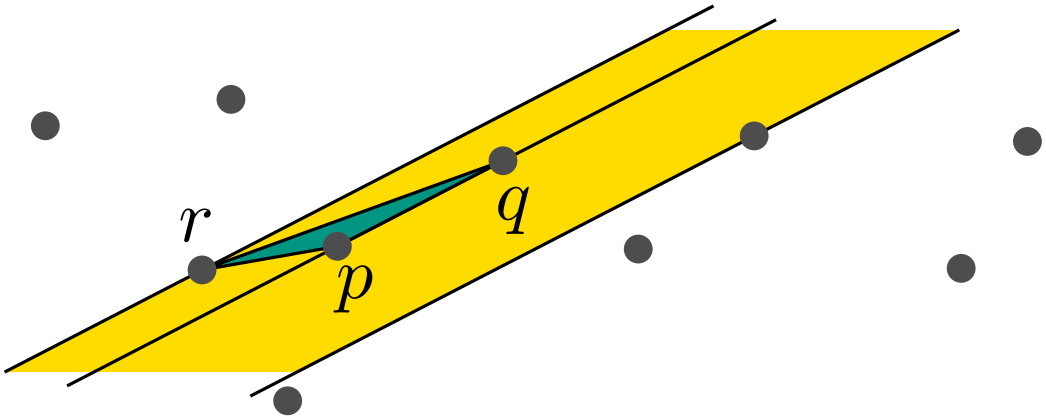
Kleinstes Dreieck

Gegeben eine Menge P von n Punkten in $\mathbb{R}^2$, finde ein flächenminimales Dreieck Δpqr mit $p, q, r \in P$.



Kleinstes Dreieck

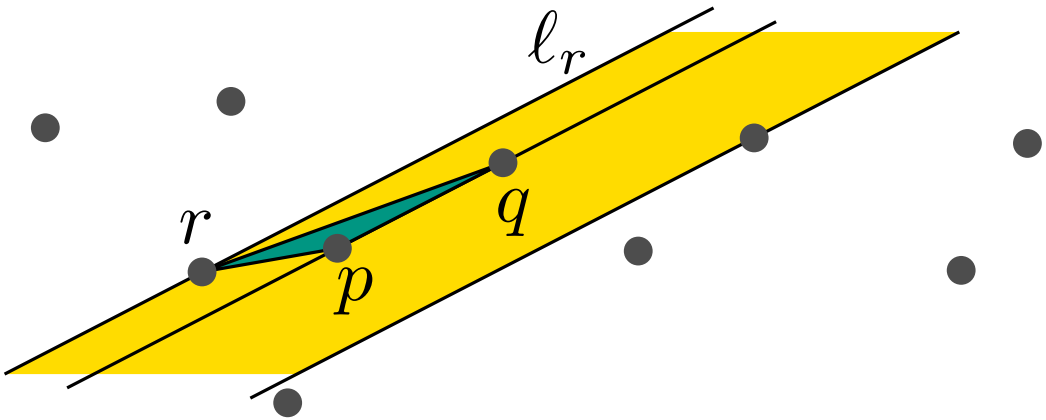
Gegeben eine Menge P von n Punkten in $\mathbb{R}^2$, finde ein flächenminimales Dreieck Δpqr mit $p, q, r \in P$.



Seien $p, q \in P$. Der Punkt $r \in P \setminus \{p, q\}$, der Δpqr minimiert, liegt auf dem Rand des größten leeren Korridors entlang pq.

Kleinstes Dreieck

Gegeben eine Menge P von n Punkten in $\mathbb{R}^2$, finde ein flächenminimales Dreieck Δpqr mit $p, q, r \in P$.

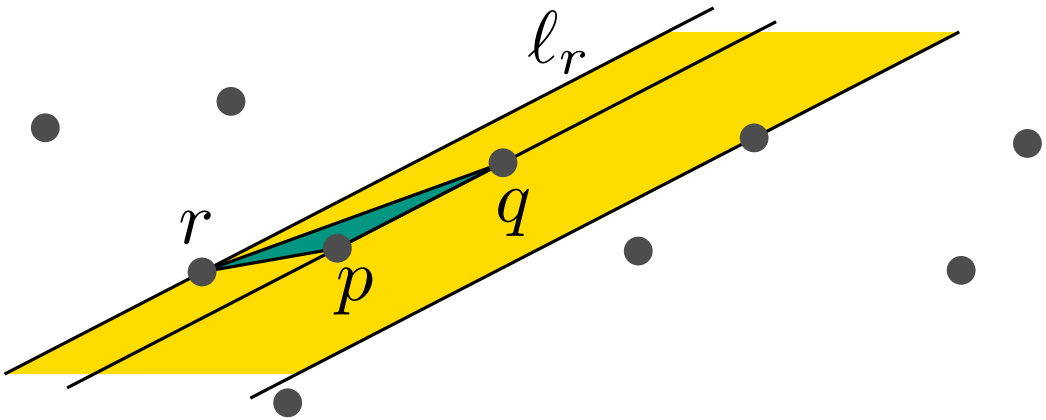


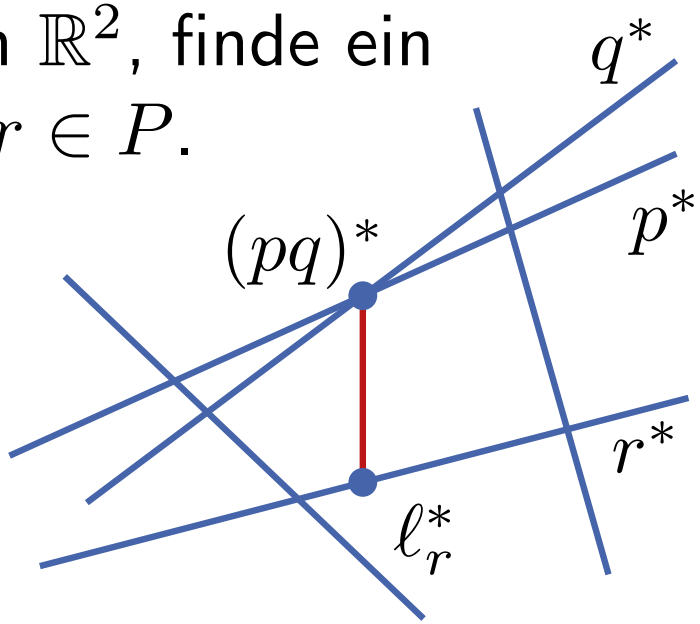
Seien $p, q \in P$. Der Punkt $r \in P \setminus \{p, q\}$, der Δpqr minimiert, liegt auf dem Rand des größten leeren Korridors entlang pq.

Zwischen pq und der Geraden ℓ_r durch r parallel zu pq liegt kein weiterer Punkt aus P.

Kleinstes Dreieck

Gegeben eine Menge P von n Punkten in $\mathbb{R}^2$, finde ein flächenminimales Dreieck Δpqr mit $p, q, r \in P$.





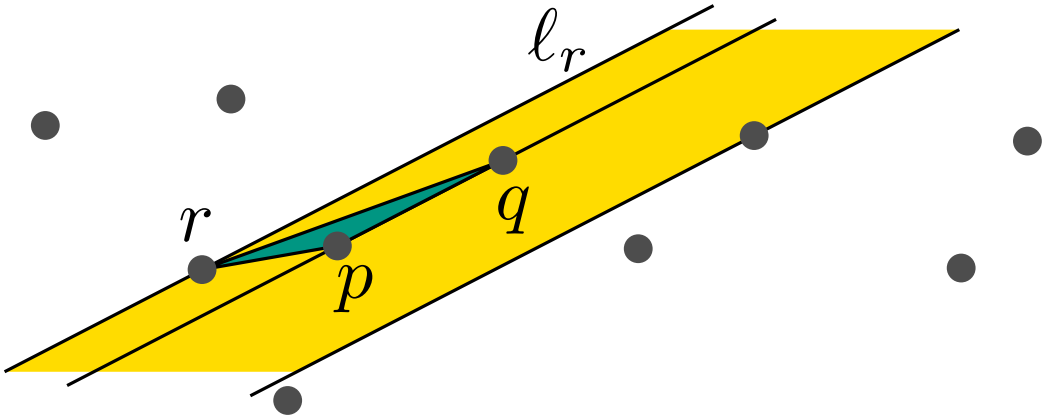
Seien $p, q \in P$. Der Punkt $r \in P \setminus \{p, q\}$, der Δpqr minimiert, liegt auf dem Rand des größten leeren Korridors entlang pq.

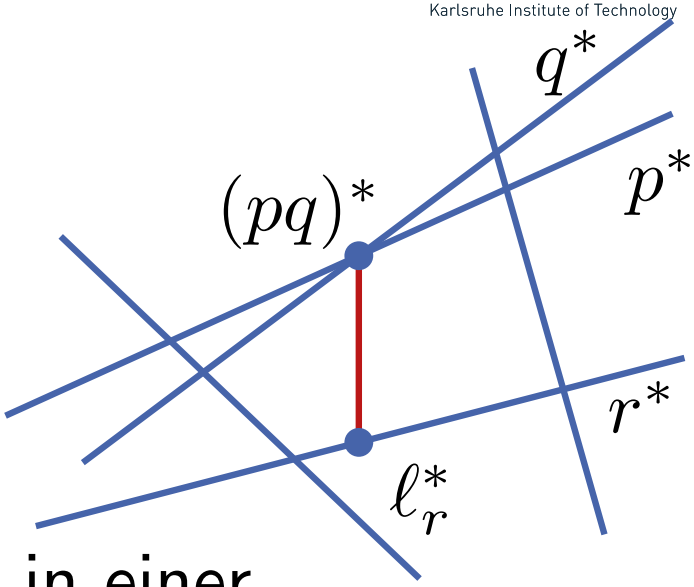
Zwischen pq und der Geraden ℓ_r durch r parallel zu pq liegt kein weiterer Punkt aus P.

Im Dualen:

- ℓ_r^* liegt auf r^*
- ℓ_r^* und $(pq)^*$ haben gleiche x-Koordinate
- keine Gerade $p^* \in P^*$ schneidet $\overline{\ell_r^*(pq)^*}$

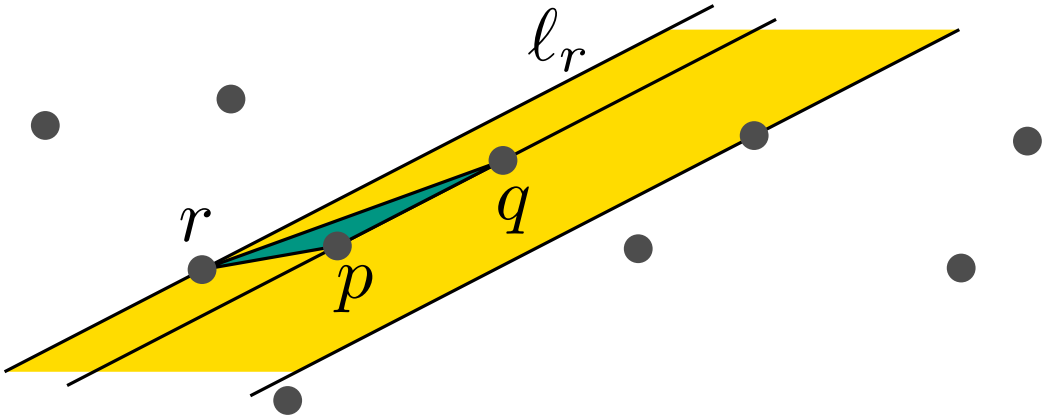
Berechnung im Dualen

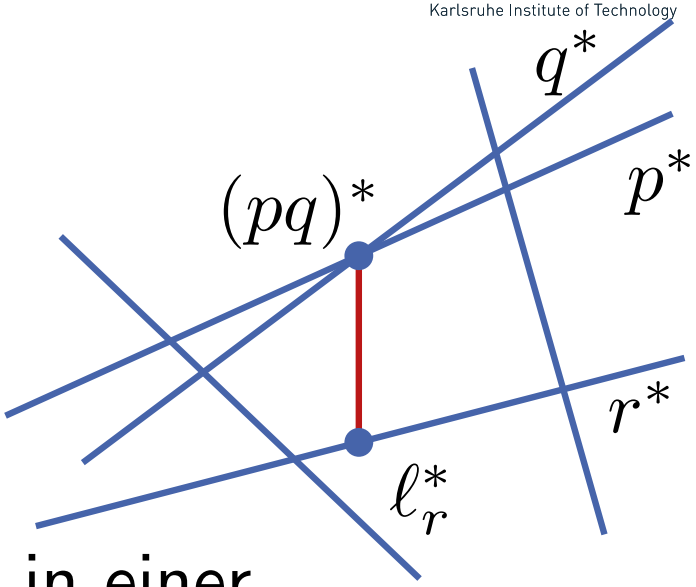




- ℓ_r^* liegt vertikal über oder unter $(pq)^*$ in einer gemeinsamen Zelle von $\mathcal{A}(P^*) \Rightarrow$ nur zwei Kandidaten

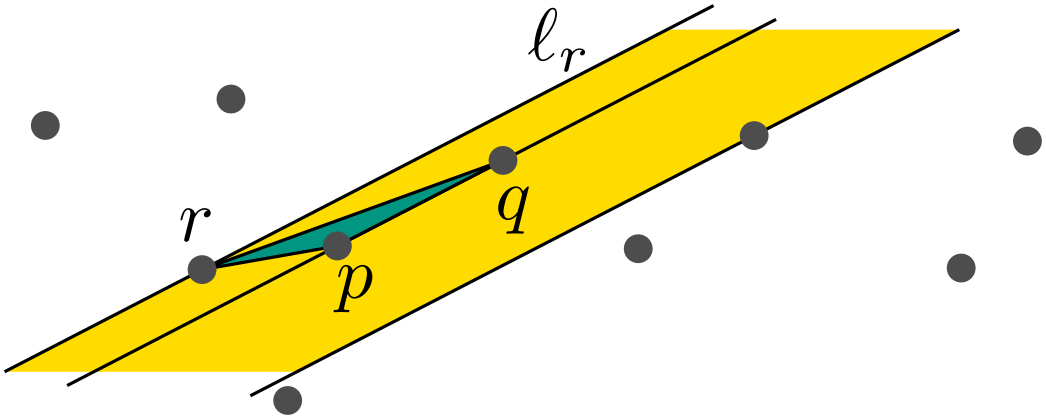
Berechnung im Dualen

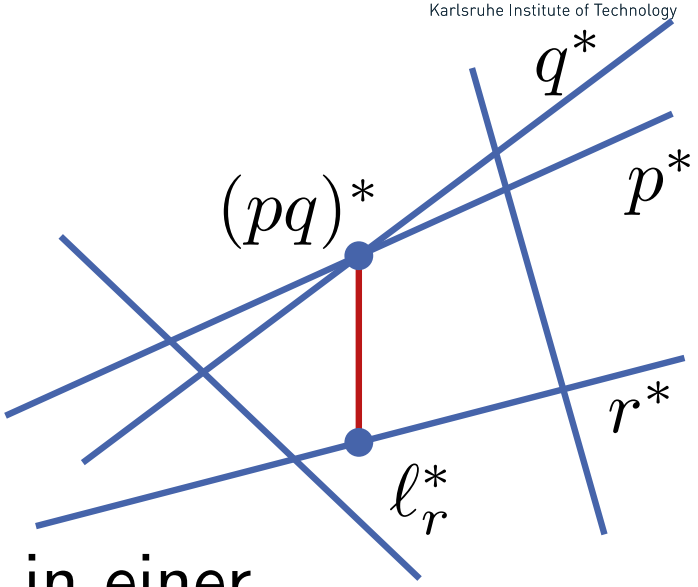




- ℓ_r^* liegt vertikal über oder unter $(pq)^*$ in einer gemeinsamen Zelle von $\mathcal{A}(P^*) \Rightarrow$ nur zwei Kandidaten
- Berechne in $O(n^2)$ Zeit das Arrangement $\mathcal{A}(P^*)$

Berechnung im Dualen

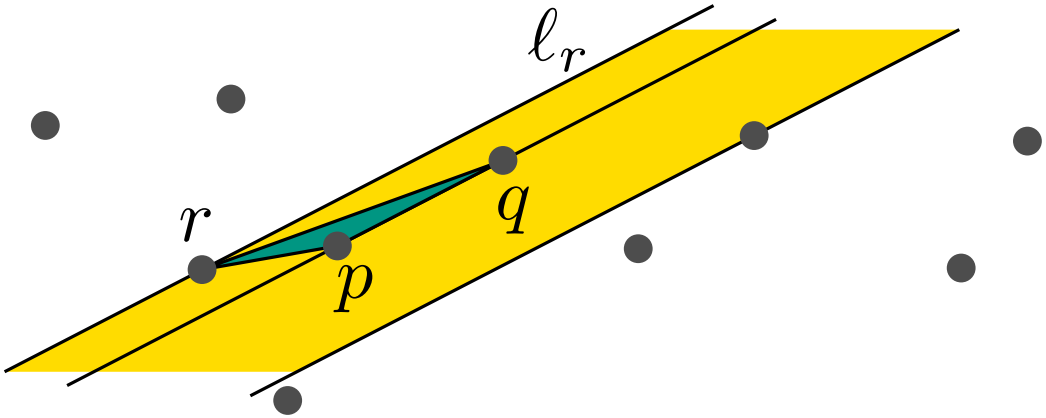


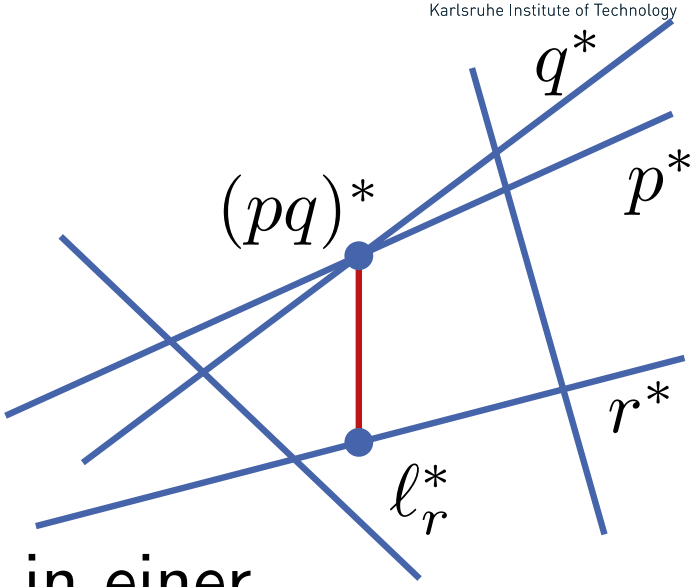


- ℓ_r^* liegt vertikal über oder unter $(pq)^*$ in einer gemeinsamen Zelle von $\mathcal{A}(P^*) \Rightarrow$ nur zwei Kandidaten
- Berechne in $O(n^2)$ Zeit das Arrangement $\mathcal{A}(P^*)$
- Berechne in jeder Zelle die vertikalen Nachbarn der Knoten $\rightarrow$ Zeit linear in Zellenkomplexität

wie?

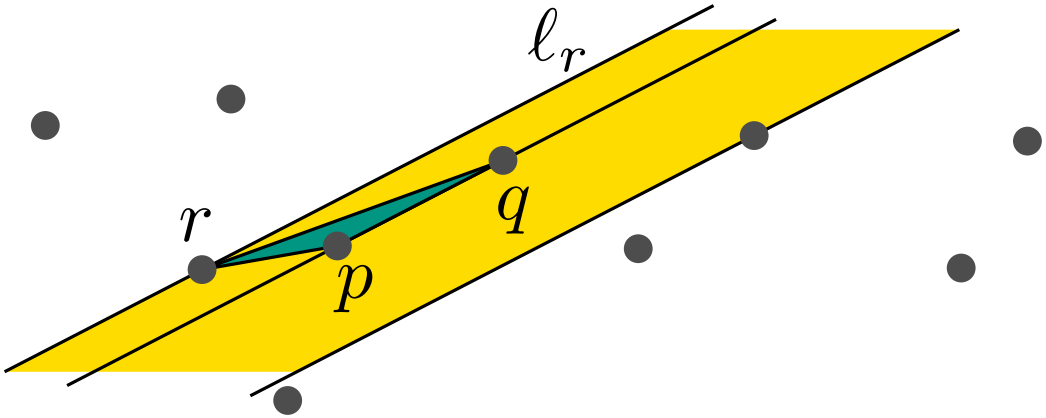
Berechnung im Dualen

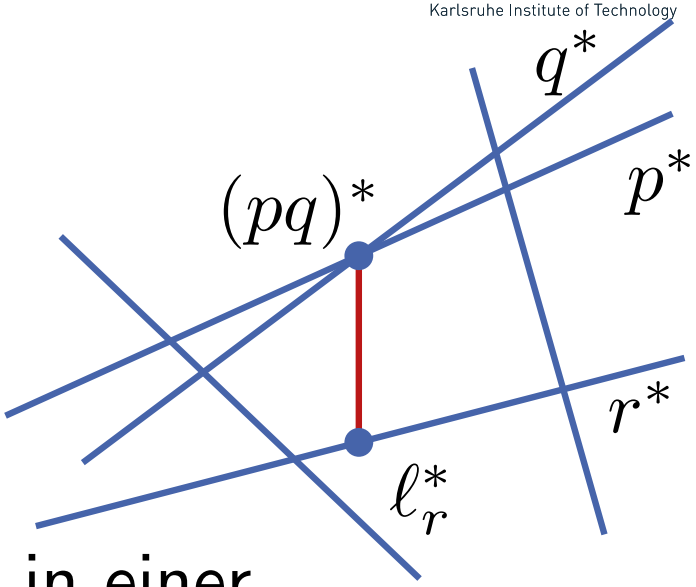




- ℓ_r^* liegt vertikal über oder unter $(pq)^*$ in einer gemeinsamen Zelle von $\mathcal{A}(P^*) \Rightarrow$ nur zwei Kandidaten
- Berechne in $O(n^2)$ Zeit das Arrangement $\mathcal{A}(P^*)$
- Berechne in jeder Zelle die vertikalen Nachbarn der Knoten $\rightarrow$ Zeit linear in Zellenkomplexität wie?
- für alle $O(n^2)$ Kandidaten-Tripel $(pq)^* r^*$ berechne in $O(1)$ Zeit Fläche Δpqr

Berechnung im Dualen





- ℓ_r^* liegt vertikal über oder unter $(pq)^*$ in einer gemeinsamen Zelle von $\mathcal{A}(P^*) \Rightarrow$ nur zwei Kandidaten
- Berechne in $O(n^2)$ Zeit das Arrangement $\mathcal{A}(P^*)$
- Berechne in jeder Zelle die vertikalen Nachbarn der Knoten $\rightarrow$ Zeit linear in Zellenkomplexität wie?
- für alle $O(n^2)$ Kandidaten-Tripel $(pq)^*r^*$ berechne in $O(1)$ Zeit Fläche Δpqr
- findet Minimum in insgesamt $O(n^2)$ Zeit

Diskussion

Dualität ist ein sehr nützliches Werkzeug in der algorithmischen Geometrie!

Diskussion

Dualität ist ein sehr nützliches Werkzeug in der algorithmischen Geometrie!

Kann man die Dualität auch in höheren Dimensionen nutzen?

Diskussion

Dualität ist ein sehr nützliches Werkzeug in der algorithmischen Geometrie!

Kann man die Dualität auch in höheren Dimensionen nutzen?

Ja, auch zwischen d-dimensionalen Punkten und Hyperebenen gibt es eine entsprechende inzidenz- und ordnungserhaltende Dualitätsabbildung.

Diskussion

Dualität ist ein sehr nützliches Werkzeug in der algorithmischen Geometrie!

Kann man die Dualität auch in höheren Dimensionen nutzen?

Ja, auch zwischen d-dimensionalen Punkten und Hyperebenen gibt es eine entsprechende inzidenz- und ordnungserhaltende Dualitätsabbildung.

Wie steht es um höherdimensionale Arrangements?

Diskussion

Dualität ist ein sehr nützliches Werkzeug in der algorithmischen Geometrie!

Kann man die Dualität auch in höheren Dimensionen nutzen?

Ja, auch zwischen d-dimensionalen Punkten und Hyperebenen gibt es eine entsprechende inzidenz- und ordnungserhaltende Dualitätsabbildung.

Wie steht es um höherdimensionale Arrangements?

Das Arrangement von n Hyperebenen im $\mathbb{R}^d$ hat Komplexität $\Theta(n^d)$. Eine Verallgemeinerung des Zonensatzes führt zu einem $O(n^d)$-Zeit Algorithmus.