Vorlesung Algorithmische Geometrie

Konvexe Hülle in $\mathbb{R}^3$

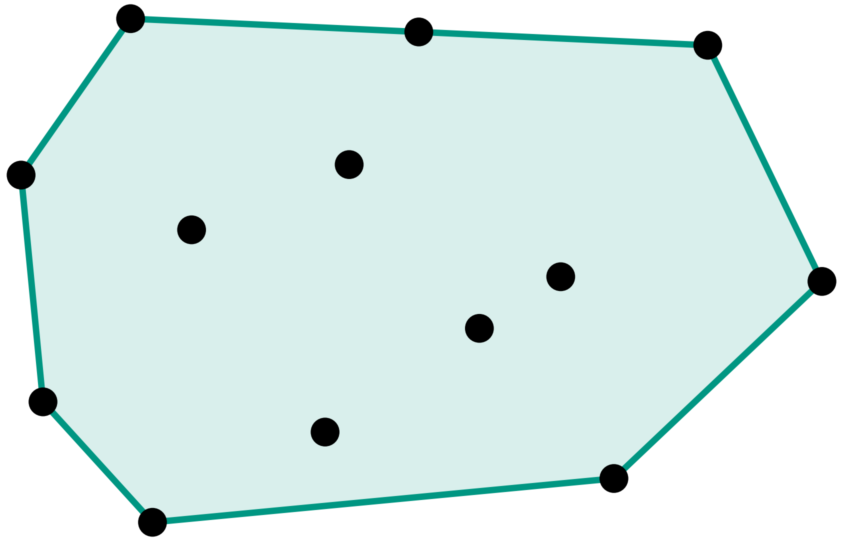
INSTITUT FÜR THEORETISCHE INFORMATIK · FAKULTÄT FÜR INFORMATIK

Martin Nöllenburg

15.07.2014

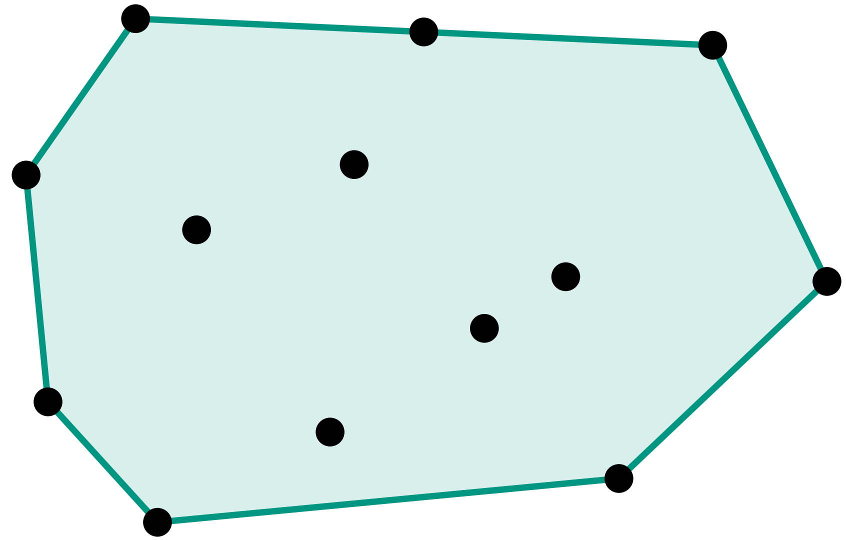
Wdh: Konvexe Hülle in $\mathbb{R}^2$ (VL1)

Def: Eine Menge $S \subseteq \mathbb{R}^2$ heißt **konvex**, wenn für je zwei Punkte $p, q \in S$ auch gilt $\overline{pq} \in S$.
Die **konvexe Hülle** $CH(P)$ einer Punktmenge P ist die kleinste konvexe Menge, die P enthält.



Wdh: Konvexe Hülle in $\mathbb{R}^2$ (VL1)

Def: Eine Menge $S \subseteq \mathbb{R}^2$ heißt **konvex**, wenn für je zwei Punkte $p, q \in S$ auch gilt $\overline{pq} \in S$.
Die **konvexe Hülle** $CH(P)$ einer Punktmenge P ist die kleinste konvexe Menge, die P enthält.



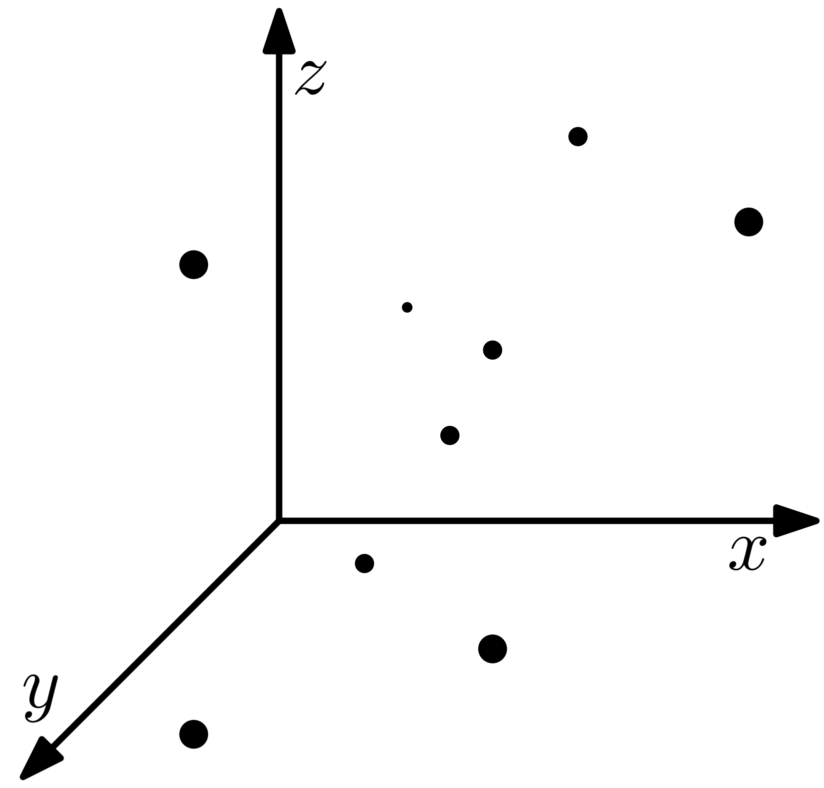
Wir haben gesehen:

- $CH(P)$ ist konvexes Polygon mit Ecken in P und $O(n)$ Kanten
- drei Algorithmen zur Berechnung von $CH(P)$ in Zeit $O(n \log n)$, $O(nh)$ und $O(n \log h)$, wobei $h =$ Anzahl Ecken von $CH(P)$

Konvexe Hülle in $\mathbb{R}^3$

Def: Eine Menge $S \subseteq \mathbb{R}^3$ heißt **konvex**, wenn für je zwei Punkte $p, q \in S$ auch gilt $\overline{pq} \in S$.
Die **konvexe Hülle** $CH(P)$ einer Punktmenge P ist die kleinste konvexe Menge, die P enthält.

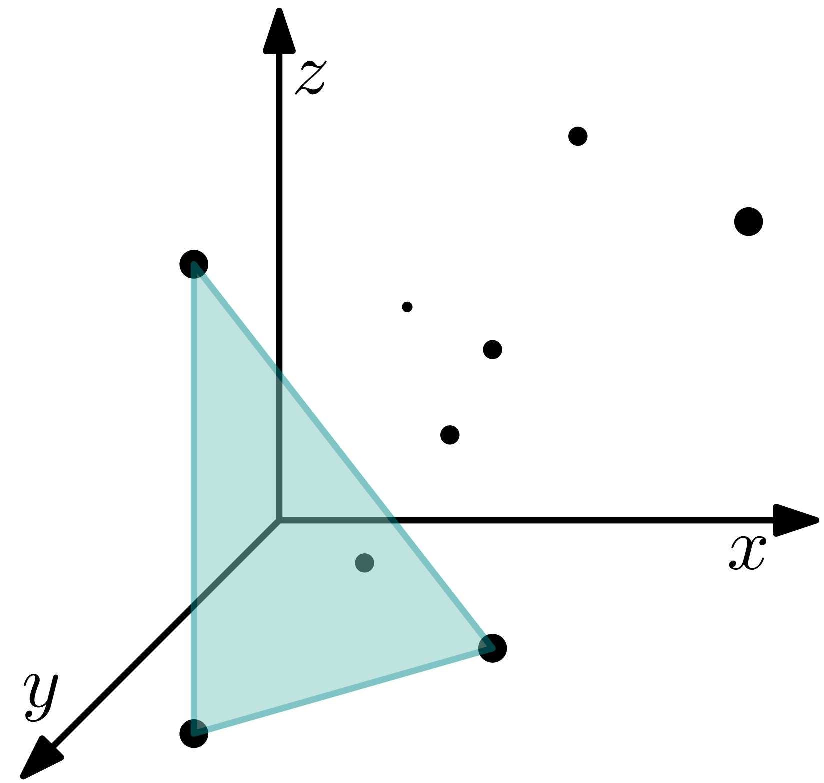
Was ist $CH(P)$ in $\mathbb{R}^3$?



Konvexe Hülle in $\mathbb{R}^3$

Def: Eine Menge $S \subseteq \mathbb{R}^3$ heißt **konvex**, wenn für je zwei Punkte $p, q \in S$ auch gilt $\overline{pq} \in S$.
Die **konvexe Hülle** $CH(P)$ einer Punktmenge P ist die kleinste konvexe Menge, die P enthält.

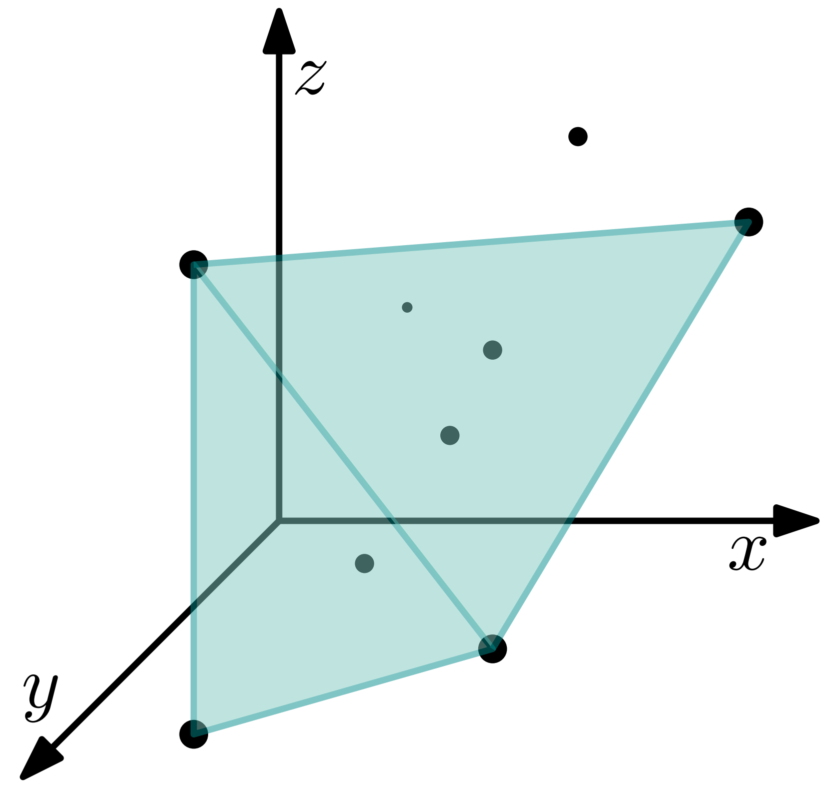
Was ist $CH(P)$ in $\mathbb{R}^3$?



Konvexe Hülle in $\mathbb{R}^3$

Def: Eine Menge $S \subseteq \mathbb{R}^3$ heißt **konvex**, wenn für je zwei Punkte $p, q \in S$ auch gilt $\overline{pq} \in S$.
Die **konvexe Hülle** $CH(P)$ einer Punktmenge P ist die kleinste konvexe Menge, die P enthält.

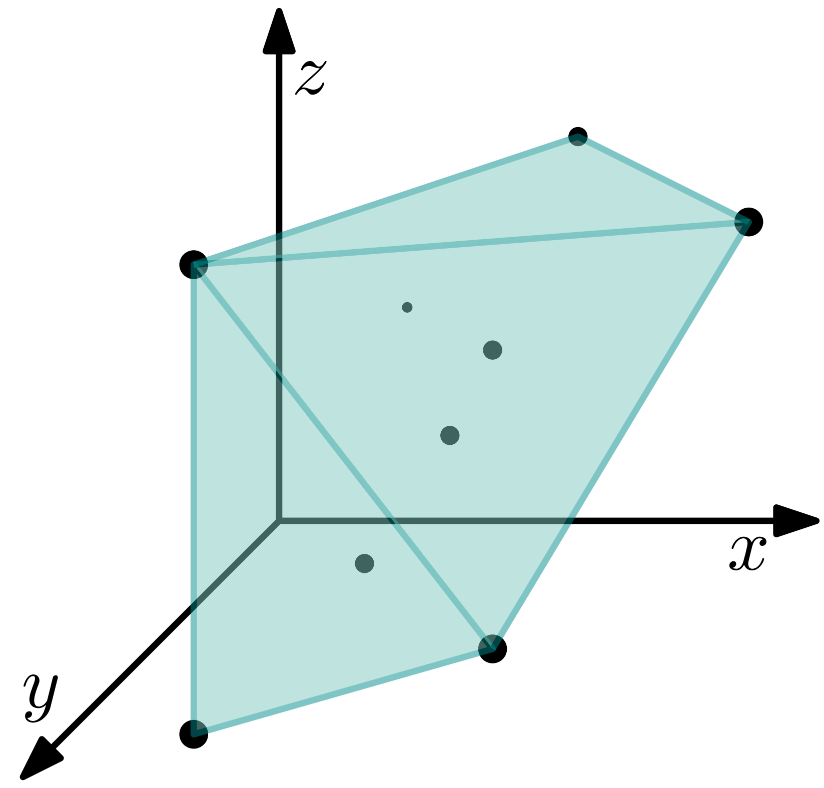
Was ist $CH(P)$ in $\mathbb{R}^3$?



Konvexe Hülle in $\mathbb{R}^3$

Def: Eine Menge $S \subseteq \mathbb{R}^3$ heißt **konvex**, wenn für je zwei Punkte $p, q \in S$ auch gilt $\overline{pq} \in S$.
Die **konvexe Hülle** $CH(P)$ einer Punktmenge P ist die kleinste konvexe Menge, die P enthält.

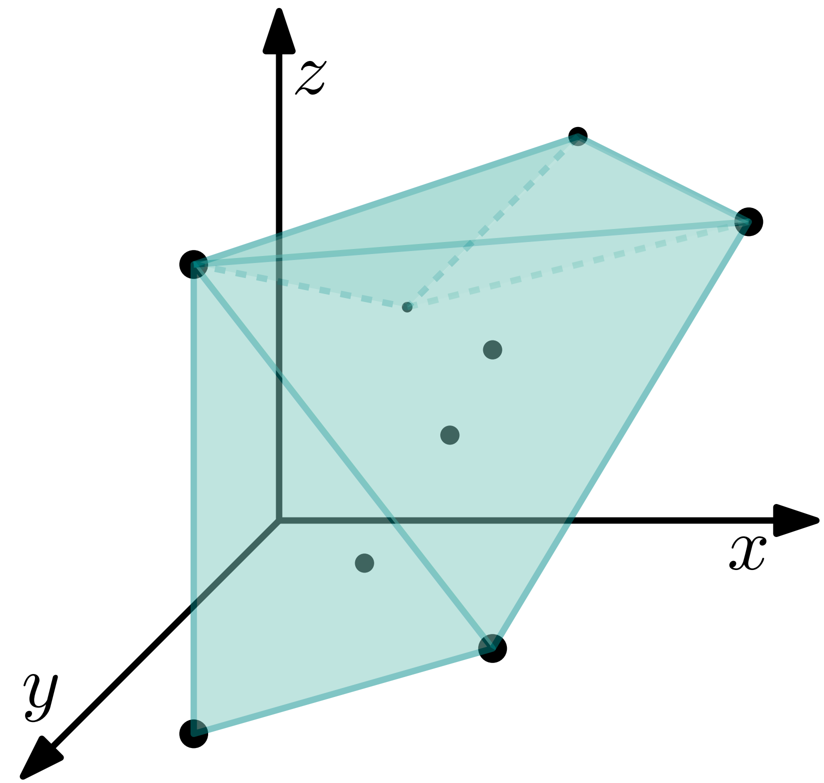
Was ist $CH(P)$ in $\mathbb{R}^3$?



Konvexe Hülle in $\mathbb{R}^3$

Def: Eine Menge $S \subseteq \mathbb{R}^3$ heißt **konvex**, wenn für je zwei Punkte $p, q \in S$ auch gilt $\overline{pq} \in S$.
Die **konvexe Hülle** $CH(P)$ einer Punktmenge P ist die kleinste konvexe Menge, die P enthält.

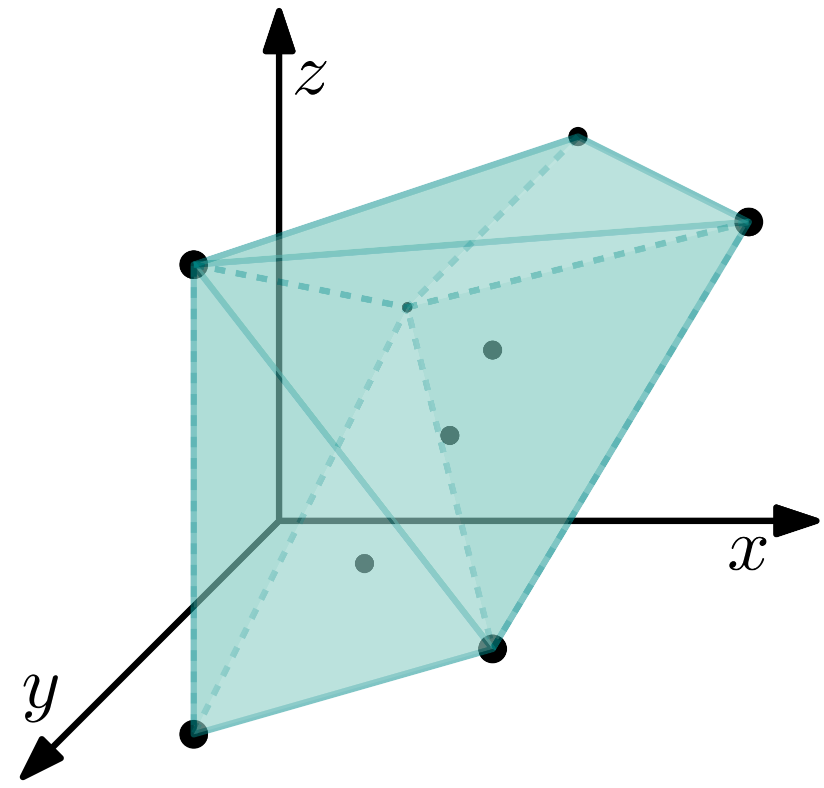
Was ist $CH(P)$ in $\mathbb{R}^3$?



Konvexe Hülle in $\mathbb{R}^3$

Def: Eine Menge $S \subseteq \mathbb{R}^3$ heißt **konvex**, wenn für je zwei Punkte $p, q \in S$ auch gilt $\overline{pq} \in S$.
Die **konvexe Hülle** $CH(P)$ einer Punktmenge P ist die kleinste konvexe Menge, die P enthält.

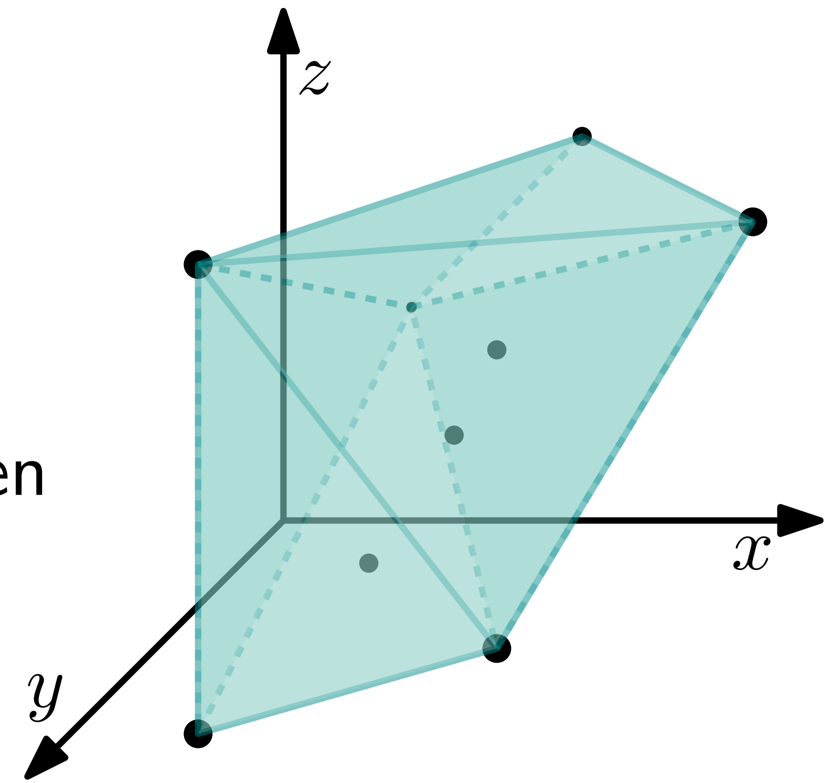
Was ist $CH(P)$ in $\mathbb{R}^3$?



Konvexe Hülle in $\mathbb{R}^3$

Def: Eine Menge $S \subseteq \mathbb{R}^3$ heißt **konvex**, wenn für je zwei Punkte $p, q \in S$ auch gilt $\overline{pq} \in S$.
Die **konvexe Hülle** $CH(P)$ einer Punktmenge P ist die kleinste konvexe Menge, die P enthält.

Was ist $CH(P)$ in $\mathbb{R}^3$?

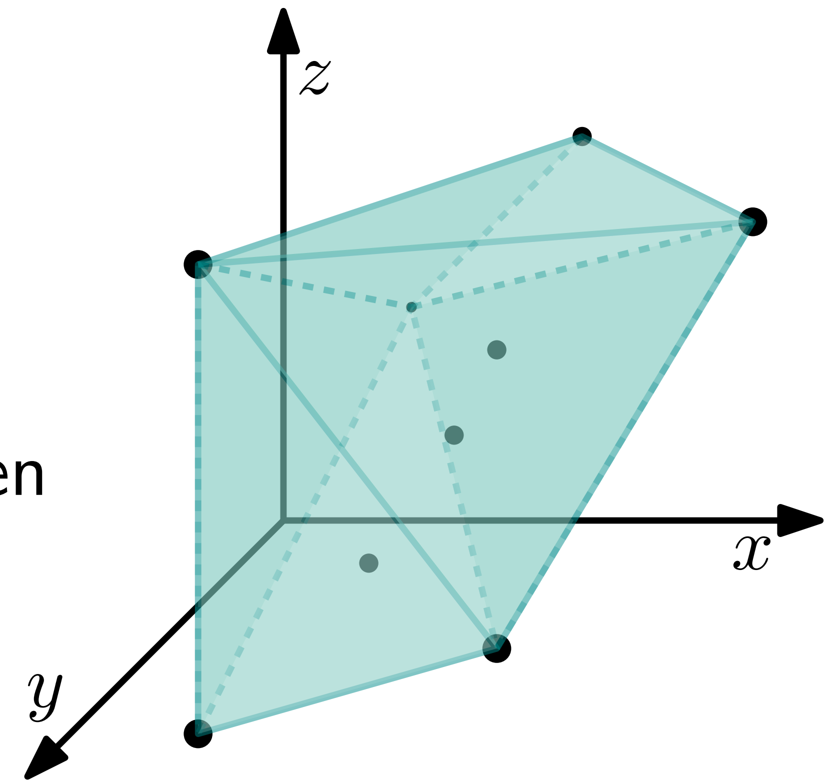


Satz 1:
$CH(P)$ ist ein konvexes Polyeder mit Ecken $\subseteq P$ ($|P| = n$). Für die Anzahlen f der Flächen und m der Kanten gilt

Konvexe Hülle in $\mathbb{R}^3$

Def: Eine Menge $S \subseteq \mathbb{R}^3$ heißt **konvex**, wenn für je zwei Punkte $p, q \in S$ auch gilt $\overline{pq} \in S$.
Die **konvexe Hülle** $CH(P)$ einer Punktmenge P ist die kleinste konvexe Menge, die P enthält.

Was ist $CH(P)$ in $\mathbb{R}^3$?



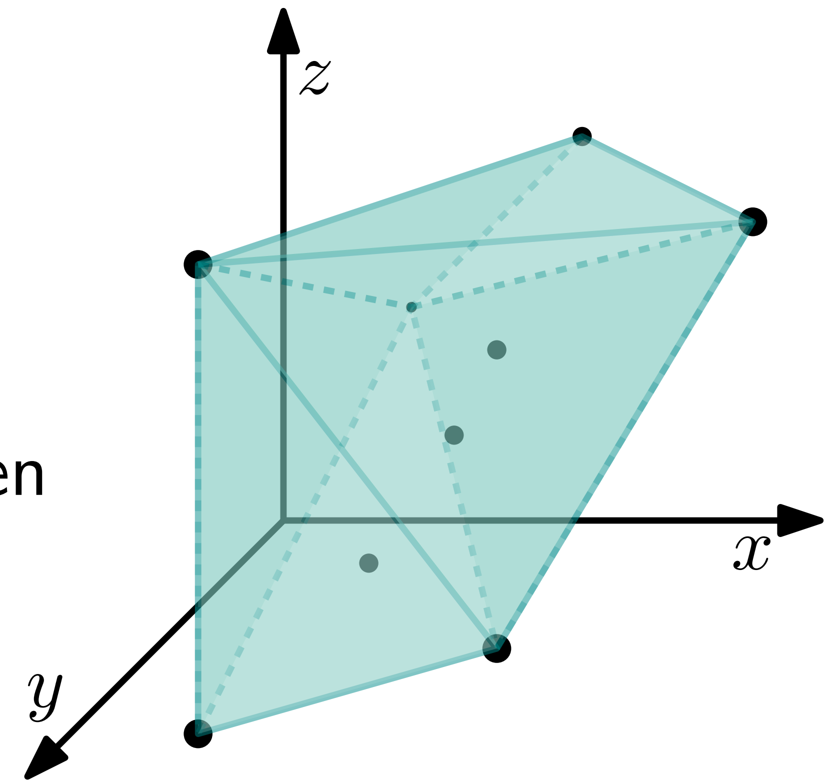
Satz 1:
$CH(P)$ ist ein konvexes Polyeder mit Ecken $\subseteq P$ ($|P| = n$). Für die Anzahlen f der Flächen und m der Kanten gilt $f \leq 2n - 4$ und $m \leq 3n - 6$.

Beweis: Satz von Euler $\square$

Konvexe Hülle in $\mathbb{R}^3$

Def: Eine Menge $S \subseteq \mathbb{R}^3$ heißt **konvex**, wenn für je zwei Punkte $p, q \in S$ auch gilt $\overline{pq} \in S$.
Die **konvexe Hülle** $CH(P)$ einer Punktmenge P ist die kleinste konvexe Menge, die P enthält.

Was ist $CH(P)$ in $\mathbb{R}^3$?



Satz 1:
$CH(P)$ ist ein konvexes Polyeder mit Ecken $\subseteq P$ ($|P| = n$). Für die Anzahlen f der Flächen und m der Kanten gilt $f \leq 2n - 4$ und $m \leq 3n - 6$.

Beweis: Satz von Euler □

$\rightarrow$ Repräsentation über DCEL wie für planare Graphen

Inkrementelle Konstruktion

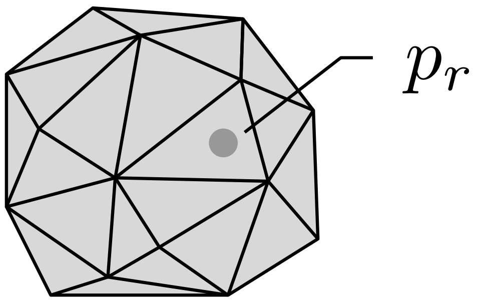
Angenommen wir kennen $CH(P_{r-1})$ für die ersten $r-1$ Punkte $P_{r-1} = \{p_1, \dots, p_{r-1}\}$, wie bestimmt man $CH(P_r)$?

Inkrementelle Konstruktion

Angenommen wir kennen $CH(P_{r-1})$ für die ersten $r-1$ Punkte $P_{r-1} = \{p_1, \ldots, p_{r-1}\}$, wie bestimmt man $CH(P_r)$?

Fall 1: $p_r \in CH(P_{r-1})$

$\Rightarrow CH(P_r) = CH(P_{r-1})$

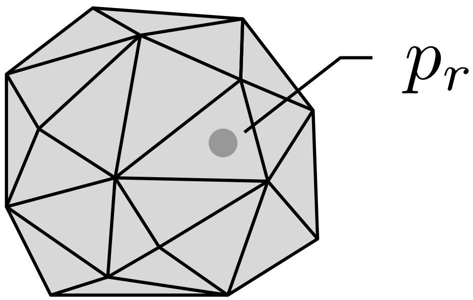


Inkrementelle Konstruktion

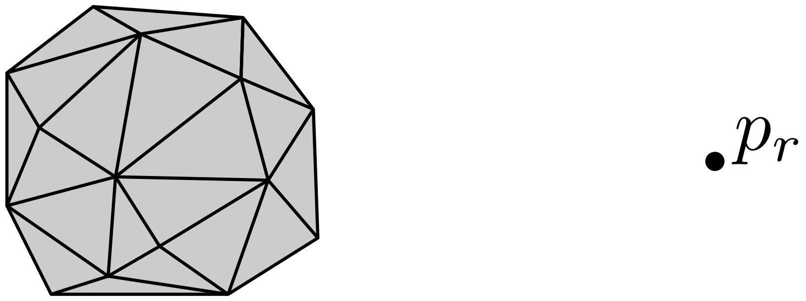
Angenommen wir kennen $CH(P_{r-1})$ für die ersten $r-1$ Punkte $P_{r-1} = \{p_1, \ldots, p_{r-1}\}$, wie bestimmt man $CH(P_r)$?

Fall 1: $p_r \in CH(P_{r-1})$

$\Rightarrow CH(P_r) = CH(P_{r-1})$



Fall 2: $p_r \notin CH(P_{r-1})$

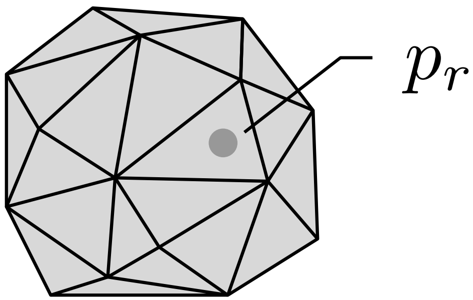


Inkrementelle Konstruktion

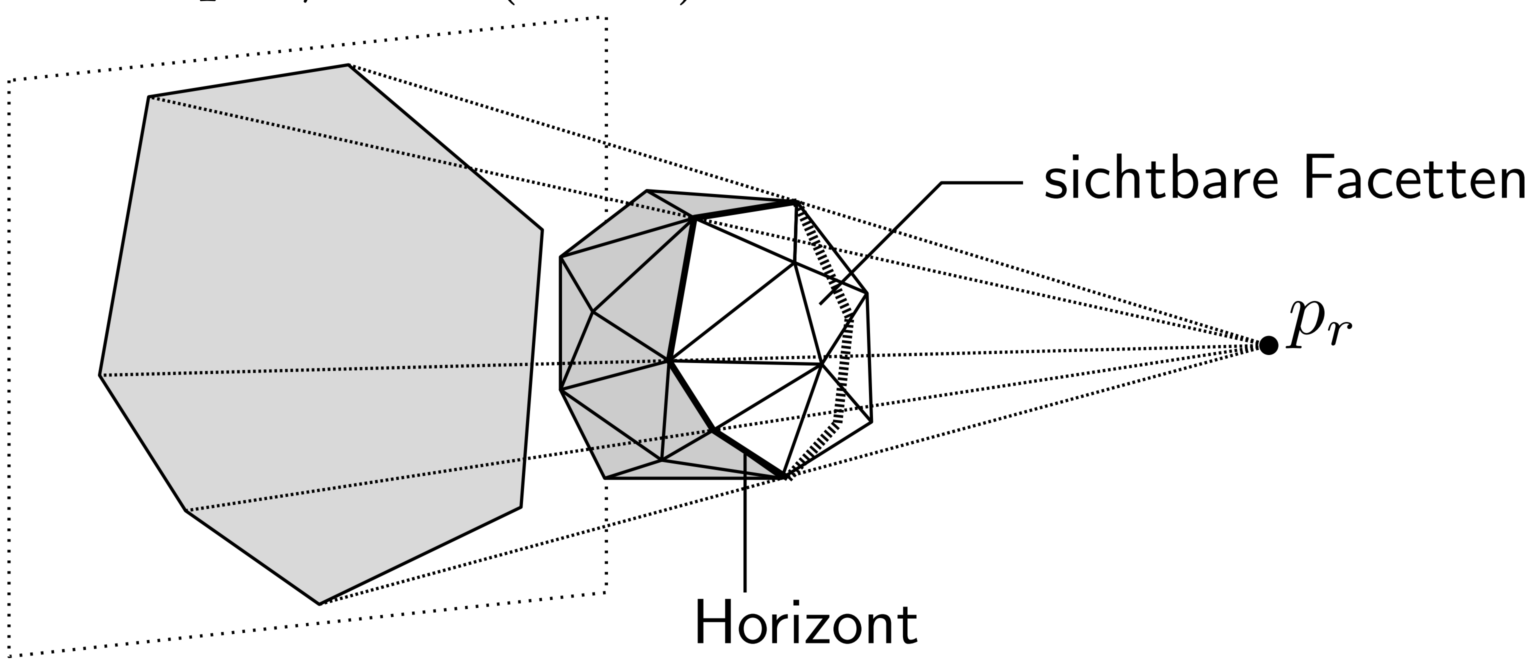
Angenommen wir kennen $CH(P_{r-1})$ für die ersten $r-1$ Punkte $P_{r-1} = \{p_1, \ldots, p_{r-1}\}$, wie bestimmt man $CH(P_r)$?

Fall 1: $p_r \in CH(P_{r-1})$

$$\Rightarrow CH(P_r) = CH(P_{r-1})$$



Fall 2: $p_r \notin CH(P_{r-1})$

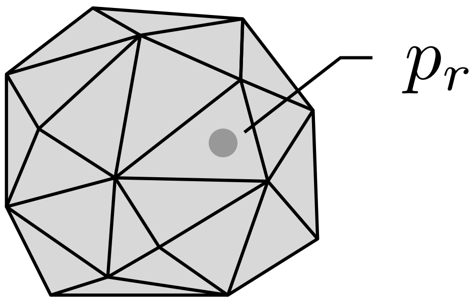


Inkrementelle Konstruktion

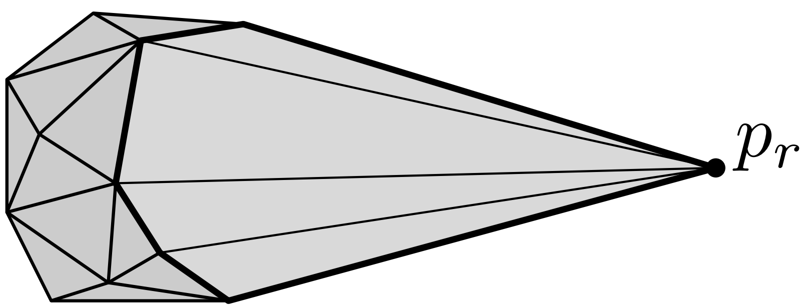
Angenommen wir kennen $CH(P_{r-1})$ für die ersten $r-1$ Punkte $P_{r-1} = \{p_1, \ldots, p_{r-1}\}$, wie bestimmt man $CH(P_r)$?

Fall 1: $p_r \in CH(P_{r-1})$

$\Rightarrow CH(P_r) = CH(P_{r-1})$



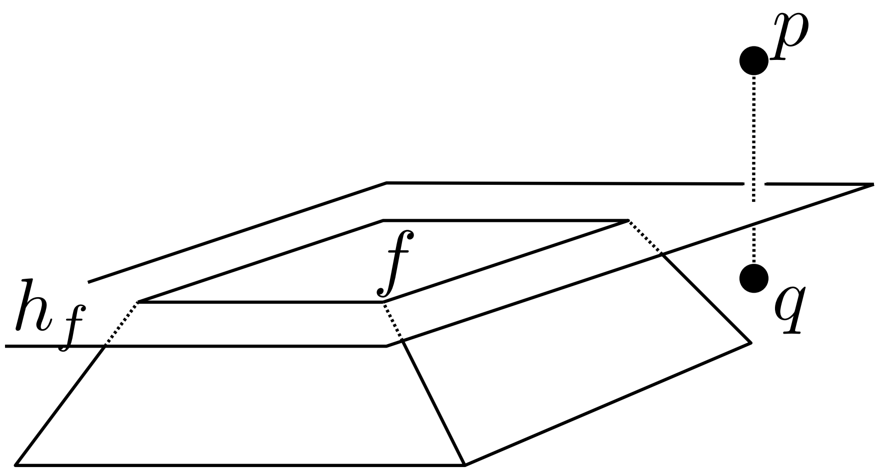
Fall 2: $p_r \notin CH(P_{r-1})$



Ersetze sichtbare Facetten durch neue Dreiecke mit allen Horizontkanten

Sichtbare Facetten

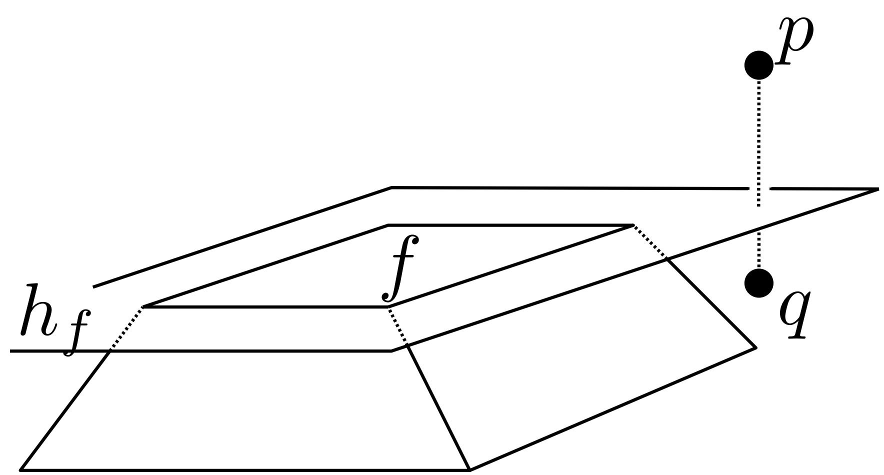
Wann ist eine Facette f von $CH(P_{r-1})$ von p_r aus sichtbar?



Facette f sichtbar von p aber nicht von q aus

Sichtbare Facetten

Wann ist eine Facette f von $CH(P_{r-1})$ von p_r aus sichtbar?

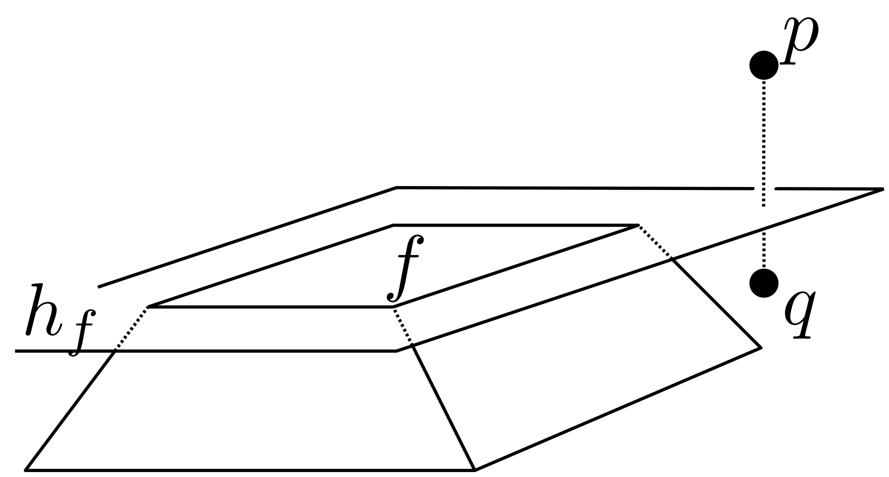


Facette f sichtbar von p aber nicht von q aus

- betrachte Halbebene h_f, die f enthält
- $CH(P_{r-1})$ liegt komplett auf einer Seite von h_f
- alle Punkte auf der offenen anderen Seite von h_f sehen f

Sichtbare Facetten

Wann ist eine Facette f von $CH(P_{r-1})$ von p_r aus sichtbar?



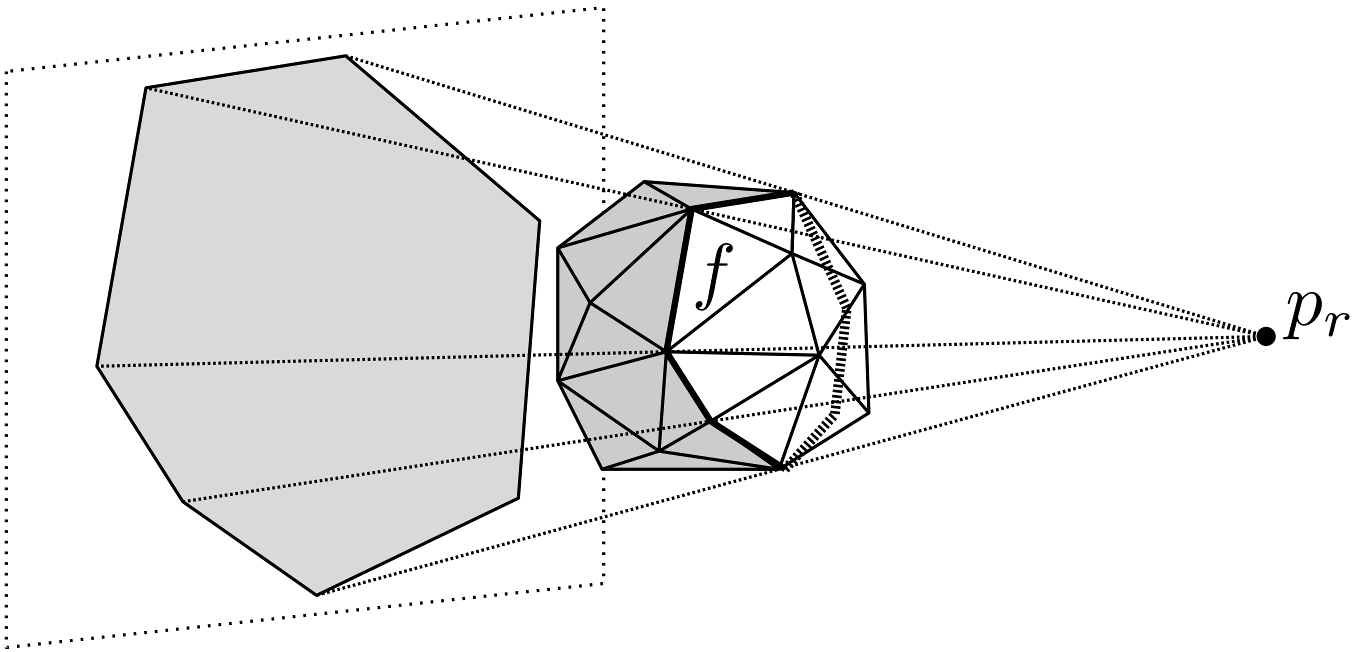
Facette f sichtbar von p aber nicht von q aus

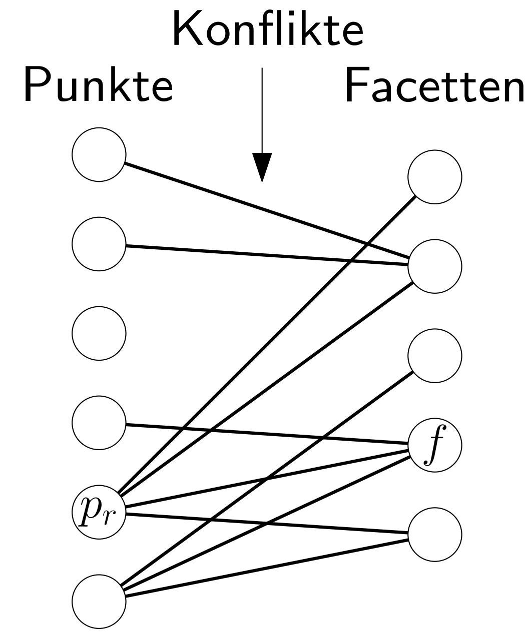
- betrachte Halbebene h_f, die f enthält
- $CH(P_{r-1})$ liegt komplett auf einer Seite von h_f
- alle Punkte auf der offenen anderen Seite von h_f sehen f

Aufgabe: entwerfen Sie einen inkrementellen Algorithmus zur Berechnung der konvexen Hülle

Schnelle Bestimmung der sichtbaren Facetten

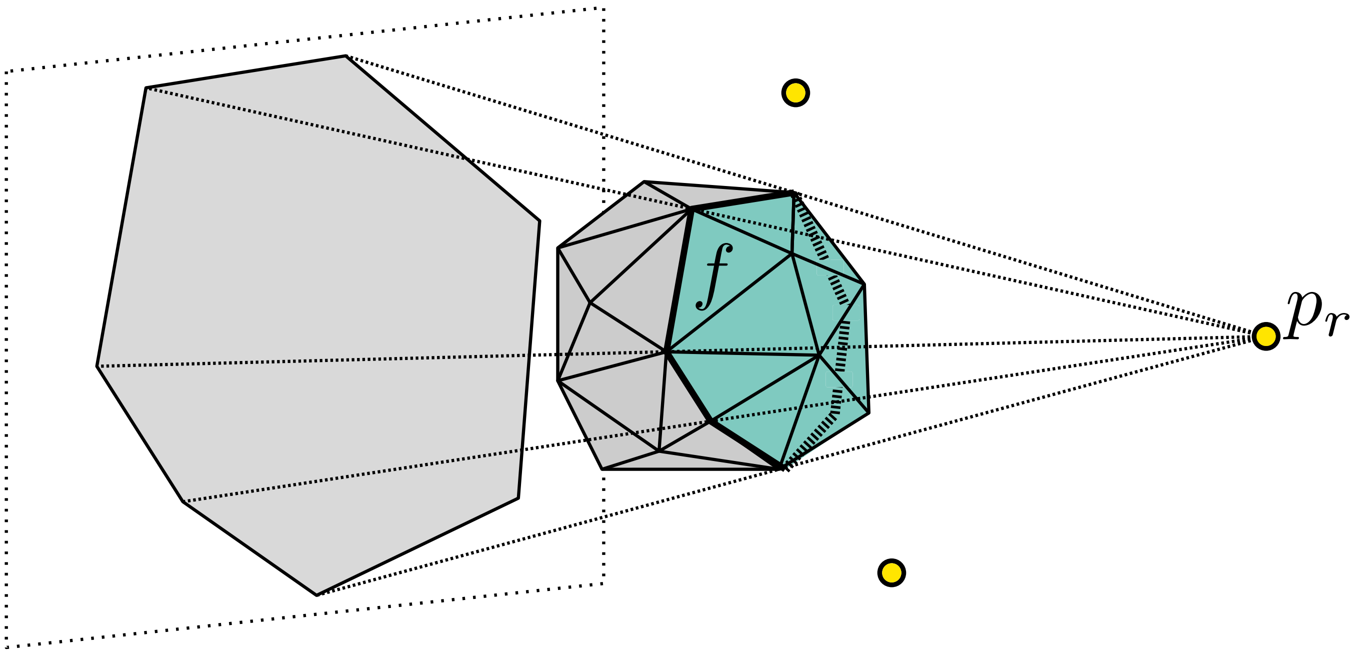
Erstelle bipartiten **Konfliktgraph** G, der verbleibende Punkte, aktuelle Facetten und Sichtbarkeiten (Konflikte) repräsentiert

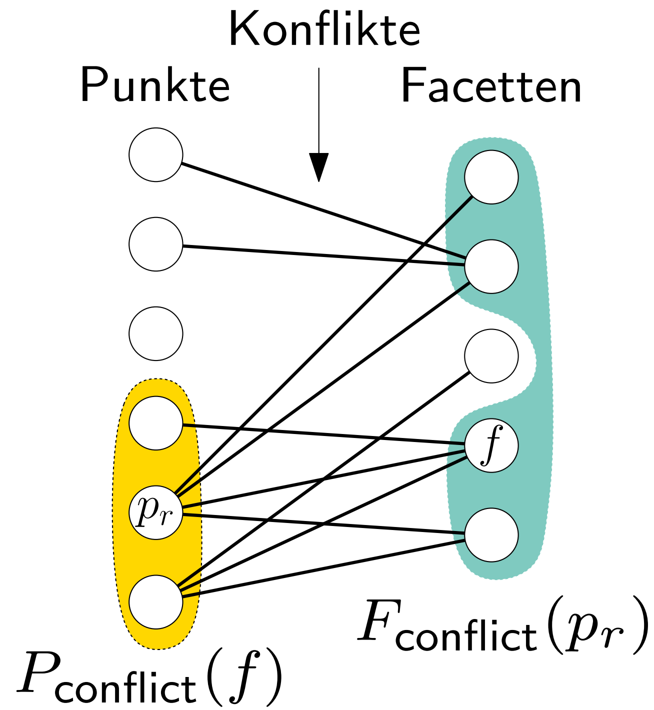




Schnelle Bestimmung der sichtbaren Facetten

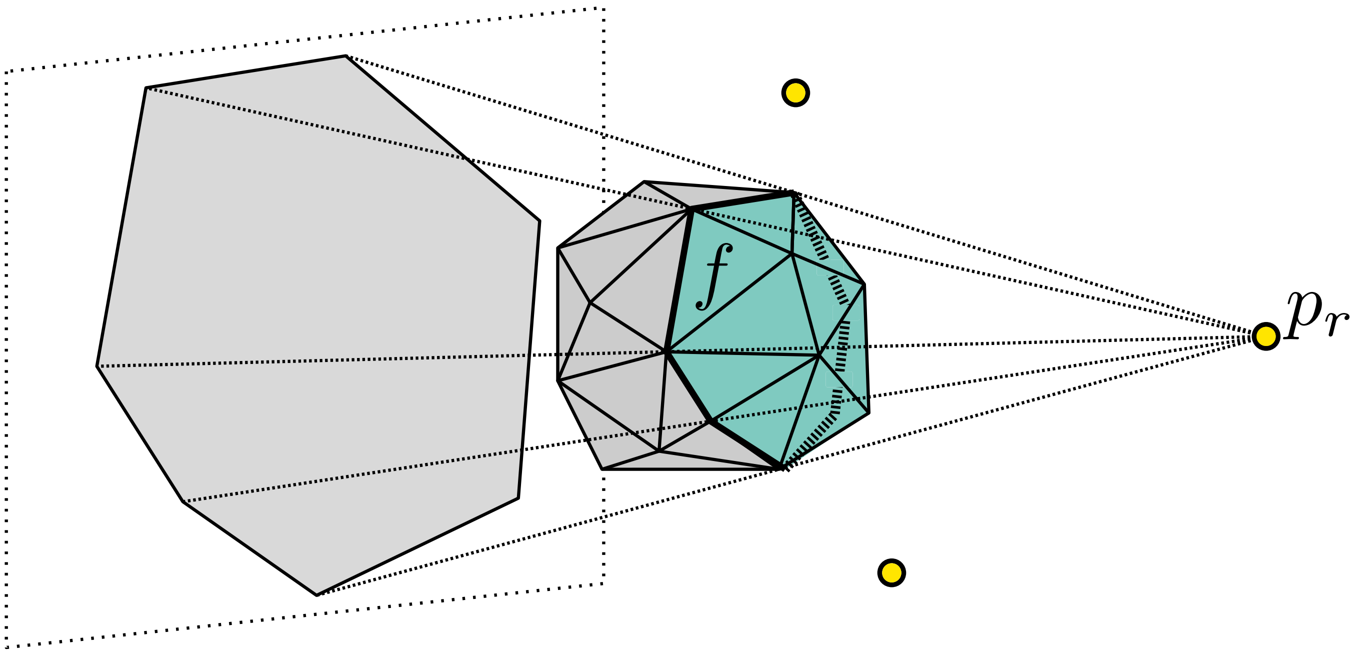
Erstelle bipartiten **Konfliktgraph** G, der verbleibende Punkte, aktuelle Facetten und Sichtbarkeiten (Konflikte) repräsentiert

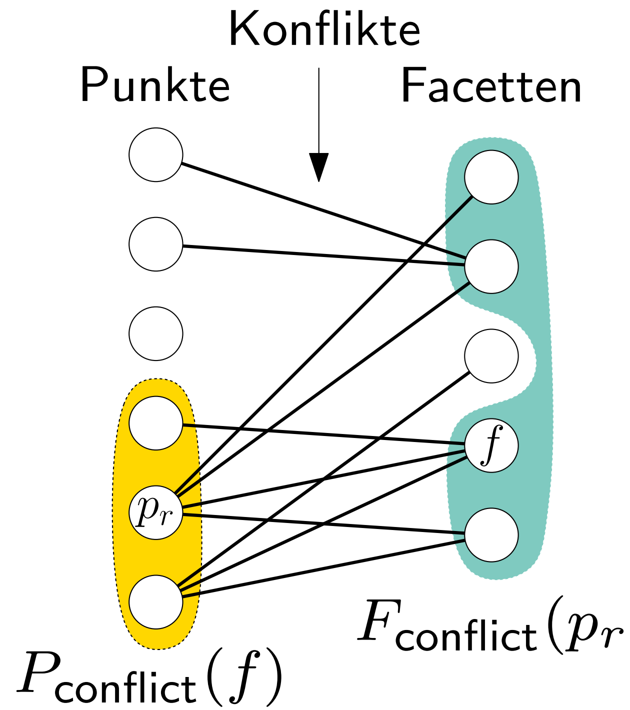




Schnelle Bestimmung der sichtbaren Facetten

Erstelle bipartiten **Konfliktgraph** G, der verbleibende Punkte, aktuelle Facetten und Sichtbarkeiten (Konflikte) repräsentiert

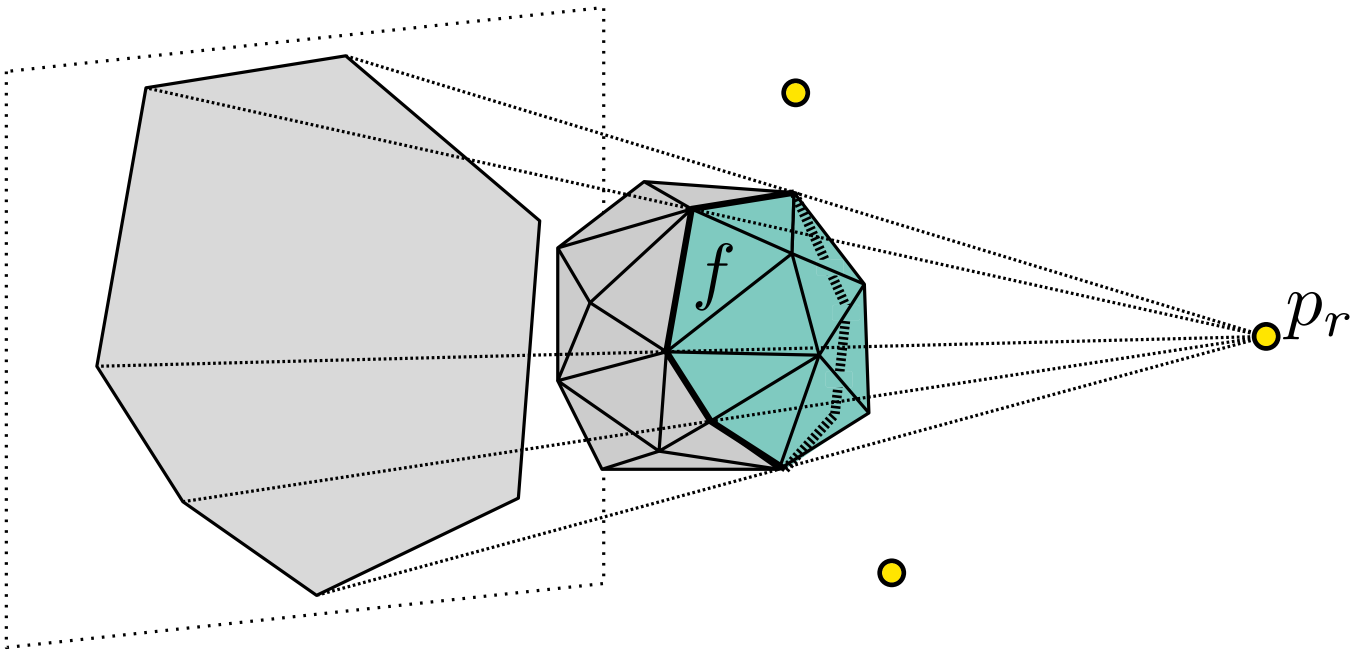


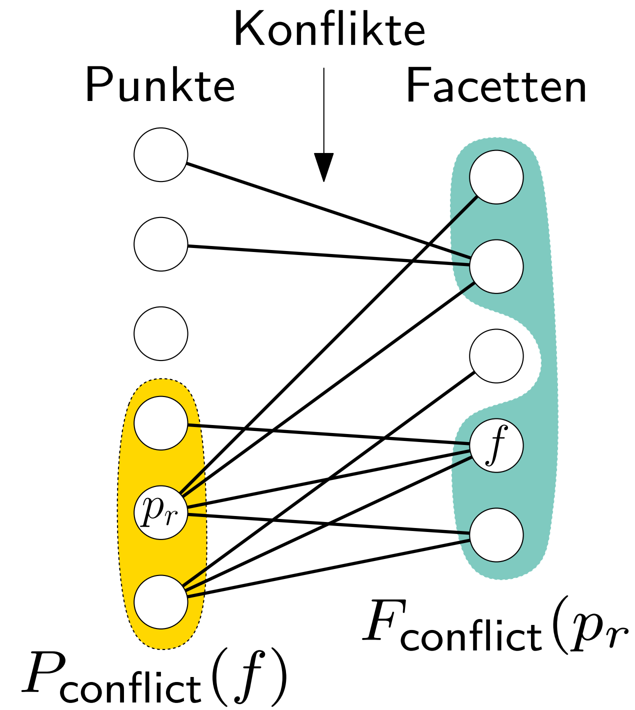


- $P_{\text{conflict}}(f) = \{$unbearbeitete Punkte, die f sehen$\}$
- $F_{\text{conflict}}(p_r) = \{$Facetten der konvexen Hülle, die p_r sieht$\}$
- Kanten zw. f und $P_{\text{conflict}}(f)$ bzw. zw. p und $F_{\text{conflict}}(p)$

Schnelle Bestimmung der sichtbaren Facetten

Erstelle bipartiten **Konfliktgraph** G, der verbleibende Punkte, aktuelle Facetten und Sichtbarkeiten (Konflikte) repräsentiert



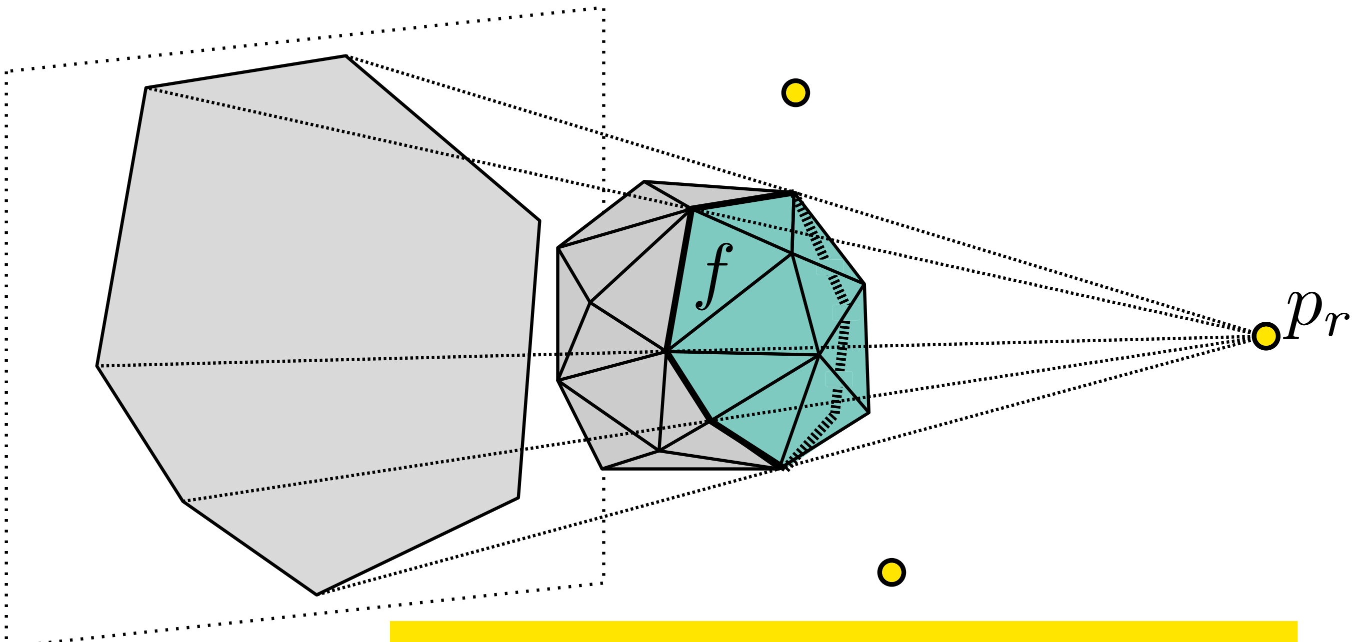


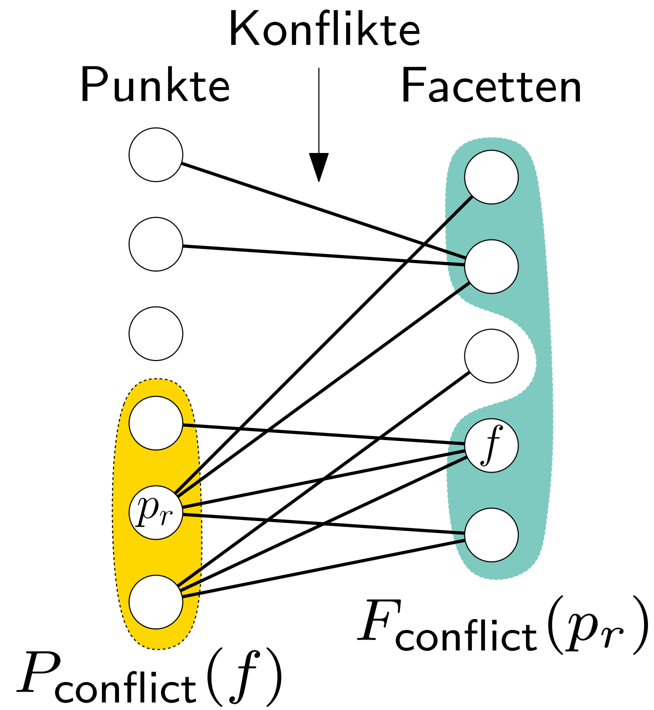
- $P_{\text{conflict}}(f) = \{$unbearbeitete Punkte, die f sehen$\}$
- $F_{\text{conflict}}(p_r) = \{$Facetten der konvexen Hülle, die p_r sieht$\}$
- Kanten zw. f und $P_{\text{conflict}}(f)$ bzw. zw. p und $F_{\text{conflict}}(p)$

$\rightarrow$ sichtbare Facetten und Horizont aus $F_{\text{conflict}}(p_r)$ ablesbar

Schnelle Bestimmung der sichtbaren Facetten

Erstelle bipartiten **Konfliktgraph** G, der verbleibende Punkte, aktuelle Facetten und Sichtbarkeiten (Konflikte) repräsentiert





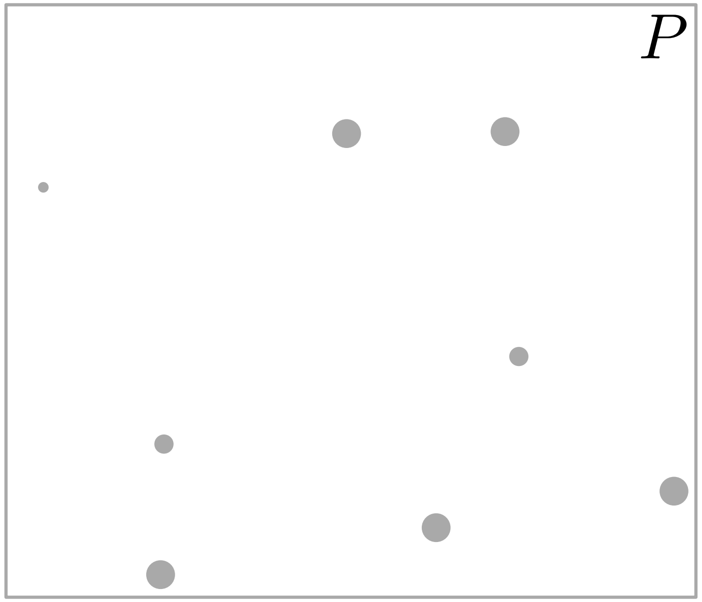
Wie aktualisiert man G?

- $P_{\text{conflict}}(f) = \{$unbearbeitete Punkte, die f sehen$\}$
- $F_{\text{conflict}}(p_r) = \{$Facetten der konvexen Hülle, die p_r sieht$\}$
- Kanten zw. f und $P_{\text{conflict}}(f)$ bzw. zw. p und $F_{\text{conflict}}(p)$

$\boxed{\rightarrow \text{ sichtbare Facetten und Horizont aus } F_{\text{conflict}}(p_r) \text{ ablesbar}}$

3DConvexHull-Algorithmus

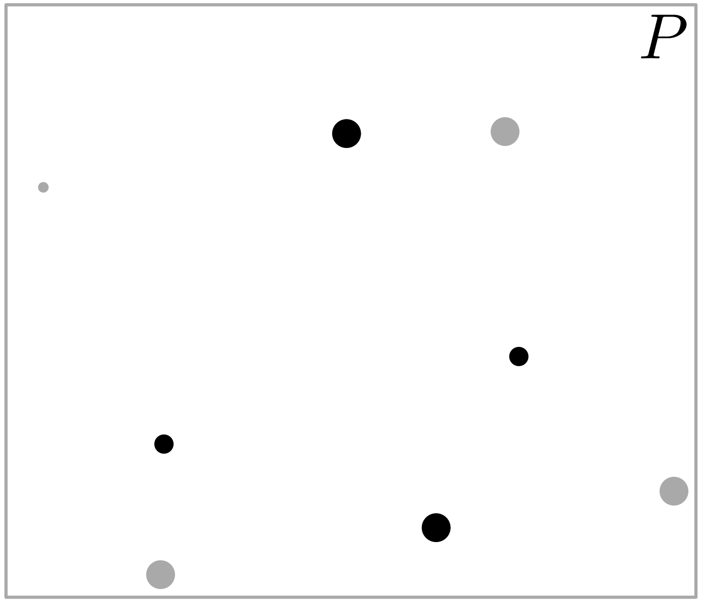
3dConvexHull(P)



3DConvexHull-Algorithmus

3dConvexHull(P)

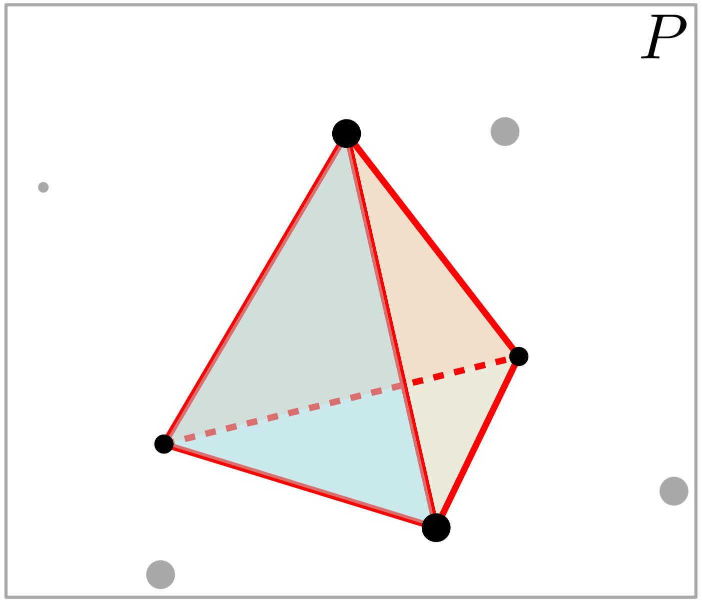
$P' \leftarrow \{p_1, p_2, p_3, p_4\} \subseteq P$ (nicht koplan.)



3DConvexHull-Algorithmus

3dConvexHull(P)

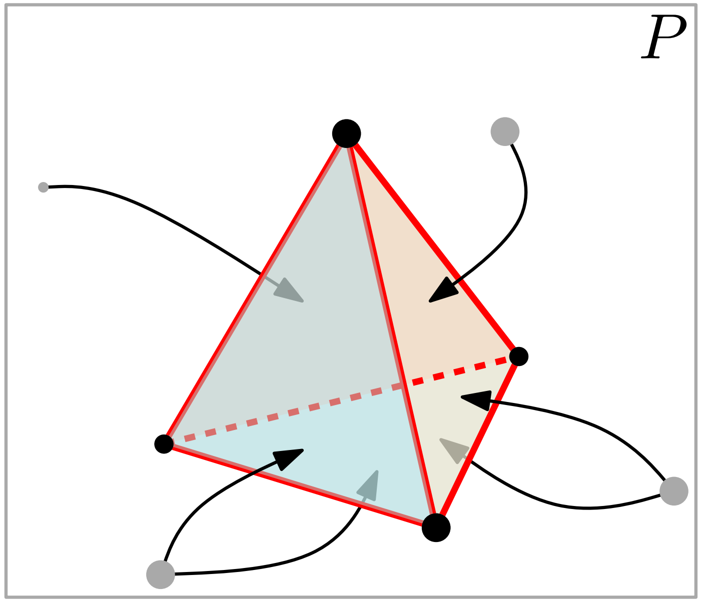
$P' \leftarrow \{p_1, p_2, p_3, p_4\} \subseteq P$ (nicht koplan.)
$C \leftarrow \mathrm{CH}(P')$



3DConvexHull-Algorithmus

3dConvexHull(P)

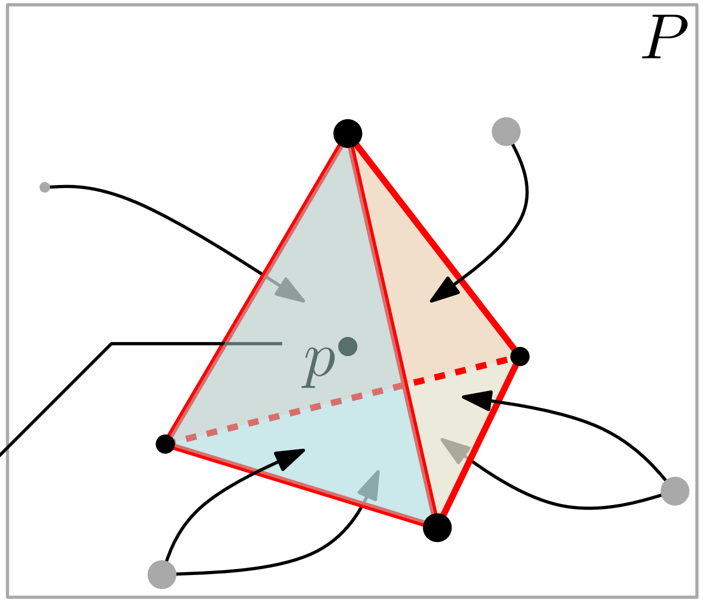
$P' \leftarrow \{p_1, p_2, p_3, p_4\} \subseteq P$ (nicht koplan.)
$C \leftarrow \mathrm{CH}(P')$
initialisiere Konfliktgraph G



3DConvexHull-Algorithmus

3dConvexHull(P)

$P' \leftarrow \{p_1, p_2, p_3, p_4\} \subseteq P$ (nicht koplan.)
$C \leftarrow$ CH(P')
initialisiere Konfliktgraph G
foreach $p \in P \setminus P'$ in Zufallsreihenfolge **do**
    if $F_{\text{conflict}}(p) \neq \emptyset$ **then**

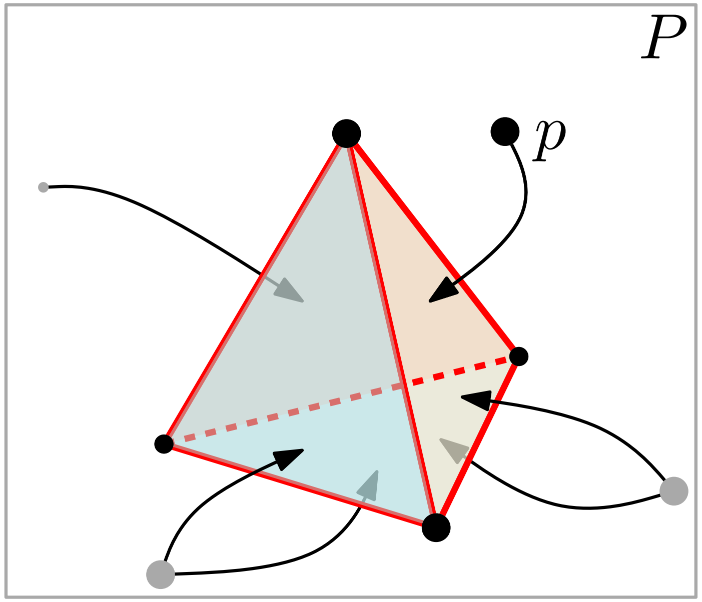


hier: $F_{\text{conflict}}(p) = \emptyset$, d.h. p liegt in $C \rightarrow$ nichts zu tun

3DConvexHull-Algorithmus

3dConvexHull(P)

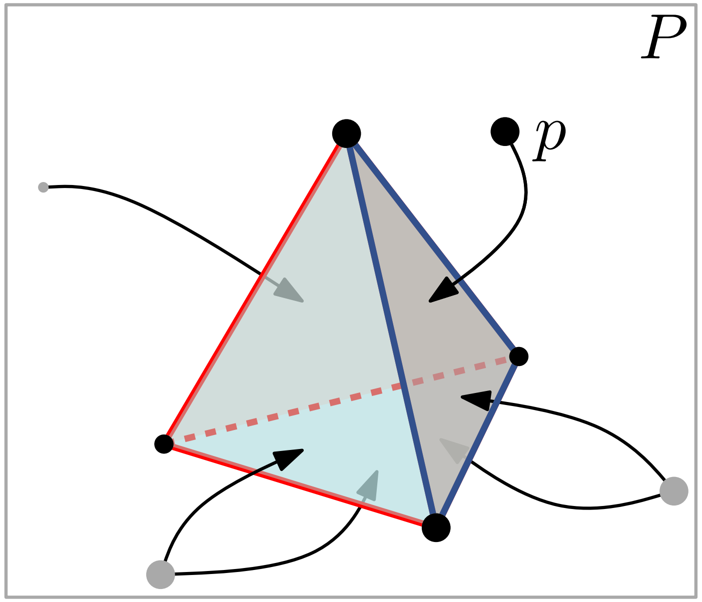
$P' \leftarrow \{p_1, p_2, p_3, p_4\} \subseteq P$ (nicht koplan.)
$C \leftarrow \mathrm{CH}(P')$
initialisiere Konfliktgraph G
foreach $p \in P \setminus P'$ in Zufallsreihenfolge **do**
  if $F_{\text{conflict}}(p) \neq \emptyset$ **then**



3DConvexHull-Algorithmus

3dConvexHull(P)

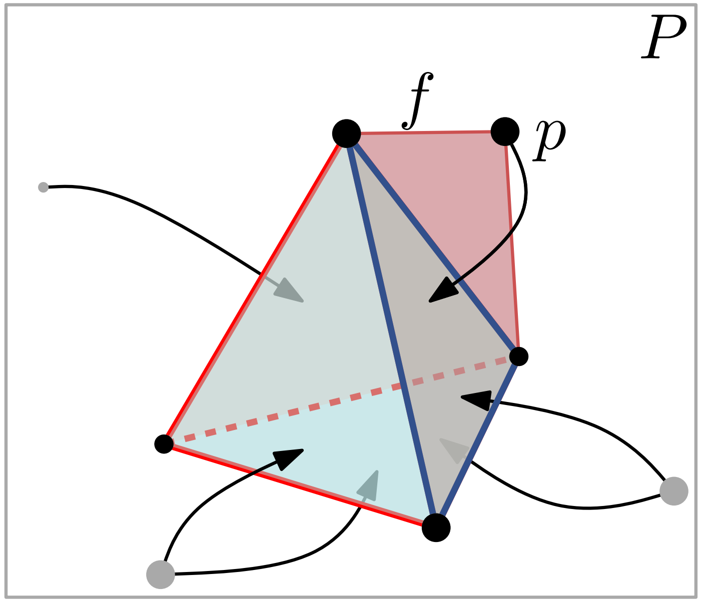
$P' \leftarrow \{p_1, p_2, p_3, p_4\} \subseteq P$ (nicht koplan.)
$C \leftarrow \mathrm{CH}(P')$
initialisiere Konfliktgraph G
foreach $p \in P \setminus P'$ in Zufallsreihenfolge **do**
  if $F_{\text{conflict}}(p) \neq \emptyset$ **then**
    lösche alle Facetten $F_{\text{conflict}}(p)$ aus C
    $\mathcal{L} \leftarrow$ Horizont-Kanten von p



3DConvexHull-Algorithmus

3dConvexHull(P)

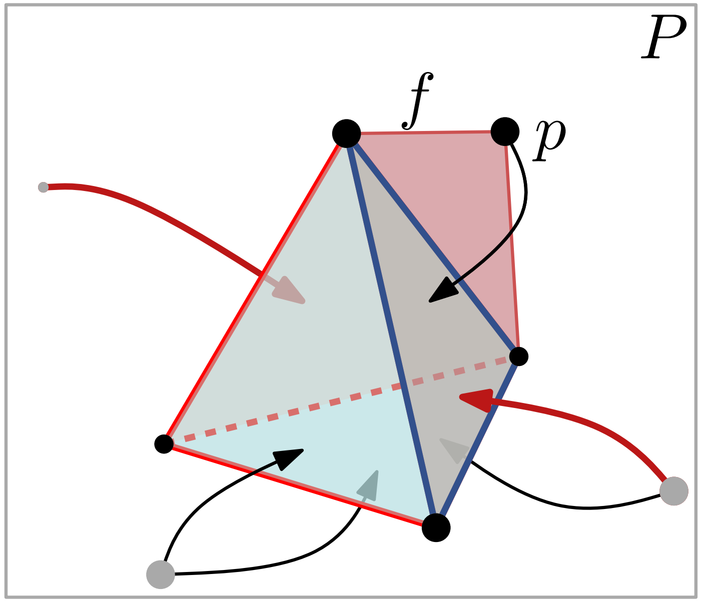
$P' \leftarrow \{p_1, p_2, p_3, p_4\} \subseteq P$ (nicht koplan.)
$C \leftarrow \mathrm{CH}(P')$
initialisiere Konfliktgraph G
foreach $p \in P \setminus P'$ in Zufallsreihenfolge **do**
    if $F_{\text{conflict}}(p) \neq \emptyset$ **then**
        lösche alle Facetten $F_{\text{conflict}}(p)$ aus C
        $\mathcal{L} \leftarrow$ Horizont-Kanten von p
        foreach $e \in \mathcal{L}$ **do**
            $f \leftarrow$ neue Facette zu (e, p) in C; $v_f \leftarrow$ neuer Knoten für f in G



3DConvexHull-Algorithmus

3dConvexHull(P)

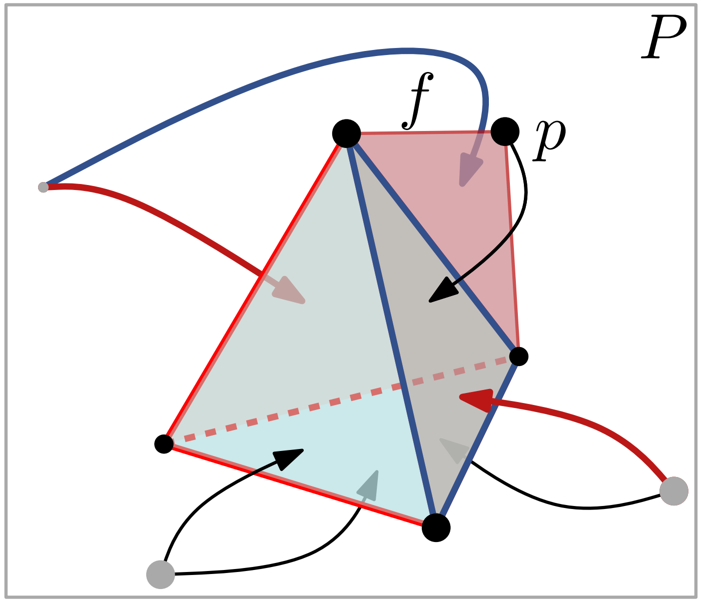
$P' \leftarrow \{p_1, p_2, p_3, p_4\} \subseteq P$ (nicht koplan.)
$C \leftarrow \mathsf{CH}(P')$
initialisiere Konfliktgraph G
foreach $p \in P \setminus P'$ in Zufallsreihenfolge **do**
  if $F_{\text{conflict}}(p) \neq \emptyset$ **then**
    lösche alle Facetten $F_{\text{conflict}}(p)$ aus C
    $\mathcal{L} \leftarrow$ Horizont-Kanten von p
    foreach $e \in \mathcal{L}$ **do**
      $f \leftarrow$ neue Facette zu (e, p) in C; $v_f \leftarrow$ neuer Knoten für f in G
      $(f_1, f_2) \leftarrow$ Facetten vorher inzident zu e in C
      $P(e) \leftarrow P_{\text{conflict}}(f_1) \cup P_{\text{conflict}}(f_2)$



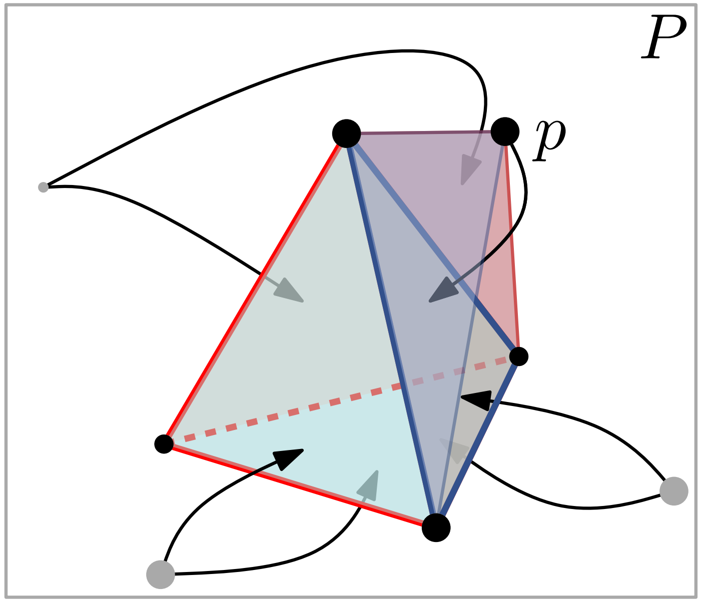
3DConvexHull-Algorithmus

3dConvexHull(P)

$P' \leftarrow \{p_1, p_2, p_3, p_4\} \subseteq P$ (nicht koplan.)
$C \leftarrow$ CH(P')
initialisiere Konfliktgraph G
foreach $p \in P \setminus P'$ in Zufallsreihenfolge **do**
  if $F_{\text{conflict}}(p) \neq \emptyset$ **then**
    lösche alle Facetten $F_{\text{conflict}}(p)$ aus C
    $\mathcal{L} \leftarrow$ Horizont-Kanten von p
    foreach $e \in \mathcal{L}$ **do**
      $f \leftarrow$ neue Facette zu (e, p) in C; $v_f \leftarrow$ neuer Knoten für f in G
      $(f_1, f_2) \leftarrow$ Facetten vorher inzident zu e in C
      $P(e) \leftarrow P_{\text{conflict}}(f_1) \cup P_{\text{conflict}}(f_2)$
      foreach $p \in P(e)$ **do**
        if f sichtbar von p **then** füge Kante (p, f) zu G hinzu



3DConvexHull-Algorithmus



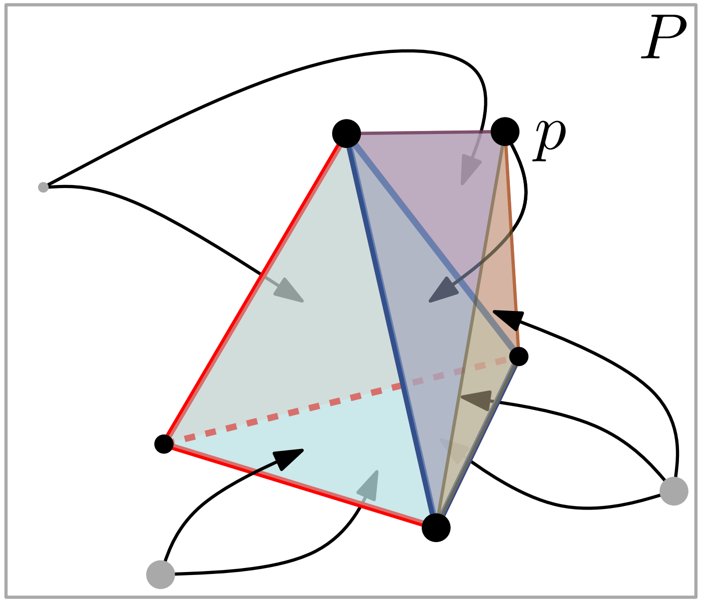
3dConvexHull(P)

$P' \leftarrow \{p_1, p_2, p_3, p_4\} \subseteq P$ (nicht koplan.)
$C \leftarrow \mathrm{CH}(P')$
initialisiere Konfliktgraph G
foreach $p \in P \setminus P'$ in Zufallsreihenfolge **do**
  if $F_{\text{conflict}}(p) \neq \emptyset$ **then**
    lösche alle Facetten $F_{\text{conflict}}(p)$ aus C
    $\mathcal{L} \leftarrow$ Horizont-Kanten von p
    foreach $e \in \mathcal{L}$ **do**
      $f \leftarrow$ neue Facette zu (e, p) in C; $v_f \leftarrow$ neuer Knoten für f in G
      $(f_1, f_2) \leftarrow$ Facetten vorher inzident zu e in C
      $P(e) \leftarrow P_{\text{conflict}}(f_1) \cup P_{\text{conflict}}(f_2)$
      foreach $p \in P(e)$ **do**
        if f sichtbar von p **then** füge Kante (p, f) zu G hinzu

3DConvexHull-Algorithmus

3dConvexHull(P)

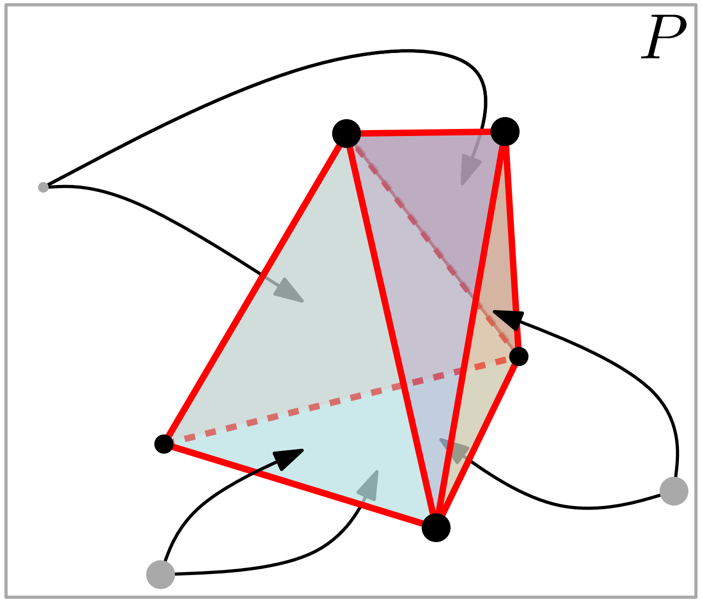
$P' \leftarrow \{p_1, p_2, p_3, p_4\} \subseteq P$ (nicht koplan.)
$C \leftarrow \mathrm{CH}(P')$
initialisiere Konfliktgraph G
foreach $p \in P \setminus P'$ in Zufallsreihenfolge **do**
  if $F_{\text{conflict}}(p) \neq \emptyset$ **then**
    lösche alle Facetten $F_{\text{conflict}}(p)$ aus C
    $\mathcal{L} \leftarrow$ Horizont-Kanten von p
    foreach $e \in \mathcal{L}$ **do**
      $f \leftarrow$ neue Facette zu (e, p) in C; $v_f \leftarrow$ neuer Knoten für f in G
      $(f_1, f_2) \leftarrow$ Facetten vorher inzident zu e in C
      $P(e) \leftarrow P_{\text{conflict}}(f_1) \cup P_{\text{conflict}}(f_2)$
      foreach $p \in P(e)$ **do**
        if f sichtbar von p **then** füge Kante (p, f) zu G hinzu



3DConvexHull-Algorithmus

3dConvexHull(P)

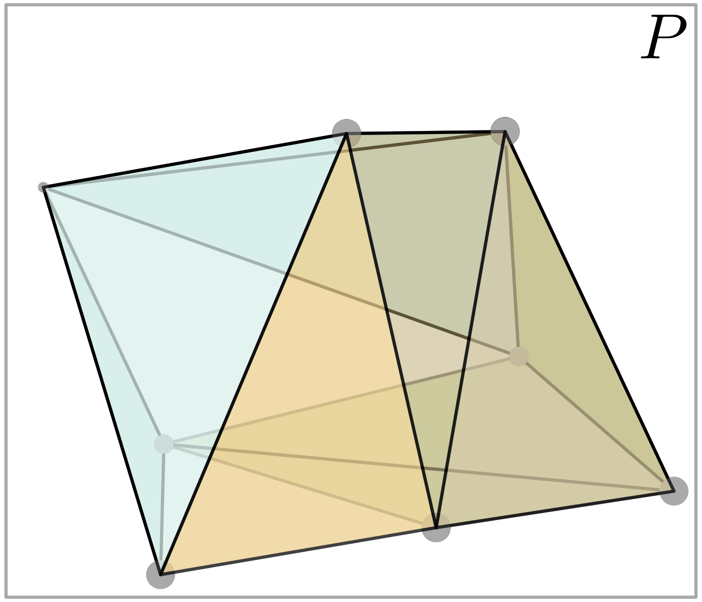
$P' \leftarrow \{p_1, p_2, p_3, p_4\} \subseteq P$ (nicht koplan.)
$C \leftarrow \mathrm{CH}(P')$
initialisiere Konfliktgraph G
foreach $p \in P \setminus P'$ in Zufallsreihenfolge **do**
  if $F_{\text{conflict}}(p) \neq \emptyset$ **then**
    lösche alle Facetten $F_{\text{conflict}}(p)$ aus C
    $\mathcal{L} \leftarrow$ Horizont-Kanten von p
    foreach $e \in \mathcal{L}$ **do**
      $f \leftarrow$ neue Facette zu (e, p) in C; $v_f \leftarrow$ neuer Knoten für f in G
      $(f_1, f_2) \leftarrow$ Facetten vorher inzident zu e in C
      $P(e) \leftarrow P_{\text{conflict}}(f_1) \cup P_{\text{conflict}}(f_2)$
      foreach $p \in P(e)$ **do**
        if f sichtbar von p **then** füge Kante (p, f) zu G hinzu
    lösche $\{p\} \cup F_{\text{conflict}}(p)$ aus G



3DConvexHull-Algorithmus

3dConvexHull(P)

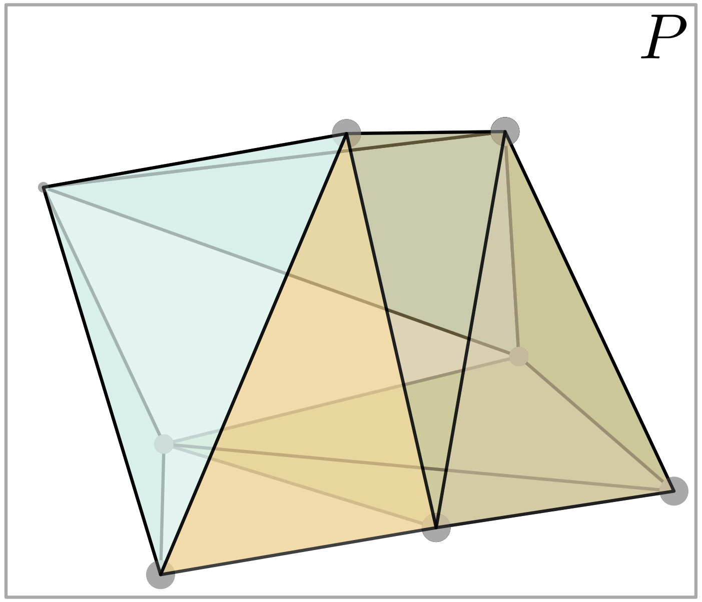
$P' \leftarrow \{p_1, p_2, p_3, p_4\} \subseteq P$ (nicht koplan.)
$C \leftarrow \mathsf{CH}(P')$
initialisiere Konfliktgraph G
foreach $p \in P \setminus P'$ in Zufallsreihenfolge **do**
  if $F_{\text{conflict}}(p) \neq \emptyset$ **then**
    lösche alle Facetten $F_{\text{conflict}}(p)$ aus C
    $\mathcal{L} \leftarrow$ Horizont-Kanten von p
    foreach $e \in \mathcal{L}$ **do**
      $f \leftarrow$ neue Facette zu (e, p) in C; $v_f \leftarrow$ neuer Knoten für f in G
      $(f_1, f_2) \leftarrow$ Facetten vorher inzident zu e in C
      $P(e) \leftarrow P_{\text{conflict}}(f_1) \cup P_{\text{conflict}}(f_2)$
      foreach $p \in P(e)$ **do**
        if f sichtbar von p **then** füge Kante (p, f) zu G hinzu
    lösche $\{p\} \cup F_{\text{conflict}}(p)$ aus G
return C



3DConvexHull-Algorithmus

3dConvexHull(P)

$P' \leftarrow \{p_1, p_2, p_3, p_4\} \subseteq P$ (nicht koplan.)
$C \leftarrow$ CH(P')
initialisiere Konfliktgraph G
foreach $p \in P \setminus P'$ in Zufallsreihenfolge **do**
  if $F_{\text{conflict}}(p) \neq \emptyset$ **then**
    lösche alle Facetten $F_{\text{conflict}}(p)$ aus C
    $\mathcal{L} \leftarrow$ Horizont-Kanten von p
    foreach $e \in \mathcal{L}$ **do**
      $f \leftarrow$ neue Facette zu (e, p) in C; $v_f \leftarrow$ neuer Knoten für f in G
      $(f_1, f_2) \leftarrow$ Facetten vorher inzident zu e in C
      $P(e) \leftarrow P_{\text{conflict}}(f_1) \cup P_{\text{conflict}}(f_2)$
      foreach $p \in P(e)$ **do**
        if f sichtbar von p **then** füge Kante (p, f) zu G hinzu
    lösche $\{p\} \cup F_{\text{conflict}}(p)$ aus G
return C



Laufzeit?

Randomisierte Analyse

Laufzeit hängt insbesondere von # erzeugter Facetten ab

Randomisierte Analyse

Laufzeit hängt insbesondere von # erzeugter Facetten ab

Lemma 1: Erwartete Anzahl erzeugter Facetten $\leq 6n - 20$.

Randomisierte Analyse

Laufzeit hängt insbesondere von # erzeugter Facetten ab

Lemma 1: Erwartete Anzahl erzeugter Facetten $\leq 6n - 20$.

Beweis: $\rightarrow$ Rückwärtsanalyse

Randomisierte Analyse

Laufzeit hängt insbesondere von # erzeugter Facetten ab

Lemma 1: Erwartete Anzahl erzeugter Facetten $\leq 6n - 20$.

Beweis: $\rightarrow$ Rückwärtsanalyse

Satz 2: Die konvexe Hülle einer Menge von n Punkten in $\mathbb{R}^3$ kann in erwartet $O(n \log n)$ Zeit berechnet werden.

Randomisierte Analyse

Laufzeit hängt insbesondere von # erzeugter Facetten ab

Lemma 1: Erwartete Anzahl erzeugter Facetten $\leq 6n - 20$.

Beweis: $\rightarrow$ Rückwärtsanalyse

Satz 2: Die konvexe Hülle einer Menge von n Punkten in $\mathbb{R}^3$ kann in erwartet $O(n \log n)$ Zeit berechnet werden.

Beweis:

$P' \leftarrow \{p_1, p_2, p_3, p_4\} \subseteq P$ (nicht koplan.)
$C \leftarrow \mathrm{CH}(P')$; initialisiere Konfliktgraph G
foreach $p \in P \setminus P'$ in Zufallsreihenfolge **do**
  if $F_{\text{conflict}}(p) \neq \emptyset$ **then**
    lösche alle Facetten $F_{\text{conflict}}(p)$ aus C
    $\mathcal{L} \leftarrow$ Horizont-Kanten von p
    foreach $e \in \mathcal{L}$ **do**
      $f \leftarrow$ neue Facette zu (e, p) in C; $v_f \leftarrow$ neuer Knoten für f in G
      $(f_1, f_2) \leftarrow$ Facetten vorher inzident zu e in C
      $P(e) \leftarrow P_{\text{conflict}}(f_1) \cup P_{\text{conflict}}(f_2)$
      foreach $p \in P(e)$ **do**
        if f sichtbar von p **then** füge Kante (p, f) zu G hinzu
    lösche $\{p\} \cup F_{\text{conflict}}(p)$ aus G
return C

Randomisierte Analyse

Laufzeit hängt insbesondere von # erzeugter Facetten ab

Lemma 1: Erwartete Anzahl erzeugter Facetten $\leq 6n - 20$.

Beweis: $\rightarrow$ Rückwärtsanalyse

Satz 2: Die konvexe Hülle einer Menge von n Punkten in $\mathbb{R}^3$ kann in erwartet $O(n \log n)$ Zeit berechnet werden.

Beweis:

$P' \leftarrow \{p_1, p_2, p_3, p_4\} \subseteq P$ (nicht koplan.)
$C \leftarrow \mathrm{CH}(P')$; initialisiere Konfliktgraph G
foreach $p \in P \setminus P'$ in Zufallsreihenfolge **do**
  if $F_{\text{conflict}}(p) \neq \emptyset$ **then**
    lösche alle Facetten $F_{\text{conflict}}(p)$ aus C
    $\mathcal{L} \leftarrow$ Horizont-Kanten von p
    foreach $e \in \mathcal{L}$ **do**
      $f \leftarrow$ neue Facette zu (e, p) in C; $v_f \leftarrow$ neuer Knoten für f in G
      $(f_1, f_2) \leftarrow$ Facetten vorher inzident zu e in C
      $P(e) \leftarrow P_{\text{conflict}}(f_1) \cup P_{\text{conflict}}(f_2)$
      foreach $p \in P(e)$ **do**
        if f sichtbar von p **then** füge Kante (p, f) zu G hinzu
    lösche $\{p\} \cup F_{\text{conflict}}(p)$ aus G
return C

Randomisierte Analyse

Laufzeit hängt insbesondere von # erzeugter Facetten ab

Lemma 1: Erwartete Anzahl erzeugter Facetten $\leq 6n - 20$.

Beweis: $\rightarrow$ Rückwärtsanalyse

Satz 2: Die konvexe Hülle einer Menge von n Punkten in $\mathbb{R}^3$ kann in erwartet $O(n \log n)$ Zeit berechnet werden.

Beweis:

$O(n)$

$P' \leftarrow \{p_1, p_2, p_3, p_4\} \subseteq P$ (nicht koplan.)
$C \leftarrow \mathrm{CH}(P')$; initialisiere Konfliktgraph G
foreach $p \in P \setminus P'$ in Zufallsreihenfolge **do**
  if $F_{\text{conflict}}(p) \neq \emptyset$ **then**
    lösche alle Facetten $F_{\text{conflict}}(p)$ aus C
    $\mathcal{L} \leftarrow$ Horizont-Kanten von p
    foreach $e \in \mathcal{L}$ **do**
      $f \leftarrow$ neue Facette zu (e, p) in C; $v_f \leftarrow$ neuer Knoten für f in G
      $(f_1, f_2) \leftarrow$ Facetten vorher inzident zu e in C
      $P(e) \leftarrow P_{\text{conflict}}(f_1) \cup P_{\text{conflict}}(f_2)$
      foreach $p \in P(e)$ **do**
        if f sichtbar von p **then** füge Kante (p, f) zu G hinzu
    lösche $\{p\} \cup F_{\text{conflict}}(p)$ aus G
return C

Randomisierte Analyse

Laufzeit hängt insbesondere von # erzeugter Facetten ab

Lemma 1: Erwartete Anzahl erzeugter Facetten $\leq 6n - 20$.

Beweis: $\rightarrow$ Rückwärtsanalyse

Satz 2: Die konvexe Hülle einer Menge von n Punkten in $\mathbb{R}^3$ kann in erwartet $O(n \log n)$ Zeit berechnet werden.

Beweis:

$O(n)$

$P' \leftarrow \{p_1, p_2, p_3, p_4\} \subseteq P$ (nicht koplan.)
$C \leftarrow \mathrm{CH}(P')$; initialisiere Konfliktgraph G
foreach $p \in P \setminus P'$ in Zufallsreihenfolge **do**
  if $F_{\text{conflict}}(p) \neq \emptyset$ **then**
    lösche alle Facetten $F_{\text{conflict}}(p)$ aus C
    $\mathcal{L} \leftarrow$ Horizont-Kanten von p
    foreach $e \in \mathcal{L}$ **do**
      $f \leftarrow$ neue Facette zu (e, p) in C; $v_f \leftarrow$ neuer Knoten für f in G
      $(f_1, f_2) \leftarrow$ Facetten vorher inzident zu e in C
      $P(e) \leftarrow P_{\text{conflict}}(f_1) \cup P_{\text{conflict}}(f_2)$
      foreach $p \in P(e)$ **do**
        if f sichtbar von p **then** füge Kante (p, f) zu G hinzu
    lösche $\{p\} \cup F_{\text{conflict}}(p)$ aus G
return C

(Lemma 1) $O(n)$

Randomisierte Analyse

Laufzeit hängt insbesondere von # erzeugter Facetten ab

Lemma 1: Erwartete Anzahl erzeugter Facetten $\leq 6n - 20$.

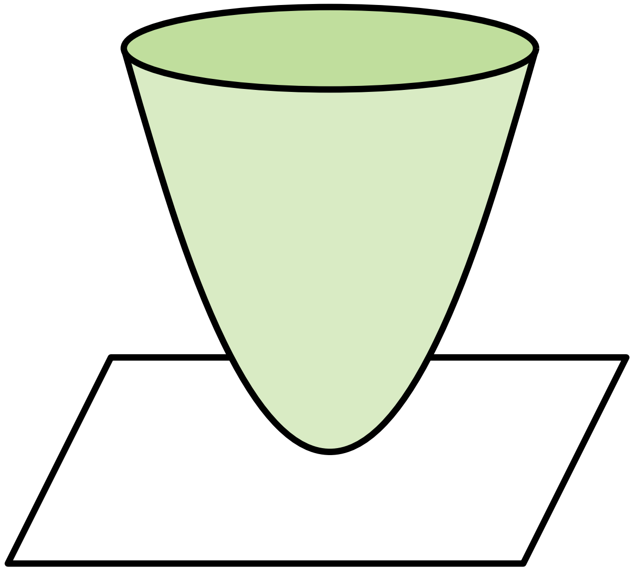
Beweis: $\rightarrow$ Rückwärtsanalyse

Satz 2: Die konvexe Hülle einer Menge von n Punkten in $\mathbb{R}^3$ kann in erwartet $O(n \log n)$ Zeit berechnet werden.

Beweis:

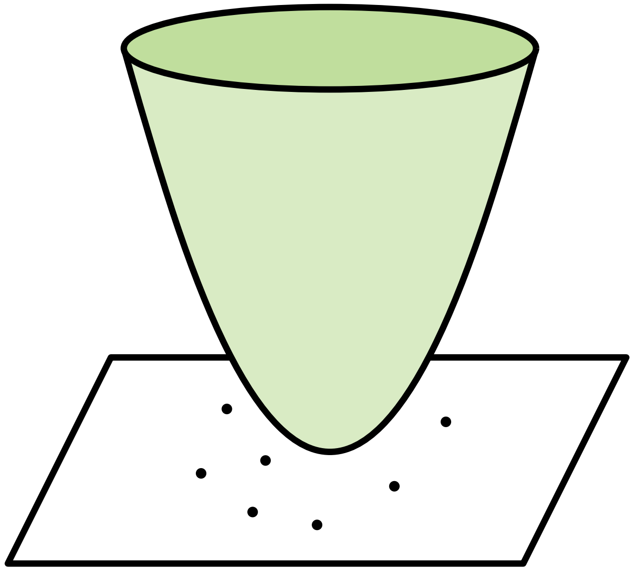
	$O(n)$	$P' \leftarrow \{p_1, p_2, p_3, p_4\} \subseteq P$ (nicht koplan.) $C \leftarrow \mathrm{CH}(P')$; initialisiere Konfliktgraph G
		foreach $p \in P \setminus P'$ in Zufallsreihenfolge **do**
(Lemma 1)	$O(n)$	**if** $F_{\text{conflict}}(p) \neq \emptyset$ **then** lösche alle Facetten $F_{\text{conflict}}(p)$ aus C $\mathcal{L} \leftarrow$ Horizont-Kanten von p **foreach** $e \in \mathcal{L}$ **do** $f \leftarrow$ neue Facette zu (e, p) in C; $v_f \leftarrow$ neuer Knoten für f in G $(f_1, f_2) \leftarrow$ Facetten vorher inzident zu e in C
([BCKO09] Kap. 9.5, 11.3)	$O(n \log n)$	$P(e) \leftarrow P_{\text{conflict}}(f_1) \cup P_{\text{conflict}}(f_2)$ **foreach** $p \in P(e)$ **do** **if** f sichtbar von p **then** füge Kante (p, f) zu G hinzu lösche $\{p\} \cup F_{\text{conflict}}(p)$ aus G
		return C

Konvexe Hülle & Delaunay Triangulierung



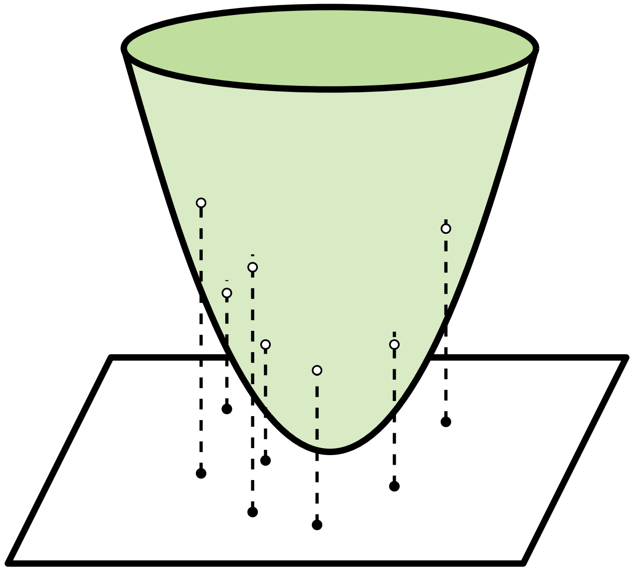
Es gibt einen erstaunlichen Bezug zw.
2D Delaunay Triangulierung und
(unterer) 3D konvexer Hülle
über den Paraboloid $B\colon z = x^2 + y^2$:

Konvexe Hülle & Delaunay Triangulierung



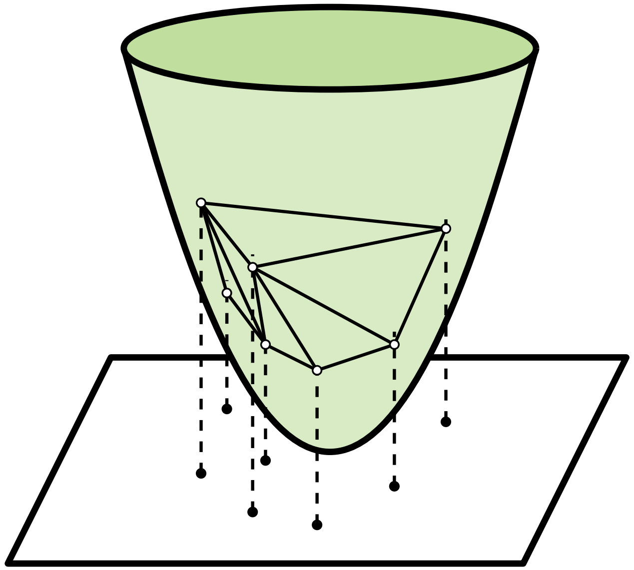
Es gibt einen erstaunlichen Bezug zw.
2D Delaunay Triangulierung und
(unterer) 3D konvexer Hülle
über den Paraboloid $B\colon z = x^2 + y^2$:

Konvexe Hülle & Delaunay Triangulierung



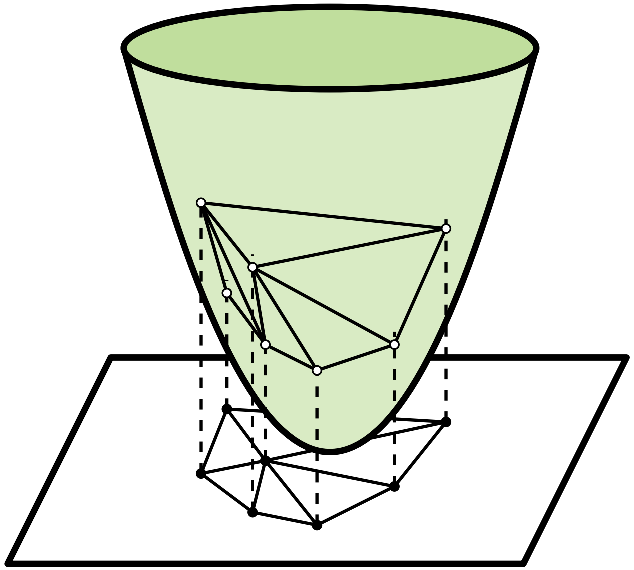
Es gibt einen erstaunlichen Bezug zw. 2D Delaunay Triangulierung und (unterer) 3D konvexer Hülle über den Paraboloid $B\colon z = x^2 + y^2$:

Konvexe Hülle & Delaunay Triangulierung



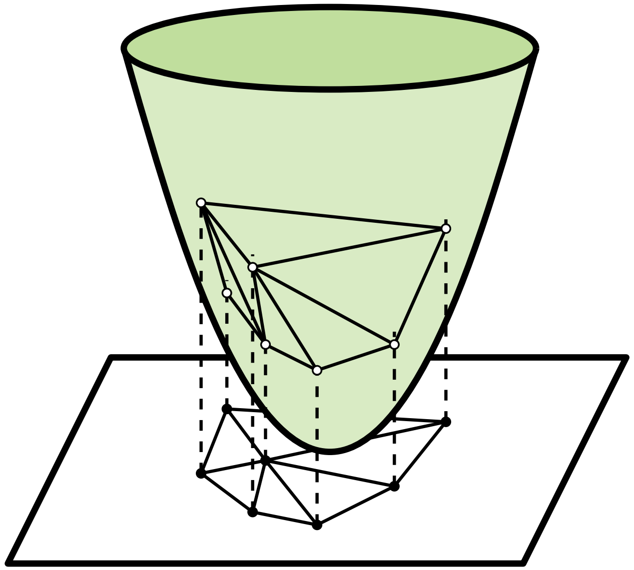
Es gibt einen erstaunlichen Bezug zw. 2D Delaunay Triangulierung und (unterer) 3D konvexer Hülle über den Paraboloid $B\colon z = x^2 + y^2$:

Konvexe Hülle & Delaunay Triangulierung



Es gibt einen erstaunlichen Bezug zw. 2D Delaunay Triangulierung und (unterer) 3D konvexer Hülle über den Paraboloid $B\colon z = x^2 + y^2$:

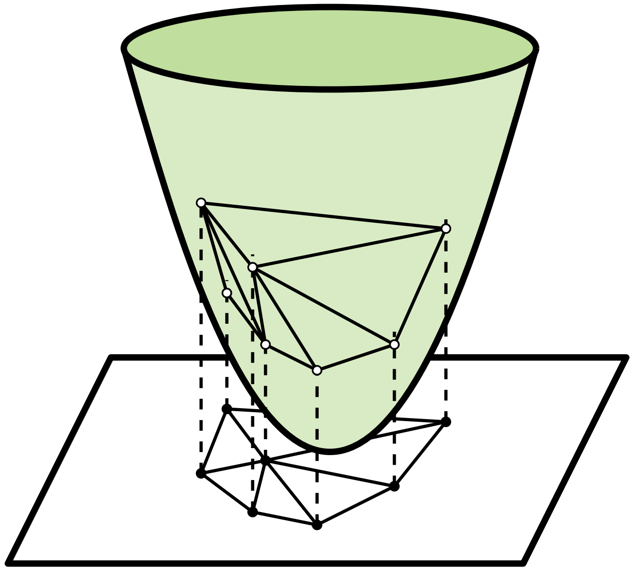
Konvexe Hülle & Delaunay Triangulierung



Es gibt einen erstaunlichen Bezug zw. 2D Delaunay Triangulierung und (unterer) 3D konvexer Hülle über den Paraboloid $B\colon z = x^2 + y^2$:

- projiziere P auf $B \to$ Punktmenge P'
- berechne $CH(P')$
- projiziere unteren Teil von $CH(P')$ zurück
- liefert Delaunay Triangulierung von P

Konvexe Hülle & Delaunay Triangulierung

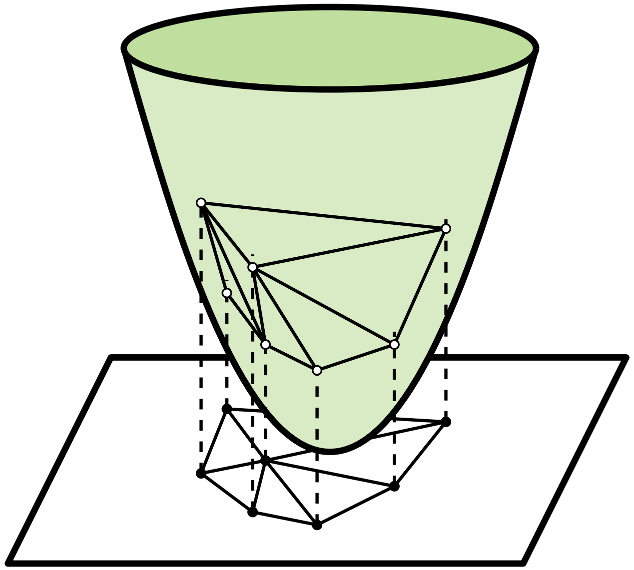


Es gibt einen erstaunlichen Bezug zw. 2D Delaunay Triangulierung und (unterer) 3D konvexer Hülle über den Paraboloid $B\colon z = x^2 + y^2$:

- projiziere P auf $B \to$ Punktmenge P'
- berechne $CH(P')$
- projiziere unteren Teil von $CH(P')$ zurück
- liefert Delaunay Triangulierung von P

Delaunay-Bedingung: Punkte p, q, r bilden Dreieck
$\Leftrightarrow$ Umkreis $C(p, q, r)$ ist leer

Konvexe Hülle & Delaunay Triangulierung



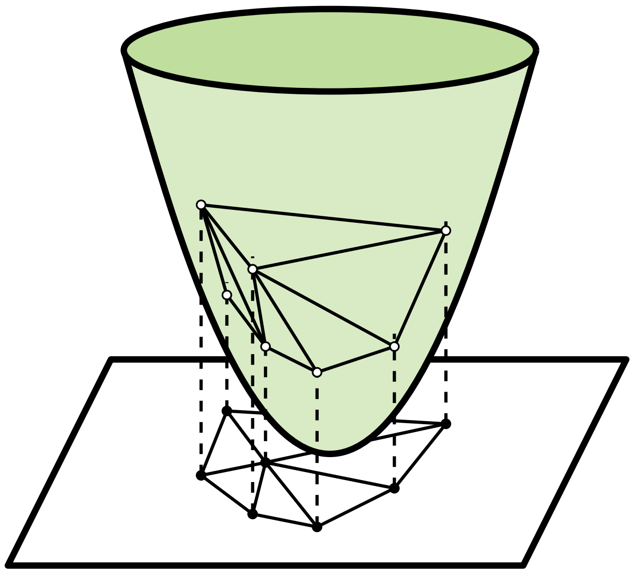
Es gibt einen erstaunlichen Bezug zw.
2D Delaunay Triangulierung und
(unterer) 3D konvexer Hülle
über den Paraboloid $B\colon z = x^2 + y^2$:

- projiziere P auf $B \rightarrow$ Punktmenge P'
- berechne $CH(P')$
- projiziere unteren Teil von $CH(P')$ zurück
- liefert Delaunay Triangulierung von P

Delaunay-Bedingung: Punkte p, q, r bilden Dreieck
$\Leftrightarrow$ Umkreis $C(p, q, r)$ ist leer

CH-Bedingung: Punkte p', q', r' bilden Facette
$\Leftrightarrow$ P' auf einer Seite der Ebene $h(p', q', r')$

Konvexe Hülle & Delaunay Triangulierung



Es gibt einen erstaunlichen Bezug zw. 2D Delaunay Triangulierung und (unterer) 3D konvexer Hülle über den Paraboloid $B\colon z = x^2 + y^2$:

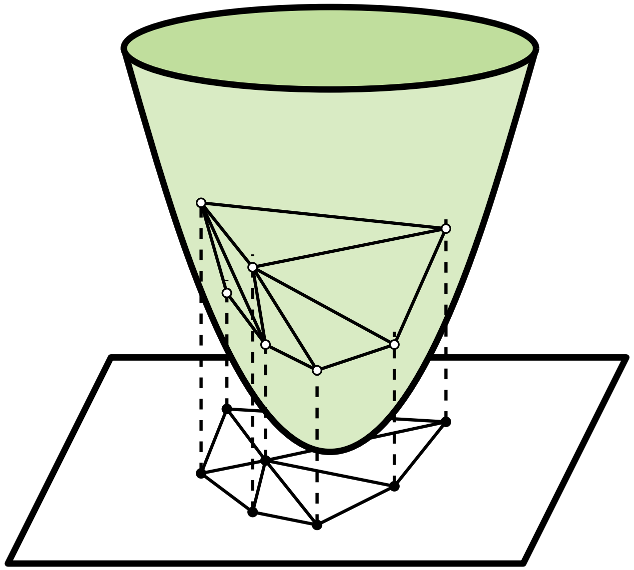
- projiziere P auf $B \rightarrow$ Punktmenge P'
- berechne $CH(P')$
- projiziere unteren Teil von $CH(P')$ zurück
- liefert Delaunay Triangulierung von P

Delaunay-Bedingung: Punkte p, q, r bilden Dreieck
$\Leftrightarrow$ Umkreis $C(p, q, r)$ ist leer

CH-Bedingung: Punkte p', q', r' bilden Facette
$\Leftrightarrow$ P' auf einer Seite der Ebene $h(p', q', r')$

Lemma 2: Für vier Punkte $p, q, r, s \in \mathbb{R}^2$ und ihre Projektionen p', q', r', s' auf B gilt: s liegt im Umkreis $C(p, q, r)$
$\Leftrightarrow$ s' liegt unterhalb Ebene $h(p', q', r')$

Konvexe Hülle & Delaunay Triangulierung



Satz 3: Für eine Punktmenge $P \subseteq \mathbb{R}^2$ in allg. Lage und $p, q, r \in P$ ist Dreieck Δpqr Delaunay-Dreieck gdw. $\Delta p'q'r'$ Facette der unteren konvexen Hülle von P'.

Delaunay-Bedingung: Punkte p, q, r bilden Dreieck
$\Leftrightarrow$ Umkreis $C(p, q, r)$ ist leer

CH-Bedingung: Punkte p', q', r' bilden Facette
$\Leftrightarrow$ P' auf einer Seite der Ebene $h(p', q', r')$

Lemma 2: Für vier Punkte $p, q, r, s \in \mathbb{R}^2$ und ihre Projektionen p', q', r', s' auf B gilt: s liegt im Umkreis $C(p, q, r)$
$\Leftrightarrow$ s' liegt unterhalb Ebene $h(p', q', r')$

Noch mehr Dualität

$\mathbb{R}^2$: $\quad \mathsf{Vor}(P) \underset{\mathsf{VL9}}{\overset{\sim}{\longleftrightarrow}} \mathsf{DT}(P)$

Noch mehr Dualität

$$\mathbb{R}^2: \quad \mathrm{Vor}(P) \xleftrightarrow[\text{VL9}]{\sim} \mathrm{DT}(P)$$

$$\Big\updownarrow \text{ Projektion}$$

$$\mathbb{R}^3: \quad \mathrm{LCH}(P')$$

Noch mehr Dualität

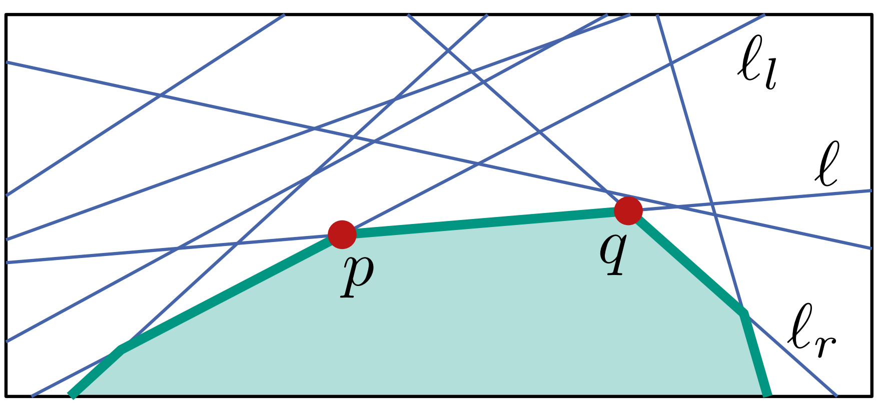
$\mathbb{R}^2$: $\quad \mathrm{Vor}(P) \xleftrightarrow[\text{VL9}]{\sim} \mathrm{DT}(P)$

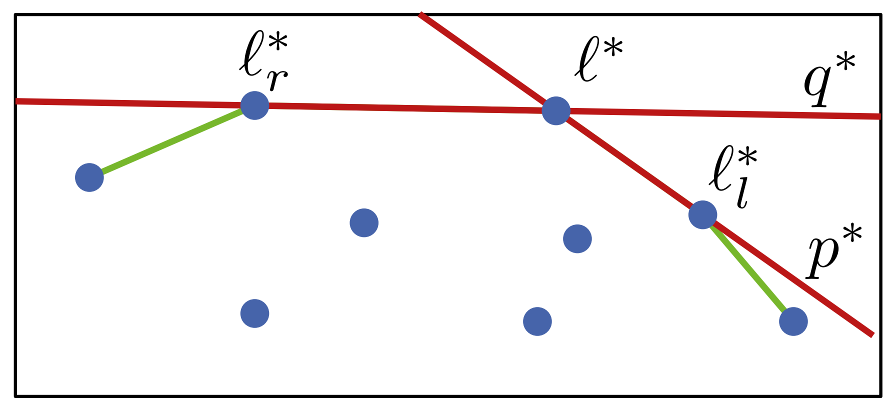
$\updownarrow$ Projektion

$\mathbb{R}^3$: $\quad \mathrm{UE}(P') \xleftrightarrow[\text{VL10}]{\sim} \mathrm{LCH}(P')$

upper envelope

(gilt auch in $\mathbb{R}^3$)
$\rightarrow$ Übung





Noch mehr Dualität

$$
\begin{array}{llcl}
\mathbb{R}^2: & \text{Vor}(P) & \xleftrightarrow[\text{VL9}]{\sim} & \text{DT}(P) \\
 & ? & & \Big\updownarrow \text{Projektion} \\
\mathbb{R}^3: & \text{UE}(P') & \xleftrightarrow[\text{VL10}]{\sim} & \text{LCH}(P')
\end{array}
$$

(gilt auch in $\mathbb{R}^3$)

$\rightarrow$ Übung

Noch mehr Dualität

$$\begin{tikzcd} \mathbb{R}^2: & \mathrm{Vor}(P) \arrow[r, leftrightarrow, "\sim", "\text{VL9}"'] \arrow[d, phantom, "?"] & \mathrm{DT}(P) \arrow[d, leftrightarrow, "\text{Projektion}"] \\ \mathbb{R}^3: & \mathrm{UE}(P') \arrow[r, leftrightarrow, "\sim", "\text{VL10}"'] & \mathrm{LCH}(P') \end{tikzcd}$$

(gilt auch in $\mathbb{R}^3$)
$\rightarrow$ Übung

Satz 4: Für eine Punktmenge $P \subseteq \mathbb{R}^2$ sei $H(P')$ die Menge der Tangentialebenen des Paraboloids B an den projizierten Punkten P'. Dann ist die Projektion von $\mathrm{UE}(H(P'))$ gerade $\mathrm{Vor}(P)$.

Noch mehr Dualität

$$\begin{tikzcd} \mathbb{R}^2: & \mathrm{Vor}(P) \arrow[r, leftrightarrow, "\sim", "\text{VL9}"'] \arrow[d, leftrightarrow, "\text{Projektion}"'] & \mathrm{DT}(P) \arrow[d, leftrightarrow, "\text{Projektion}"] \\ \mathbb{R}^3: & \mathrm{UE}(P') \arrow[r, leftrightarrow, "\sim", "\text{VL10}"'] & \mathrm{LCH}(P') \end{tikzcd}$$

(gilt auch in $\mathbb{R}^3$)
$\rightarrow$ Übung

Satz 4: Für eine Punktmenge $P \subseteq \mathbb{R}^2$ sei $H(P')$ die Menge der Tangentialebenen des Paraboloids B an den projizierten Punkten P'. Dann ist die Projektion von $\mathrm{UE}(H(P'))$ gerade $\mathrm{Vor}(P)$.

Diskussion

Optimale Laufzeit?

Diskussion

Optimale Laufzeit?

Divide-and-Conquer Algorithmus mit Laufzeit $O(n \log n)$ oder gift-wrapping mit Laufzeit $O(n \cdot h)$ ($h = \#$ Facetten der konvexen Hülle)

Diskussion

Optimale Laufzeit?

Divide-and-Conquer Algorithmus mit Laufzeit $O(n \log n)$ oder gift-wrapping mit Laufzeit $O(n \cdot h)$ ($h = \#$ Facetten der konvexen Hülle)

Was ist mit Dimensionen > 3 ?

Diskussion

Optimale Laufzeit?

Divide-and-Conquer Algorithmus mit Laufzeit $O(n \log n)$ oder gift-wrapping mit Laufzeit $O(n \cdot h)$ ($h = \#$ Facetten der konvexen Hülle)

Was ist mit Dimensionen > 3 ?

Upper Bound Theorem: Komplexität der konvexen Hülle einer Punktmenge bestehend aus n Punkten in $\mathbb{R}^d$ ist $\Theta(n^{\lfloor d/2 \rfloor})$ ist.
Der hier vorgestellte inkrementelle Algorithmus kann für höhere Dimensionen angepasst werden.
Der beste Ausgabe-sensitive Algorithmus für die Berechnung konvexer Hüllen in $\mathbb{R}^d$ hat eine Laufzeit von:

$$O(n \log h + (nh)^{1-1/(\lfloor d/2 \rfloor + 1)} \log^{O(1)} n)$$

Informationen zur Prüfung

Prüfungstermine:

- 6. August ab 9:00 Uhr
- 8. Oktober ab 9:00 Uhr
- bei Bedarf noch weiterer Termin im WS 2014/15

Anmeldung per Mail an lilian.beckert@kit.edu

Informationen zur Prüfung

Prüfungstermine:

- 6. August ab 9:00 Uhr
- 8. Oktober ab 9:00 Uhr
- bei Bedarf noch weiterer Termin im WS 2014/15

Anmeldung per Mail an lilian.beckert@kit.edu

Projektnoten:

- bitte beim Betreuer erfragen

Informationen zur Prüfung – Inhalt

- grundlegende Problemdefinitionen

Informationen zur Prüfung – Inhalt

- grundlegende Problemdefinitionen
- Aufbau und Queries von Datenstrukturen

Informationen zur Prüfung – Inhalt

- grundlegende Problemdefinitionen
- Aufbau und Queries von Datenstrukturen
- Zusammenhänge zwischen verschiedenen Datenstrukturen

Informationen zur Prüfung – Inhalt

- grundlegende Problemdefinitionen
- Aufbau und Queries von Datenstrukturen
- Zusammenhänge zwischen verschiedenen Datenstrukturen
- Eigenschaften und Ablauf der Algorithmen

Informationen zur Prüfung – Inhalt

- grundlegende Problemdefinitionen
- Aufbau und Queries von Datenstrukturen
- Zusammenhänge zwischen verschiedenen Datenstrukturen
- Eigenschaften und Ablauf der Algorithmen
- zentrale Beweisideen

Informationen zur Prüfung – Inhalt

- grundlegende Problemdefinitionen
- Aufbau und Queries von Datenstrukturen
- Zusammenhänge zwischen verschiedenen Datenstrukturen
- Eigenschaften und Ablauf der Algorithmen
- zentrale Beweisideen
- Einsatz der Algorithmen und Datenstrukturen in Anwendungen

Informationen zur Prüfung – Inhalt

- grundlegende Problemdefinitionen
- Aufbau und Queries von Datenstrukturen
- Zusammenhänge zwischen verschiedenen Datenstrukturen
- Eigenschaften und Ablauf der Algorithmen
- zentrale Beweisideen
- Einsatz der Algorithmen und Datenstrukturen in Anwendungen
- bei 5LP: Stoff der Übungen

Informationen zur Prüfung – Inhalt

- grundlegende Problemdefinitionen
- Aufbau und Queries von Datenstrukturen
- Zusammenhänge zwischen verschiedenen Datenstrukturen
- Eigenschaften und Ablauf der Algorithmen
- zentrale Beweisideen
- Einsatz der Algorithmen und Datenstrukturen in Anwendungen
- bei 5LP: Stoff der Übungen

Wie vorbereiten?

- Vorlesungsfolien **und** Tafelbeweise aus angegebener Literatur (v.a. [BCKO08] und Skript [M12])
- Übungsaufgaben
- nur *ergänzend*: weiteres Material aus der Literatur

Informationen zur Prüfung – Inhalt

- grundlegende Problemdefinitionen
- Aufbau und Queries von Datenstrukturen
- Zusammenhänge zwischen verschiedenen Datenstrukturen
- Eigenschaften und Ablauf der Algorithmen
- zentrale
- Einsatz
  Anwend

Weitere Fragen?

- bei 5LP: Stoff der Übungen

Wie vorbereiten?

- Vorlesungsfolien **und** Tafelbeweise aus angegebener Literatur (v.a. [BCKO08] und Skript [M12])
- Übungsaufgaben
- nur *ergänzend*: weiteres Material aus der Literatur

Ausblick (WS 2014/15)

- Bei Interesse an einer Masterarbeit einfach melden. Wir haben regelmäßig spannende theoretische und praktische Themen aus unseren Forschungsbereichen.
- Vorlesung *Algorithmen zur Visualisierung von Graphen*
- Vorlesung *Algorithmische Graphentheorie*
- Seminar *Algorithmentechnik*
- Praktikum *Routenplanung*

weitere Infos demnächst unter `i11www.iti.kit.edu`

Ausblick (WS 2014/15)

- Bei Interesse an einer Masterarbeit einfach melden. Wir haben regelmäßig spannende theoretische und praktische Themen aus unseren Forschungsbereichen.
- Vorlesung *Algorithmen zur Visualisierung von Graphen*
- Vorlesung *Algorithmische Graphentheorie*
- Seminar *Algorithmentechnik*
- Praktikum *Routenplanung*

weitere Infos demnächst unter `i11www.iti.kit.edu`

Viel Erfolg für die Prüfung!