

Beobachtung:

1. P_j enthält keine Kante von E_r , denn für jede Gerade, die eine E_r -Kante enthält, ex. eine E_l -Kante mit kleinerem Winkel.
2. Wegnahme von P_j kann zwei Kanten von Fläche $f_1, f_2 \in F$ verschmelzen.

Dadurch kann sich $|E_r|$ höchstens um 1 verkleinern.

$$\Rightarrow |E_r| \leq |E_r| + 1 = i$$

Indann.

$$|E_l| \leq i \text{ analog}$$

$$\Rightarrow |E_2| \leq 2i$$

$$\Rightarrow m_i \leq 5i$$

$\Rightarrow$ Folgerung: $O(n^2)$ für Konstruktion von A_n .

$\rightarrow$ 6.4.2.1 Satz (Anordnen):

Das Anordnen von n Geraden kann in Zeit $O(n^2)$ berechnet werden.

Bew.: inkrementeller Mg.

Hinzufügen von P_i kostet $O(i)$ für $i=1 \dots n$.

$\rightarrow$ 6.4.2.2 Folgerungen:

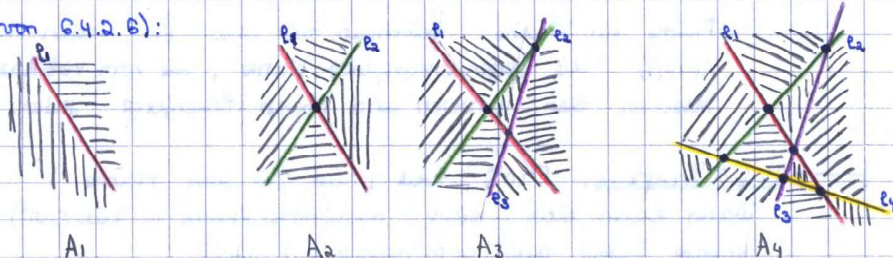
- 1) Test auf drei kollineare Pkte und
- 2) Berechnung des Dreiecks kleinster Fläche in Zeit $O(n^2)$.

→ 6.4.2.8 Laufzeit: Kosten für Einfügen von P_i sind proportional zur Anzahl der im Schritt 2. durchlaufenen Kanten (Größe der Zone von P_i)
Sei m_i diese Anzahl.

→ 6.4.2.9 Lemma: $m_i \leq 5i$ "Zone-Lemma".

→ 6.4.2.10 Beispiel: (Fortsetzung von 6.4.2.6):

- $A_1 = \{P_1\}$
- $A_2 = \{P_1, P_2\}$
- $A_3 = \{P_1, P_2, P_3\}$
- $A_4 = \{P_1, P_2, P_3, P_4\}$



Beweis:

Sei F die Menge der Flächen von A_i oberhalb (positive Seite) von P_i , deren Rand an P_i grenzt. Dann ist m_i die Zahl aller Kanten von Flächen in F .

Im Bsp: A_4 : 4 Flächen, $m_4 = (P_1, P_2, P_4) + (P_4, P_2, P_1, P_3) + (P_4, P_3, P_1) + (P_4, P_1, P_2) = 13$

Partitioniere diese Kantenmenge in zwei Teile:

E_1 : Kanten mit mindestens einem Endpunkt auf P_i in A_4 : $(P_4, P_2), (P_2, P_4, P_3), (P_3, P_4, P_1), (P_1, P_4)$

Beobachtung:

a). jede Fläche in A_{i-1} , die von P_i geschnitten wird, trägt maximal drei Kanten zu E_1 bei.

B. höchstens 3 Flächen werden von P_i geschnitten
 $\Rightarrow |E_1| \leq 3 \cdot i$

E_2 : alle anderen Kanten. in A_4 : $(P_1), (P_2), (P_3, P_2)$

O(i) pro Fläche ist möglich.

Wir zeigen: $|E_2| \leq 2 \cdot i$

Dazu zerlegen wir E_2 weiter in E_e und E_r

E_e : alle Kanten in E_2 so, dass der Schnittpunkt der unterstützenden Geraden mit P_i links von der entsprechenden Fläche liegt. $(P_3, P_2), (P_1)$

E_r : analog mit Schnittpunkt rechts. $(P_1), (P_2)$
(Parallele Segmente (falls vorhanden) zu E_e)

Hilfslemma: a) $|E_e| \leq i$

b) $|E_r| \leq i$.

Beweis: durch Induktion über i

Ind.anf: $i=1 \Rightarrow |F|=1 = m_1 \Rightarrow |E_1|=0 = |E_e| \leq 2 \cdot 1 = 2 \quad \checkmark$

Ind.schritt:

Betrachte A_{i-1} = Arrangement für $\{P_1 \dots P_{i-1}\}$

Wir fügen nun die i -te Gerade P_i hinzu.

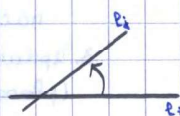
Zurückführung auf Induktionsannahme:

Wähle die Gerade P_j $1 \leq j \leq i-1$, die mit P_i den kleinsten Winkel (oberhalb von P_i und rechts von P_j) bildet

Ab hier Beweis für E_r (E_e analog):

Sei E_r' die E_r Menge beim Einfügen von P_i in Arrangement von $\{P_1 \dots P_{i-1}\} \setminus \{P_j\}$.

Ind. ann. $\Rightarrow |E_r'| \leq i-1$



Frage: Wie verhalten sich $|E_r'|$ und $|E_r|$ zueinander?

Daraus ergibt sich folgende Methode zur Berechnung von $\text{near}(p_i, p_j)$:
 Verschiebe Gerade l_i parallel nach oben und unten bis der nächste Pkt getroffen.
 Im Dualen: Verschiebe Pkt p_i in Arrangements senkrecht nach oben bzw. nach unten bis die nächste Gerade getroffen wird. (d.h. bis zum gegenüberliegenden Rand der Fläche).

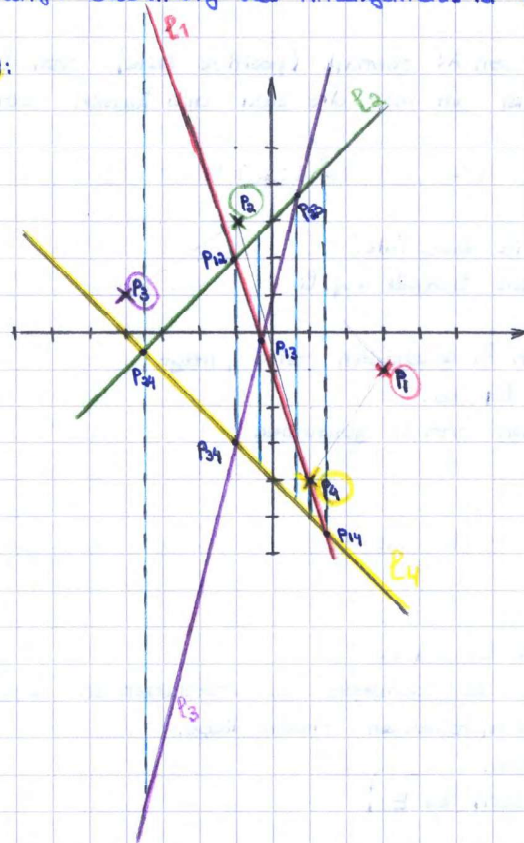
Fläche für Fläche berechnen wir also für jeden Knoten (repräsentiert ein Paar (p_i, p_j)) die gegenüberliegende Kante, die vom vertikalen Strahl getroffen wird.
 Mischen der unteren und oberen Eckenfolge jeder Fläche (von links nach rechts).

→ 6.4.2.5 Laufzeit: $O(n^2)$, weil linear für jede Fläche.

Dann kann das Dreieck minimaler Fläche in Zeit $O(n^2)$ berechnet werden, durch lineare Suche auf $\{(p_i, p_j, \text{near}(p_i, p_j)) : i \neq j\}$

Voraussetzung: Darstellung des Arrangements ist gegeben.

→ 6.4.2.6 Beispiel:



- ♦ $p_1 = (3, -1), D(p_1): y = -3x - 1$
- ♦ $p_2 = (-1, 3), D(p_2): y = x + 3$
- ♦ $p_3 = (-4, 1), D(p_3): y = 4x + 1$
- ♦ $p_4 = (1, -4), D(p_4): y = -x - 4$

p_{12} : gelb, lila

p_{13} : grün

p_{23} : rot

p_{24} : rot

p_{34} : grün, rot

p_{14} : grün

$\text{near}(p_1, p_2) = \{p_3, p_4\}$

$\text{near}(p_2, p_3) = \{p_1\}$

$\text{near}(p_3, p_4) = \{p_1, p_2\}$

$\text{near}(p_4, p_1) = \{p_2\}$

$\text{near}(p_1, p_3) = \{p_2\}$

$\text{near}(p_1, p_4) = \{p_2\}$

Arrangement von n Geraden =
 planare Unterteilung in Flächen,
 Segmente und Knoten durch
 diese Geraden.

→ 6.4.2.7 Berechnung des Arrangements von n Geraden $l_1 \dots l_n$.

1. Möglichkeit: $O(n^2 \log n)$ mit Plane Sweep - Verfahren

2. Anderes inkrementelles Verfahren:

Berechne Folge von Arrangements $A_1 \dots A_n$; A_i enthält die Geraden $\{l_1 \dots l_i\}$

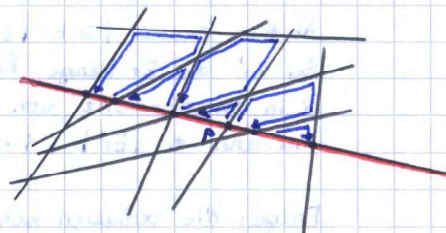
A_1 : trivial

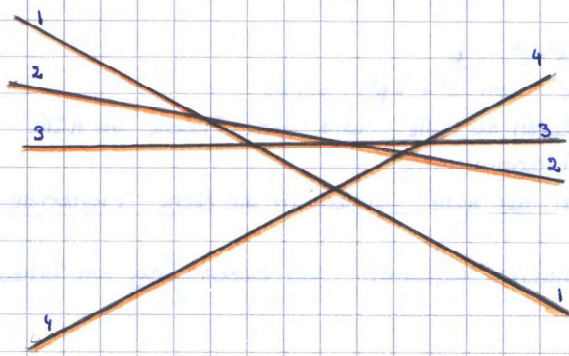
Konstruktion von A_i durch Hinzufügen von l_i zu A_{i-1} .

Idee: 1. Starte in einem beliebigen Schnittpunkt $p = l_i \cap l_j$ für ein $j < i$

2. Durchlaufe von p aus (nach links und rechts) die Flächen von A_{i-1} ab (genauer die Kanten der Fläche) bis Gerade l_i wieder getroffen wird (d.h. wir überqueren l_i mit Kante e , dann ist $e \cap l_i$ der nächste Schnittpunkt.)

3. Update der Datenstruktur (planarer Graph), d.h. einfügen neuer Kanten und Knoten.





→ 6.4.1.6 Laufzeit: $O(n^2 \log n)$

→ 6.4.1.7 Bem. Man kann G auch in Zeit $O(n^4)$ berechnen.

→ Algorithmus später

($\Rightarrow$ dann Laufzeit: $O(n^4)$).

6.4.2. Problem 2

→ 6.4.2.1. Problem: Geg: $p_1, \dots, p_n \in \mathbb{R}^2$

Finde 3 paarweise verschiedene Pkte p_i, p_j, p_k so, dass die Fläche des Dreiecks $\Delta(p_i, p_j, p_k)$ minimal ist.

$\Leftrightarrow$ gilt offensichtlich: 3 Pkte auf gemeinsamer Geraden $\Leftrightarrow$ kleinste Fläche ($=0$).

→ 6.4.2.2. Einfache Log.

Teste alle Dreiecke (Minimumssuche)

$\Rightarrow O(n^3)$ Laufzeit.

→ 6.4.2.3. Bessere Log.

Betrachte alle Paare $(p_i, p_j), i \neq j$, definiere Gerade l_{ij}

Sei $\text{near}(p_i, p_j) = \text{Pkt} \in S \setminus \{p_i, p_j\}$ mit min. Abstand zu l_{ij} .

$\Rightarrow \Delta(p_i, p_j, \text{near}(p_i, p_j))$ hat min. Fläche unter allen Dreiecken $(p_i, p_j, \cdot)$

$\Rightarrow$ lineare Suche auf der Menge $\Delta(p_i, p_j, \text{near}(p_i, p_j))$ liefert das Ergebnis.

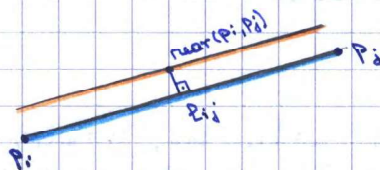
$\Rightarrow$ Laufzeit $O(n^2)$, wenn $\text{near}(p_i, p_j)$ bekannt

Nun: Berechnung von $\text{near}(p_i, p_j) \forall (p_i, p_j)$

a) trivial: $O(n^3)$ (lin. Suche $\forall$ Paare)

b) Dualität.

→ 6.4.2.4 Dualität:



Betrachte Gerade l durch $p_i = \text{near}(p_i, p_j)$ parallel zu l_{ij}

$\Rightarrow$ dann enthält der Streifen zw. l_{ij} und l keinen Pkt, d.h.

Beim Verschieben von l_{ij} nach l wird kein Pkt getroffen.

Im Dualen: Betrachte $l_i = D(p_i)$, $l_j = D(p_j)$ schneiden sich im Pkt

$p_{ij} := D^{-1}(l_{ij})$. Die parallele l geht über in einen Pkt $p = D^{-1}(l)$.

1) p hat gleiche x -Koordinate wie p_{ij} (dual zw parallelen Ges.)

2) p liegt auf $l_k = D(p_k)$

3) Strecke $\overline{p_{ij} p}$ schneidet keine Gerade außer am Rand.

Schritte:

1. $p = (x, y) \mapsto (x, y, x^2 + y^2) = p'$
2. Berechne Konv. Hülle von $S' = \{p' : p \in S\}$.
 $\Rightarrow$ Alle Pkte sind Ecken der Hülle und jede Fläche der Hülle entspricht dem Dreieck der Delaunay-Triangulierung.
3. Projiziere die Kanten der Hülle zurück in die Ebene (ignoriere z-Koordinaten).

Das liefert ein alternatives Verfahren zur Berechnung von Delaunay-Triangulierungen.

6.3.4. Geometrisches Prädikat:

size-of-circle (a, b, c, d)

$a, b, c, d \in \mathbb{R}^2$ (a, b, c nicht kollinear) definieren gerichteten Kreis.

Frage: Wo liegt d ? (Bzgl. der Kreislinie)

$\Rightarrow$ links: +1, auf: 0, recht: -1.

Implementierung:

Größe a, b, c, d auf Paraboloid $\rightarrow a', b', c', d'$

size of circle $(a, b, c, d) = \text{orientation}(a', b', c', d') = \text{sign}$

$$\begin{vmatrix} a_x & a_y & a_x^2 + a_y^2 & 1 \\ b_x & b_y & b_x^2 + b_y^2 & 1 \\ c_x & c_y & c_x^2 + c_y^2 & 1 \\ d_x & d_y & d_x^2 + d_y^2 & 1 \end{vmatrix}$$

6.4. Arrangements und Dualität:

(2 weitere Anwendungen der Dualität).

6.4.1. Problem 1

$\rightarrow$ 6.4.1.1. Problem: Geg: $p_1, \dots, p_n \in \mathbb{R}^2$

Frage: 'Liegen drei Pkte auf einer gemeinsamen Geraden?'

dh. $\exists p_i, p_j, p_k$ mit $i < j < k$ u. p_i, p_j, p_k kollinear.

$\rightarrow$ 6.4.1.2 Einfache Lsg:

Test $\forall$ Tupel $(a, b, c) \in \mathbb{S}^3$ ob $\text{orientation}(a, b, c) = 0$

$\Rightarrow$ Zeit $O(n^3)$

$\rightarrow$ 6.4.1.3 Bessere Lsg:

Betrachte alle Paare $(a, b) \in \mathbb{S}^2$ mit $a \neq b$ und sortiere die entsprechenden Geraden nach ihrer Steigung.

$\Rightarrow n^2$ Geraden.

Test: durchsuche sortierte Liste der Geradengleichungen nach 3-fach Vorkommen.

Laufzeit: 1) Sortieren: $O(n^2 \log n)$ $O(n^2 \log n^2) = O(n^2 \cdot 2 \log n) = O(n^2 \log n)$

2) lineare Suche $O(n^2)$

$\Rightarrow$ Gesamt: $O(n^2 \log n)$

$\rightarrow$ 6.4.1.4 Dualität:

$S = \{p_1, \dots, p_n\}$.

Betrachte die dualen Geraden $L = \{l_1, \dots, l_n\}$, $l_i = D(p_i)$

3 Pkte p_i, p_j, p_k sind kollinear $\Leftrightarrow$ 3 Geraden schneiden sich in einem Pkt.

wg. Folgerung Kap 1.

$\rightarrow$ 6.4.1.5 Algorithmus:

Betr. die n dualen Geraden $l_1, \dots, l_n$ und die Zerlegung der Ebene durch diese Geraden. ($\rightarrow$ Arrangement der Geraden) $O(n^2)$ Zeit um Schnipkte zu bestimmen

$\Rightarrow$ Planarer Graph $G = (V, E)$ mit $|V| \leq n^2$ ($\#$ Schnipkte $\leq n^2$)

$\Rightarrow |E| \leq 3n^2 - 6 = O(n^2)$

3 Pkte in S sind kollinear $\Leftrightarrow \exists$ Knoten $v \in V$ mit $\deg(v) > 4$.

Test in $O(n^2)$: teste alle Knoten von G .

Problem: Aufbau von G .

Lösung: Modifiziertes Plane Sweep (Segmentenschnitt)

- Sweep startet bei $-\infty$ mit $Y =$ Liste der Geraden nach Steigung absteigend sortiert
- Event-Pkte: nur Schnittpkte ($O(n^2)$)
- Sweep endet, wenn Y nach Steigung aufsteigend sortiert

1) + 2) $\Rightarrow$ Insgesamt wird im Durchschritt jede Kante höchstens einmal betrachtet.
 $\Rightarrow$ Laufzeit: $O(n)$

6.2.5. Satz: Die Konv. Hülle von n Pkten im $\mathbb{R}^2$ kann in Zeit $O(n \log n)$ berechnet werden.

Bew: a) sortieren $\rightarrow O(n \log n)$

b) Divide & Conquer $\rightarrow T(n) = 2 \cdot T(n/2) + O(n)$.

6.3 Anwendung von 3d-Konvexe Hülle (Delaunay - Triangulierung).

6.3.1. Einführung:

Ziel: Berechnung des Voronoi-Diagramms (bzw. Delaunay-Triangulierung) im $\mathbb{R}^2$.
 dualer Graph zum Voronoi-Diagramm.

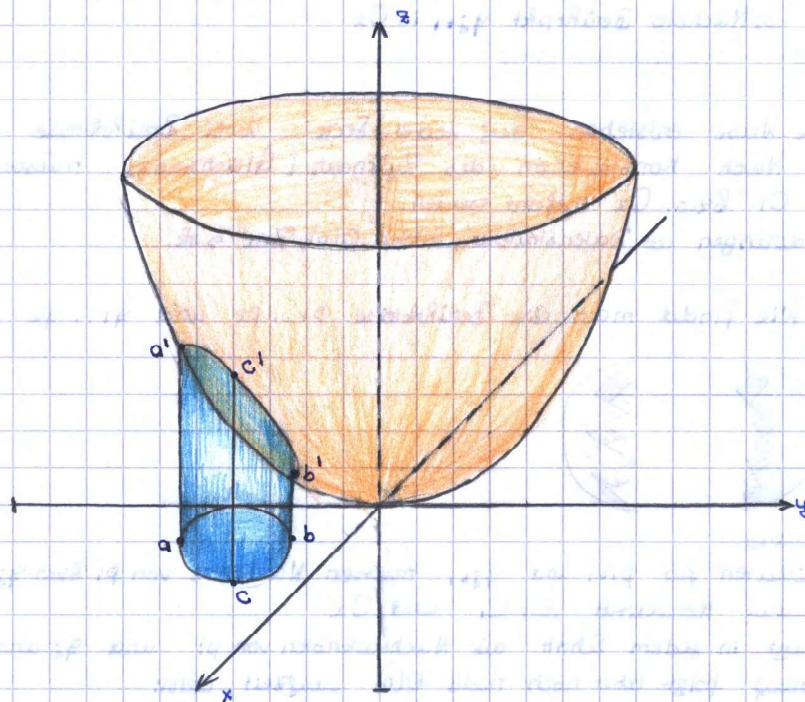
Trianguliere eine Menge $S \subset \mathbb{R}^2$ so, dass $\forall \Delta abc$ gilt:

Der Umkreis von Δabc enthält im Inneren keinen Pkt aus S

Verbindung zw Konvexer Hülle im $\mathbb{R}^3$:

6.3.2. Idee: Übersetze die Tatsache, dass alle Pkte außerhalb oder auf Kreis $O(a,b,c)$ liegen, in die Eigenschaft der konvexen Hülle, dass alle Pkte auf einer Seite der Ebene durch drei Pkte liegen. Diese drei Pkte definieren eine Fläche der Hülle.

6.3.3. Umsetzung: Projektiere jeden Pkt $p \in S \subset \mathbb{R}^2$, $p = (x,y)$ auf einen Paraboloiden, d.h. bilde p ab auf $p' = (x,y,z)$ mit $z = x^2 + y^2$



Betrachte nun Delaunay - Eigenschaft:

Beh. beliebigen Kreis im $\mathbb{R}^2$ durch 3 Pkte a, b, c

Dann definieren die gelisteten Pkte a', b', c' eine Ebene $E(a,b,c)$ im $\mathbb{R}^3$, die den Paraboloiden schneidet. (in einer Ellipse).

Dann gilt:

- 1) Alle Pkte auf dem Rand des Kreises $O(a,b,c)$ werden abgebildet aus diese Ellipse.
- 2) Alle Pkte im Inneren werden auf die positive Seite der Ebene $E(a,b,c)$ abgebildet.
- 3) Alle Pkte außen — " — negative — " —