

## Kapitel III: Drei-dimensionale Konvexe Hüllen.

### 6.1. Einführung:

#### 6.1.1. Problem:

Geg: Menge  $S$  von  $n$  Pkten im  $\mathbb{R}^3$ ,  $p \in S, p = (x, y, z)$  (kart. Koordin.)

Ges: CH(S) (= kleinste konvexe Menge, die  $S$  enthält).

Analog zum Gummiband-Modell: Gummiband-Fläche

$\Rightarrow$  CH(S) ist ein konvexes Polyeder.

Ausgabe: Oberfläche des Polyeders.

- besteht aus Ecken, Kanten und Flächen (Euler-Formel)
- kann beschrieben werden durch einen planaren Graphen.

#### 6.1.2. Darstellung des (planaren) Oberflächengraphen:

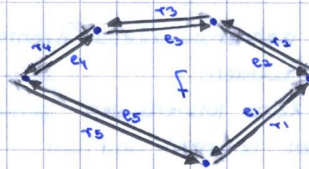
als zweifach gerichteten (bidirected) Graphen.

1. Für jeden Knoten (Ecke)  $v$  ordnen wir die ausgehenden Kanten gg. den Uhrzeigersinn (bei Ansicht von außen)

2. Jede Kante  $e = (u, v)$  hat einen Verweis auf ihre Gegenkante  $\bar{e} = (v, u) =: \text{rev}(e)$

3. Die Flächen sind dann implizit definiert  
 $e_2 = \text{Vorgänger}(\text{rev}(e_1))$  (in der Adjazenzliste)

Das definiert einen Kantenzyklus für jede Fläche  $f$ .

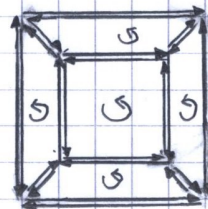
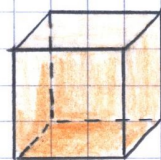


$$e_2 = G.\text{face\_cycle\_successor}(e_1) \\ G.\text{cyclic\_pred}(G.\text{reversal}(e_1))$$

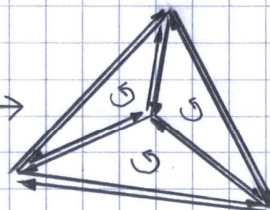
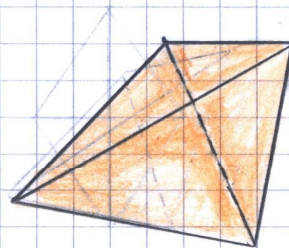
4. In den CH-Algorithmen werden zunächst alle Flächen Dreiecke sein.

Bei allg. eingebetteten planaren Graphen ist u.U. die äußere Fläche kein Dreieck.

#### 6.1.3. Beispiel:



← nach außen gegen den Uhrzeigersinn besser !!



#### 6.1.4. Geometrische Prädikate:

Für  $a, b, c, d \in \mathbb{R}^3$ :

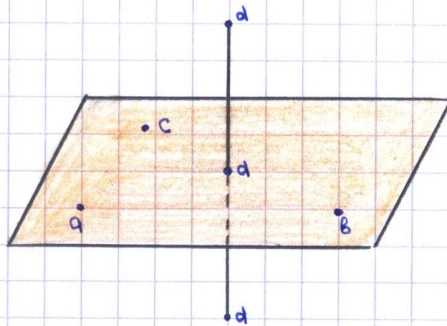
$$\text{orientation}(a, b, c, d) := \text{sign det} \begin{pmatrix} a_x & a_y & a_z & 1 \\ b_x & b_y & b_z & 1 \\ c_x & c_y & c_z & 1 \\ d_x & d_y & d_z & 1 \end{pmatrix}$$

Diese Determinante heißt Spatprodukt.

= Vielfaches des Volumens (mit Vorzeichen) des Simplex  $(a, b, c, d)$

d.h.  $\text{orientation}(a, b, c, d)$  sagt, auf welcher Seite der Ebene durch  $a, b, c$  (a, b, c nicht kollinear) der Pkt  $d$  liegt.





> 0

= 0 (linabh., d.h. in einer gemeinsamen Ebene).

< 0

6.2 Algorithmen:

6.2.1. Inkrementeller Alg. (Variante von Graham's Scan)

→ 6.2.1.1. Alg. (analog zu  $\mathbb{R}^2$ -Alg.)

1. Sortiere Pkte nach x-Koord. (bei Gleichheit nach y, z, d.h. lexikografisch).

2. Baue Simplex aus ersten vier Pkten.

(Sonderfall: coplanar) dann suche solange bis linunabh.

3. Allg. Situation:

Füge  $p_i$  hinzu, Hülle  $CH(p_{i-1}, \{p_1, \dots, p_{i-1}\})$  liegt als Oberflächengraph vor.

Startend im Knoten  $p_{i-1}$  durchsuchen wir den Graphen und besuchen alle Dreiecke (Flächen)  $(a, b, c)$  mit Orientierung  $(a, b, c, p_i) \geq 0$ .

(Alle von der Sichtquelle  $p_i$  beleuchteten Dreiecke oder alle von  $p_i$  sichtbaren Dreiecke).

Dann entfernen wir alle Kanten von Besuchten, sichtbaren (beleuchteten) Dreiecken und alle danach isolierten Knoten.

→ Am Rand entstehen Kanten ohne Gegenkanten.

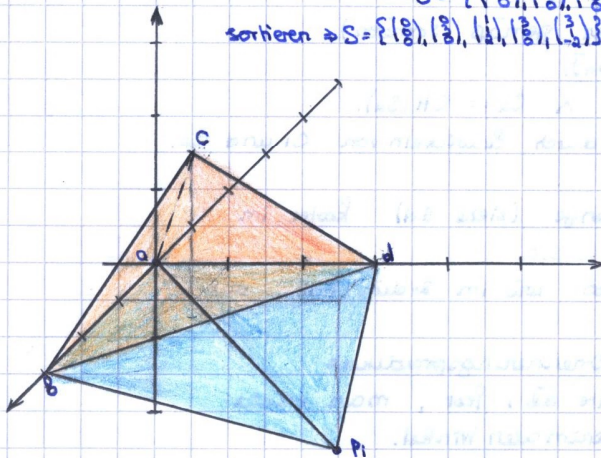
Für jede Kante  $(v, w)$  auf dem Rand füge ein neues Dreieck  $(v, w, p_i)$  hinzu.

→ Tangentialer Kegel mit Spitze  $p_i$ .

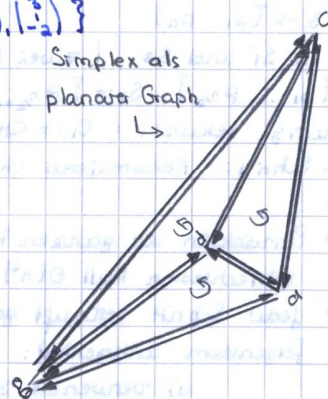
→ 6.2.1.2 Bsp:

$$S = \left\{ \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \right\}$$

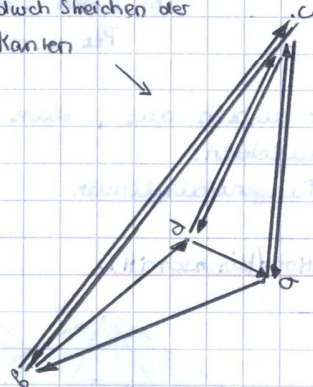
$$\text{sortieren} \Rightarrow S = \left\{ \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \right\}$$



Simplex als planarer Graph



durch Streichen der Kanten



$$\text{orient}(a, c, b, p_i) \leq 0$$

$$\text{orient}(d, b, a, p_i) \geq 0$$

$$\text{orient}(d, c, a, p_i) \leq 0$$

$$\text{orient}(c, b, a, p_i) \leq 0$$

Es entstehen neue Dreiecke:

$$\Delta d b p_i, \Delta d a p_i, \Delta a b p_i$$

$$\Delta b d a \text{ verschwindet.}$$

### → 6.2.1.3 Bem:

Am Ende bleibt also:



### → 6.2.1.4 Laufzeit:

im  $\mathbb{R}^2$ :



werden nur einmal entfernt + Kosten für zwei Tangenten.

im  $\mathbb{R}^3$ :

Rand ist nicht konstant ( $\Rightarrow$  mehr als zwei Tangenten)

$O(n)$  möglich!

Im schlechtesten Fall: Laufzeit:  $O(n^2)$ .

In der Praxis wird der Alg. dennoch eingesetzt, da es für viele Eingaben doch relativ schnell ist.

### → 6.2.1.5 Bem:

⊕ einfach, praktisch, effizient (für bel. Eingaben)

⊖ im schlechtesten Fall:  $O(n^2)$ .

→ Randomisierung: Sweep in zufällige Richtung

⊕ Drehung der Pktmenge in zufälligen Winkel.

Resultat: Erwartete Laufzeit:  $O(n \log n)$ .

## 6.2.2. Divide & Conquer - Alg:

### → 6.2.2.1. Algorithmus:

1. Sortiere  $S \rightarrow \{p_1, \dots, p_n\}$

2. Zerlege in  $S_1$  und  $S_2$  (zwei Hälften gemäß sortieren)

$S_1 := \{p_1, \dots, p_{n/2}\}$ ,  $S_2 := \{p_{n/2+1}, \dots, p_n\}$ .

3. Berechnung rekursiv:  $C_1 := CH(S_1) \wedge C_2 := CH(S_2)$ .

4. Misch-Schritt: Konstruieren  $CH(S)$  durch Einwickeln von  $C_1$  und  $C_2$ .

Bem:

• Einwickeln der ganzen Pktmenge (siehe 2d) kostet im schlechtesten Fall  $O(n^2)$ . log. S. 0

• Jeder Schritt verläuft genauso wie im 2-dim. Fall mit folgendem Unterschied:

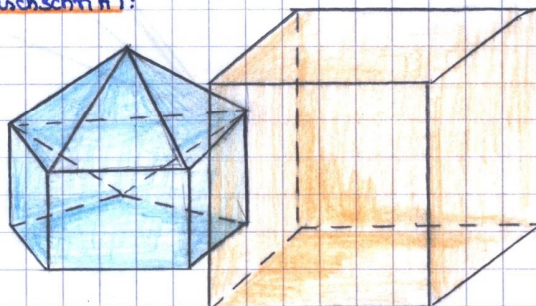
a) verwendet 3d-Ordnungsprodukt

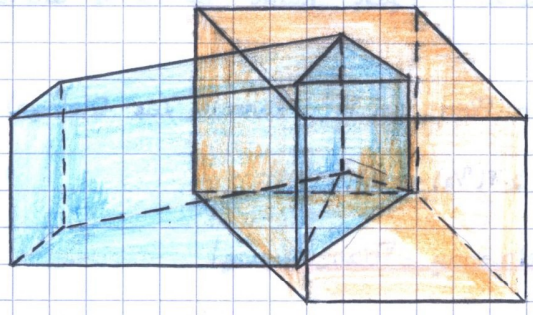
b) 2 Pkte (d.h. Kante  $\overline{ab}$ ) fest, man sucht Pkt  $c$  mit minimalen Winkel.

Wir nutzen aus, dass wir zwei konvexe Polyeder  $C_1$  und  $C_2$  einwickeln.

→ Tangentialzylinder.

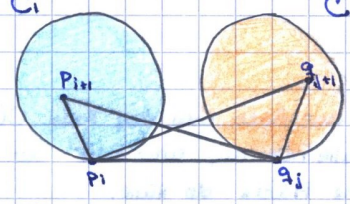
### → 6.2.2.2. Situation (Mischschritt):





Sei  $p_i \in C_1$  und  $q_j \in C_2$  so, dass die Kante  $\overline{p_i q_j}$  der aktuellen Rand des Einwickelzylinders darstellt.

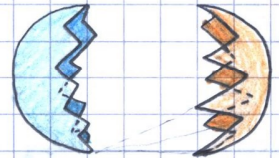
Initialisierung:  $\overline{p_i q_j}$  durch Berechnung der Tangente in den Projektionen  $C_1$   $C_2$



- Zwei Mögl: a) Nächste Berührungspunkt der Einwickelenebene  $p_{i+1}$  liegt in  $C_1$ .  
 $\Rightarrow$  füge  $\Delta p_i q_j p_{i+1}$  hinzu. (zum Zylinder) und die Drehachse der Ebene geht nun durch  $\overline{p_{i+1} q_j}$
- b) Nächster Berührungspunkt  $q_{j+1}$  in  $C_2$   
 (analog wie oben)

Auf diese Weise entstehen auf jeder Seite zwei Berührungsebene  $p_1 \dots p_e$  und  $q_1 \dots q_k$ . Nach Konstruktion des Zylinders (Gleichzeitig) müssen die abgedeckten Teile von  $C_1$  bzw.  $C_2$  entfernt werden.  
 Diese Änderungen der Datenstruktur kosten  $O(n)$  Zeit  $\Rightarrow ok$ .

6.2.2.3 Problem: Wie findet man die Berührungsebene  $p_1 \dots p_e$  und  $q_1 \dots q_k$  in Zeit  $O(n)$ ?

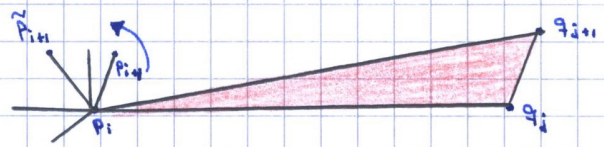


6.2.2.4 Beobachtung:

- 1) Die Kandidaten für  $p_{i+1}$  und  $q_{j+1}$  müssen Nachbarn von  $p_i$  bzw.  $q_j$  sein.  
 Dies folgt aus Konvexität von  $C_1$  und  $C_2$ .  
 $\Rightarrow$  Es genügt in jedem Schritt die Nachbarknoten von  $p_i$  und  $q_j$  anzuschauen  
 Bea: Darauf folgt aber noch nicht  $O(n)$  Laufzeit, denn:
  - a) der Grad eines Knotens kann  $O(n)$  sein.
  - b)  $p_i$  oder  $q_j$  kann für eine große Zahl von Schritten beibehalten werden.

Dann müsste man in jedem dieser Schritte immer wieder alle  $O(n)$  Nachbarn untersuchen.

- 2) Es liegt jedoch eine Monotonie-Eigenschaft vor:  
 Wenn z.B.  $p_i$  in einem Schritt festgehalten wird (d.h. Minimum der beiden Kandidaten  $p_{i+1}$  und  $q_{j+1}$  was  $q_{j+1}$ ), dann folgt der neue Kandidat  $\tilde{p}_{i+1}$  (bzgl. der neuen Drehachse) nach dem Knoten  $p_{i+1}$  in der Adjazenzliste von  $p_i$  oder ist gleich  $p_{i+1}$ . (ohne Beweis).



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

$\rightarrow$  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.

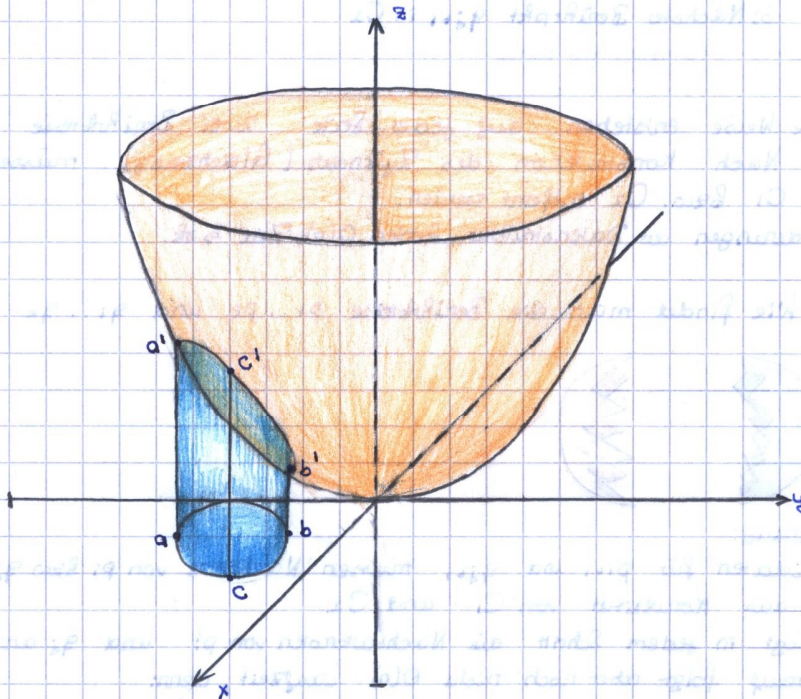
Trianguliere eine Menge  $S \subset \mathbb{R}^2$  so, dass  $\forall \Delta abc$  gilt:

Der Umkreis von  $\Delta abc$  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 gehörenden Pkte  $a', b', c'$  eine Ebene  $E(abc)$  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  $E(abc)$  abgebildet.
- 3) Alle Pkte außen " " negative " "