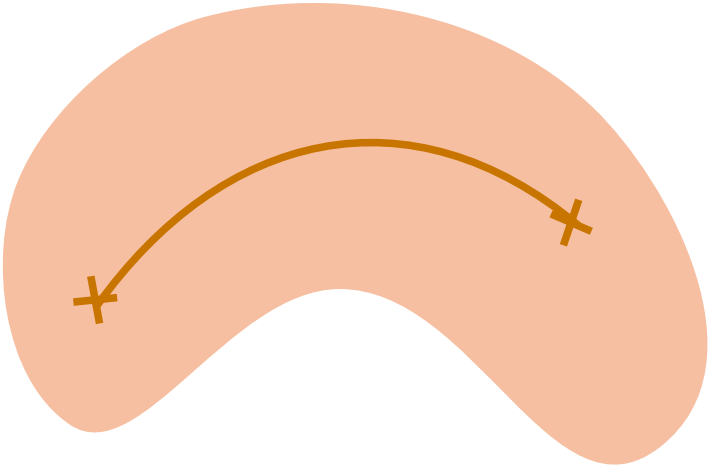
Komponenten u. Konturen

Im Folgenden suchen wir (zusammen-hängende) Komponenten in Binärbildern (nur einem Bit (0/1) pro Pixel).

Zusammenhängend bedeutet, dass es von jedem Pixel zu jedem anderen einen Weg bzgl. der 8er-Nachbarschaft gibt. Man spricht auch von wegzusammenhängend, weil es zwischen zwei bel. Punkten immer einen Weg gibt.

Objekte können durchaus aus mehreren getrennten Komponenten bestehen.

- Objektfarbe
- Hintergrundfarbe



Beispiel einer zusammenhängenden Komponente (erdnussförmiger Bereich)

© Thomas Haenselmann, Prof. Dr. rer. nat. habil.

Komponenten u. Konturen

Die Bettische Zahl

Um ein Objekt S grob zu beschreiben ohne auf seine genaue Form einzugehen nutzt man gerne die aus der (math.) Topologie bekannte **Bettische Zahl** Bk. Sie macht eine Aussage über die Anzahl der k-dimensionalen Flächen eines Raumes.

Glücklicherweise sind Bilder nur 2-dimensionale Räume, so dass uns nur B0 und B1 interessieren. Ihre Bedeutung ist wie folgt:

B0(S): Bezeichnet die Anzahl der Komponenten eines Objekts S

B1(S): Bezeichnet die Anzahl der Löcher des Objekts S.

Beispiel rechts: B0(S) = 2 / B1(S) = 2

Das Objekt S besteht aus zwei Komponenten und zwei Löchern.

Komponenten u. Konturen

Die Bettische Zahl

Wir werden gleich Verfahren kennenlernen mit denen Komponenten gefunden werden können, jedoch nicht Löcher. Warum keine Löcher?

Weil man das Objekt S einfach zu C(S) invertieren kann, also Objekt- und Hintergrundpixel vertauschen. Dadurch werden automatisch die Löcher zu Komponenten.

Es gilt: B1 = B0(C(S)) – 1 also hier

2 = 3 - 1

Bedeutung: Die Anzahl der Löcher entspricht der Anzahl der Komponenten des invertierten Bildes C(S) minus 1 (f. Hintergrund).

Die Differenz E(S)=B0(S)-B1(S)

wird als Eulersche Zahl bezeichnet.

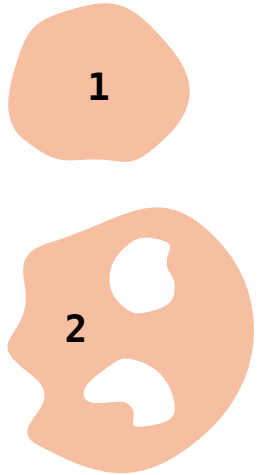
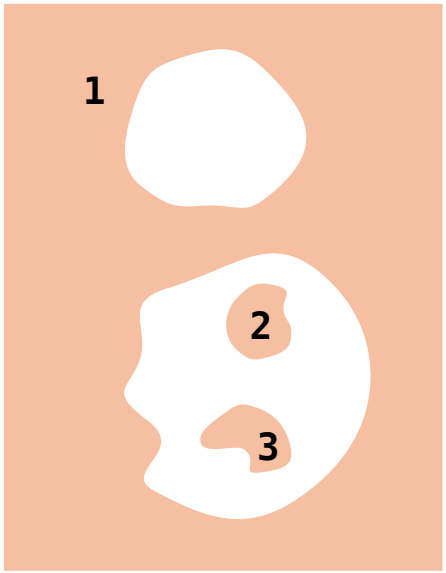


Bild des Objekts, bestehend aus zwei Komponenten.



Invertiertes Bild, bestehend aus drei Komponenten.

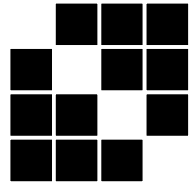
© Thomas Haenselmann, Prof. Dr. rer. nat. habil.

Praktikum
Komponenten

a) Wie viele Komponenten und Löcher hat die folgende Figur?



b) Wie viele Komponenten hat die Pixelfigur (rechts) und warum? Ist die Aufgabe so überhaupt eindeutig definiert?



© Thomas Haenselmann, Prof. Dr. rer. nat. habil.

Komponenten u. Konturen

Verfahren zur Berechnung von Komponenten

Nachbarschaftsverfahren: (Auch bekannt als Flood-Fill) Wir suchen einen bel. Pixel einer Komponente. Ausgehend von diesem breiten wir uns auf alle Nachbarn in einer 8er Umgebung aus, die auch Komponentenpixel sind. So „füllt“ sich die Komponente.

Konturensuche: Finde die Konturen durch ein Konturfolgeverfahren. Die Anzahl der Konturen entspricht B0(S).

Zeilenkoinzidenzverfahren*: Bild wird zeilenweise von links nach rechts abgetastet. Wenn links oder über mir schon mal eine Komponente war, gehört das betrachtete Pixel auch zur Komponente.

*Koinzidenz: Zeitliches/räumliches Zusammentreffen von Ereignissen

Regionensuche durch Füllalgorithmus

In welcher Reihenfolge werden die Pixel der Region besucht?

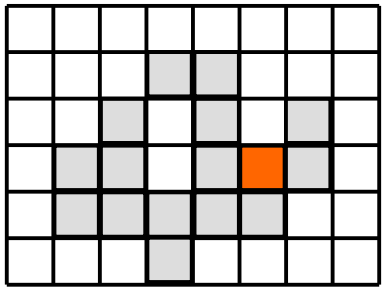
```
void FindObject(int x, int y)
{
  if(IsWhite(x, y)) return;

  MarkVisited(x, y);

  for(int dy = -1; dy < 2; dy++) {
    for(int dx = -1; dx < 2; dx++) {
      if(NeverVisited(x+dx, y+dy) == true)
        FindObject(x+dx, y+dy);
    }
  }

}

FindObject(6, 4);
```



© Thomas Haenselmann, Prof. Dr. rer. nat. habil.

Komponenten u. Konturen

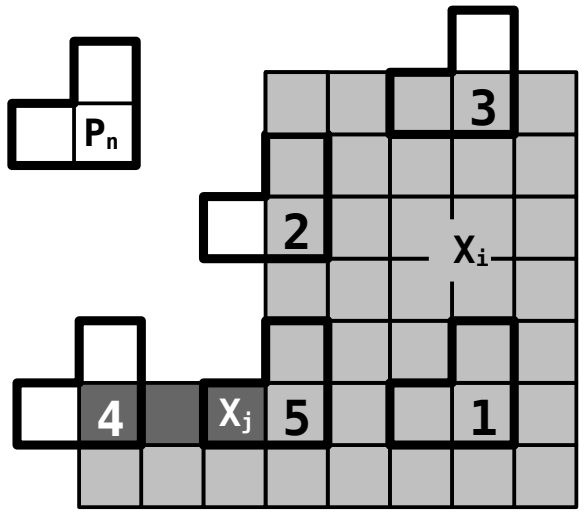
```
DO ab Zeile 2
  DO ab Spalte 2

    IF(Situation 1-3) THEN Xi=XiU{Pn}

    // neue Region :)
    IF(Situation 4) THEN Xj={Pn}; Z:=Z+1;

    // doch keine neue Region :(
    IF(Situation 5) THEN Xi=Xi U XjU{Pn};
                         Z:=Z-1;Xj={}

  END
 END
```



U: Wir vereinigen zwei Mengen
Z: Anzahl Komponenten
P_n: Aktuelles Pixel
X_i: Menge i zusammenh. Pixel

© Thomas Haenselmann, Prof. Dr. rer. nat. habil.

Zeilenkoinzidenzverfahren

Regionenpixel sind in der Darstellung grau, die weißen Pixel gelten als Hintergrund. Sie bearbeiten das Bild nach dem Zeilenkoinzidenzverfahren von der ersten bis zur letzten Zeile und innerhalb der Zeilen von links nach rechts.
Geben Sie die (x,y)-Koordinate an, falls

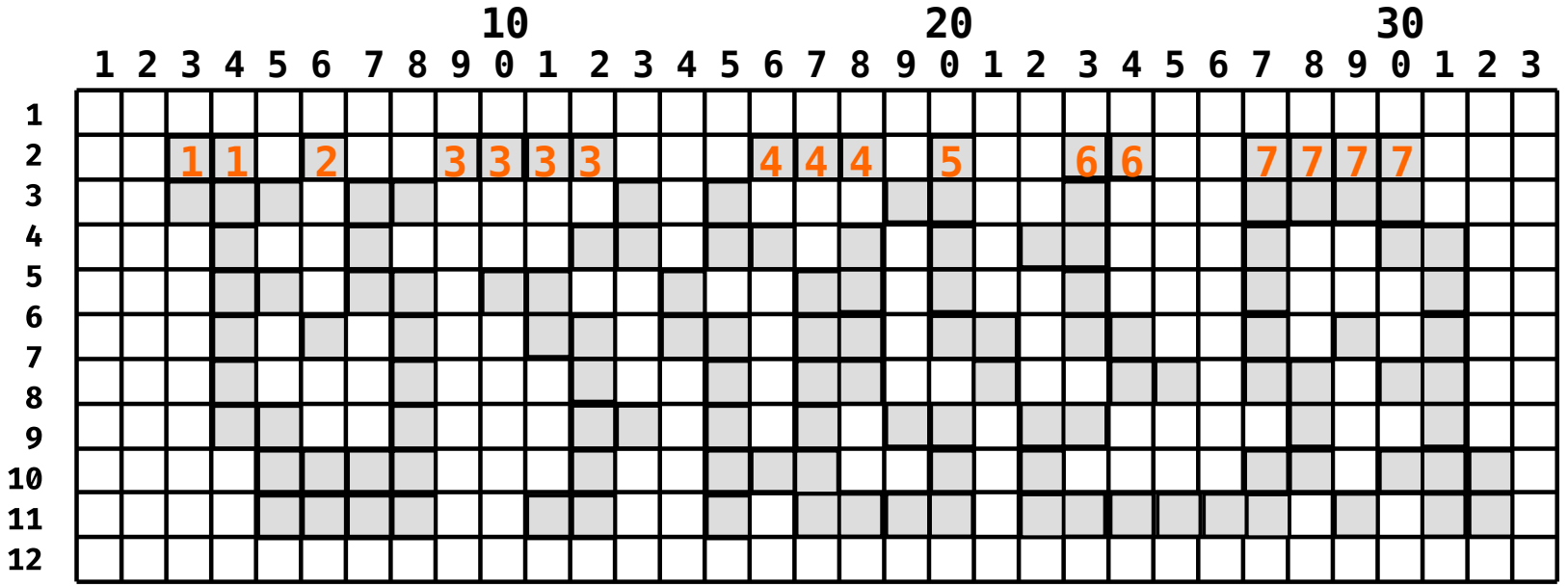
- eine neue Region gefunden wurde (Fall A) oder Sie
- eine zwei Regionen zu einer verschmelzen konnten (Fall B).

Schreiben Sie im letzteren Fall auf, welche Region (Nummer) mit welcher anderen Region verschmolzen wurde.

Tragen Sie die Regionenlabel nach Möglichkeit in die Grafik ein. Beachten Sie, dass dabei ein nachträgliches Verändern der Regionenlabel nicht zu vermeiden ist.

Schreiben Sie im letzteren Fall auf, welche Region (Nummer) mit welcher anderen Region verschmolzen wurde.
Beispiel:

```
( 3,2): Fall A, neue Region 1
( 6,2): Fall A, neue Region 2
( 9,2): Fall A, neue Region 3
(16,2): Fall A, neue Region 4
```



© Thomas Haenselmann, Prof. Dr. rer. nat. habil.

Komponenten u. Konturen

Vorüberlegung zu Abständen

Der Euklidische Abstand zwischen P und Q:

$$d_e(P,Q)=\sqrt{(P.x-Q.x)^2+(P.y-Q.y)^2}$$

Die max.-Metrik:

$$d\ (P,Q)=max(|P.x-Q.x|,|P.y-Q.y|)$$

Die Manhattan-/Häuserblock-Metrik:

$$d_4(P,Q)=|P.x-Q.x|+|P.y-Q.y|$$

Komponenten u. Konturen

Merkmale

Die folgende Auswahl verstehen sich als Baukasten von Merkmalen, die zur Lösung anwendungsspezifischer Probleme genutzt werden können.

Umfang U(S) einer Kontur

C(i): i-tes Freeman Codezeichen

Fläche

A(S): (Keine bekannte analytisch geschlossene Darstellung.)

Centroid (Schwerpunkt) $Cent_{x/y}(S)$

$$U(S)=\sum_{i=1}^{n} B(C(i))$$

$$B(C(i))=\begin{cases}1, falls\ C(i)\ gerade\\ \sqrt{2}, sonst\end{cases}$$

$$Cent_x(S)=1/n\sum_{i=1}^{n} x_i$$

$$Cent_y(S)=1/n\sum_{i=1}^{n} y_i$$

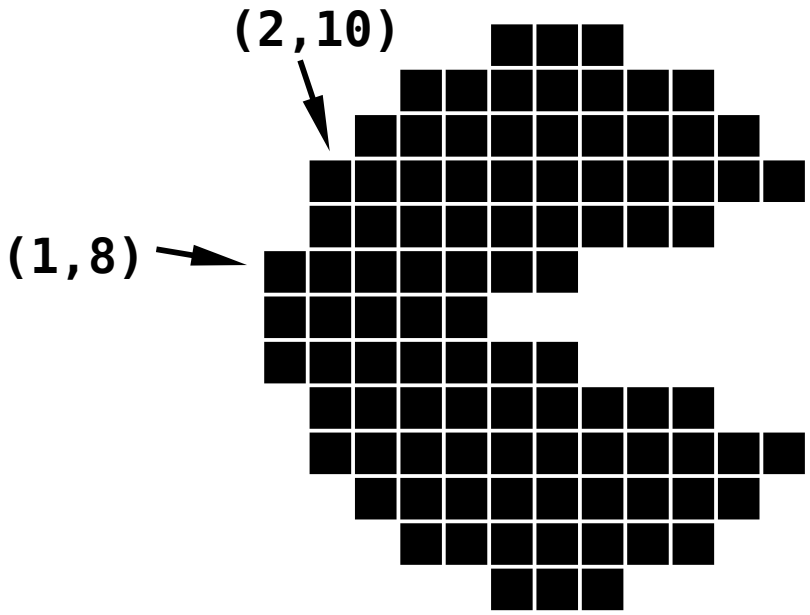
© Thomas Haenselmann, Prof. Dr. rer. nat. habil.

Praktikum
Komponentenmaße

a) Berechnen Sie den Schwerpunkt der folgenden Figur.
b) Ermitteln Sie nun das umschließende Achteck. Zur Erinnerung:

$$P_{min}(S) = min(x_i + y_i; 1 \leq i \leq n)$$
$$P_{max}(S) = max(x_i + y_i; 1 \leq i \leq n)$$
$$M_{min}(S) = min(x_i - y_i; 1 \leq i \leq n)$$
$$M_{max}(S) = max(x_i - y_i; 1 \leq i \leq n)$$

c) Wie groß sind Umfang und Flächeninhalt der Figur, wenn ein Pixel die Größe 1x1 hat?



$$U(S) = \sum_{i=1}^{n} B(C(i))$$
$$B(C(i)) = \begin{cases} 1, falls\ C(i)\ gerade \\ \sqrt{2}, sonst \end{cases}$$

© Thomas Haenselmann, Prof. Dr. rer. nat. habil.

Komponenten u. Konturen

Merkmale

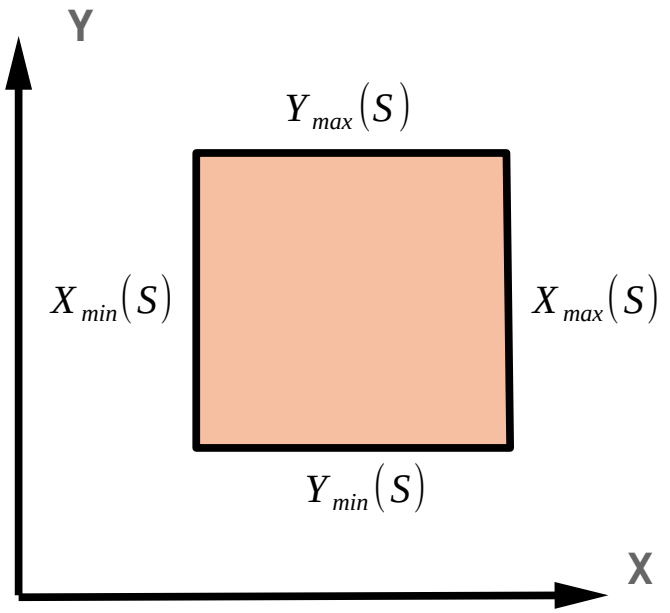
Minimal umschreibendes Rechteck:

$$X_{min}(S) = min(x_i; 1 \leq i \leq n)$$

$$X_{max}(S) = max(x_i; 1 \leq i \leq n)$$

$$Y_{min}(S) = min(y_i; 1 \leq i \leq n)$$

$$Y_{max}(S) = max(y_i; 1 \leq i \leq n)$$



© Thomas Haenselmann, Prof. Dr. rer. nat. habil.

Komponenten u. Konturen

Merkmale

Minimal umschreibendes Achteck:

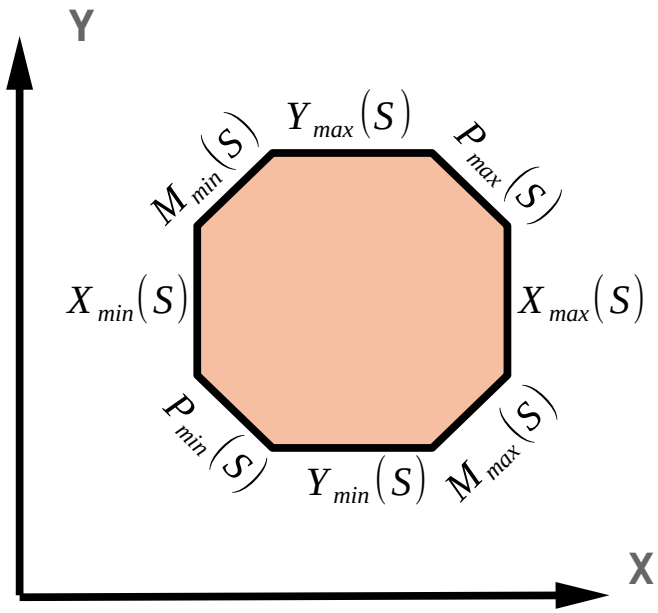
X_{min}, X_{max}, Y_{min}, Y_{max} ebenso wie zuvor, jedoch zusätzlich:

$$P_{min}(S) = min(x_i + y_i; 1 \le i \le n)$$

$$P_{max}(S) = max(x_i + y_i; 1 \le i \le n)$$

$$M_{min}(S) = min(x_i - y_i; 1 \le i \le n)$$

$$M_{max}(S) = max(x_i - y_i; 1 \le i \le n)$$



© Thomas Haenselmann, Prof. Dr. rer. nat. habil.

Komponenten u. Konturen

Merkmale

Folge der Radien Centroid → Kontur: R_i

Mittlere Radienlänge: R

Varianz der Radienlängen: V

Durchmesser von S: DIA(S)

$$R_i = d_e\left((x_i, y_i), (Cent_x, Cent_y)\right); \quad 1 \le i \le n$$

$$R = 1/n \sum_{i=1}^{n} R_i$$

$$V = 1/(n-1) \sum_{i=1}^{n} (R_i - R)^2$$

$$DIA(S) = max\left(d_e\left((x_i, y_i), (x_j, y_j)\right); \quad 1 \le i; j \le n\right)$$

© Thomas Haenselmann, Prof. Dr. rer. nat. habil.

Komponenten u. Konturen

Merkmale

Minimale Breite von S:
B(S) = Minimaler Abstand zweier paralleler Tangenten von S. (nur prozedurale Lösungen bekannt)

Rechteck (nicht achsenparallel) mit min. Fläche, das S umschreibt:
Zwei Parallelenpaare die S umschreiben und eine min. Fläche einnehmen.

Kleinster Kreis, der S umschreibt:
Definiert durch Mittelpunkt und Radius.

Komponenten u. Konturen

Abgeleitete Merkmale

Flächenverhältnisse zwischen Originalmenge, umschreibenden Acht- und Rechtecken, umschreibendem Kreis:

Abstände der Centroide (Schwerpunkte) der oben genannten Figuren.

Formfaktor: $F(S) = U(S)^2 / A(S)$

Verhältnis von Durchmesser und minimaler Breite von S uvm.

Komponenten u. Konturen

Konvexe Hülle

Das umschließende Achteck approximiert eine Form schon recht gut. Noch näher kommt man beliebigen Formen jedoch mit der konvexen Hülle.

Danach sind bessere Näherungen nur noch durch nicht-konvexe Überdeckungen möglich. Aus Gründen der schnellen Berechenbarkeit beschränkt man sich jedoch gern auf komplexe Hüllen.

Anwendungen: Computerspiele für einfache Kollisionserkennung / Robotik

© Thomas Haenselmann, Prof. Dr. rer. nat. habil.

Komponenten u. Konturen

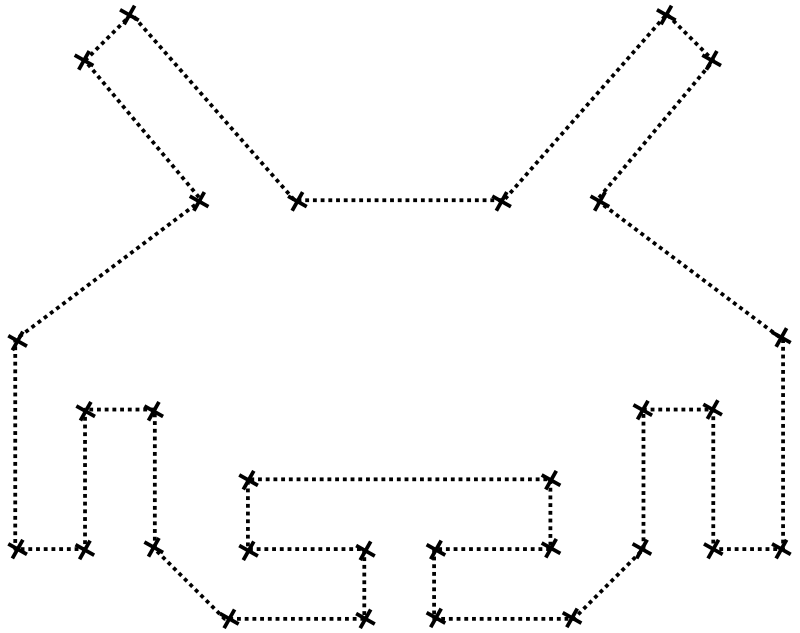
Konvexe Hülle

Vorbereitung

P0->P1 sei ein Strahl, ausgehend von Punkt P0 in Richtung P1. P0 muss garantiert auf der konvexen Hülle liegen.

Rotiere diesen Strahl gegen den Uhrzeigersinn. Dabei werden alle überstrichenen Punkte mit einer fortlaufenden Nummer versehen.

Es kann vorkommen, dass der Strahl gleich mehrere Punkte gleichzeitig trifft. In diesem Fall behalten wir nur den entferntesten Punkt und löschen alle näheren.



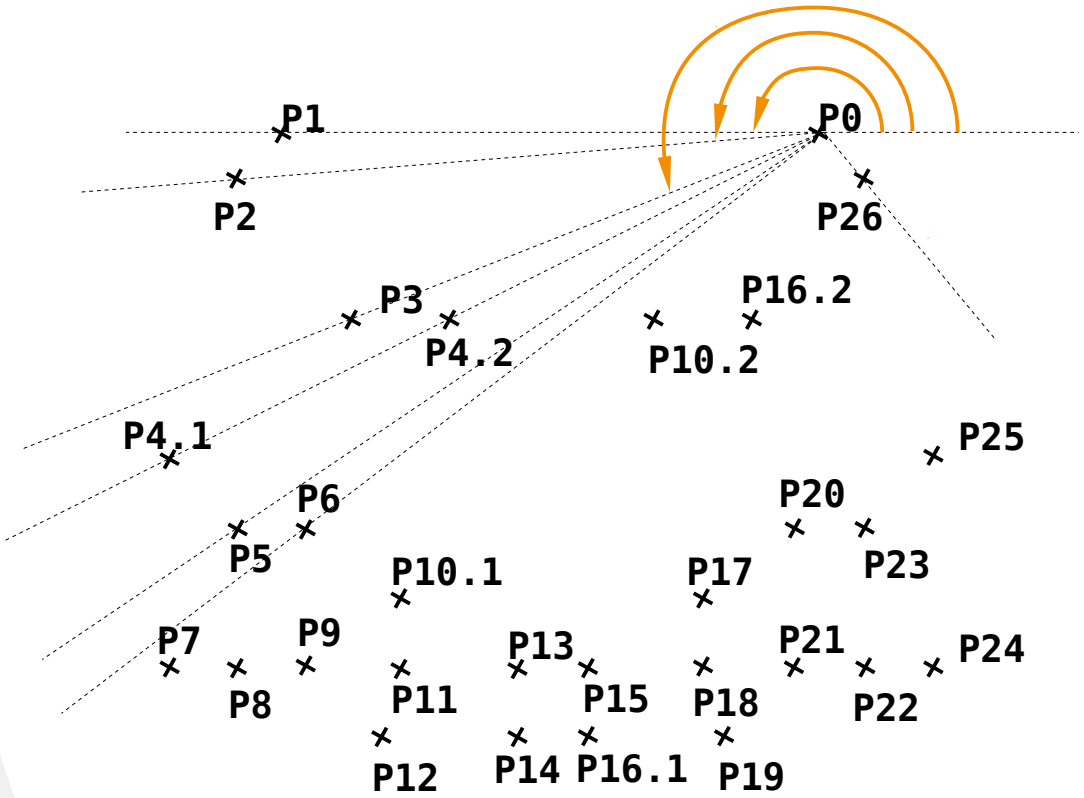
© Thomas Haenselmann, Prof. Dr. rer. nat. habil.

Komponenten u. Konturen

Konvexe Hülle

Starte mit Punkt P0 der sicher auf der konvx. Hülle liegt, z.B. wegen max. X- oder Y-Komponente.

Rotiere Strahl in eine Richtung und nummeriere Punkte fortlaufend.

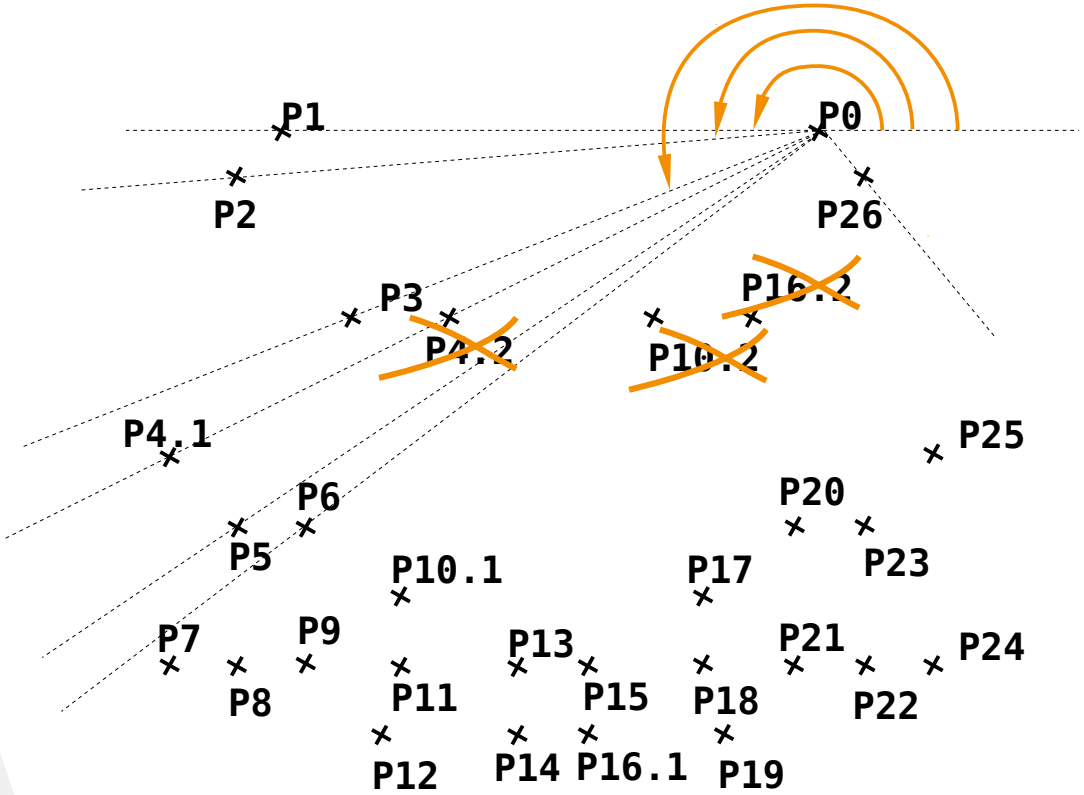


© Thomas Haenselmann, Prof. Dr. rer. nat. habil.

Komponenten u. Konturen

Konvexe Hülle

Bei gleichzeitig getroffenen Punkten: Verwirf alle näheren, behalte den Entferntesten.



© Thomas Haenselmann, Prof. Dr. rer. nat. habil.

Komponenten u. Konturen

Konvexe Hülle

Initialisierung

P_0->P_1 ist die erste Kante der konvexen Hülle.
i=2
Lege die Kante auf einen Stapel (Speicher)

Schleife

(1) Betrachte den nächsten Punkt P_i und die davor liegende Kante k zwischen P_{i-2}->P_{i-1}

(2) Diese Kante k teilt den Raum in einem rechten und linken Halbraum. Liegt P_i im linken Halbraum?

Ja: Erzeuge eine neue Kante vom Endpunkt der Kante k nach P_i.
Lege diese neue Kante auf den Stapel. i=i+1.
Fahre mit (1) fort.

Nein: Entferne die letzte Kante vom Stapel.
Wir arbeiten mit der Kante weiter, die nun auf dem Stapel ganz oben liegt. Fahre mit (2) fort.

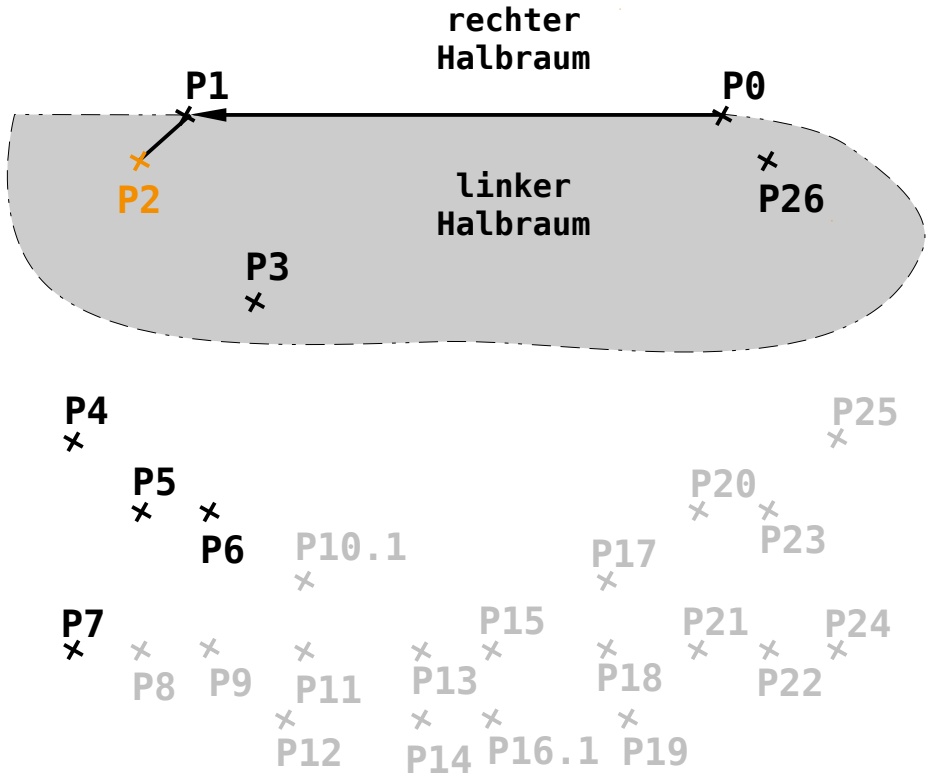
© Thomas Haenselmann, Prof. Dr. rer. nat. habil.

Komponenten u. Konturen

Konvexe Hülle

Erzeuge Kante vom ersten zum nächsten Punkt. Lege die Kante auf dem Stapel(-Speicher) ab.

Sie teilt den Raum in einen linken und einen rechten Halbraum.

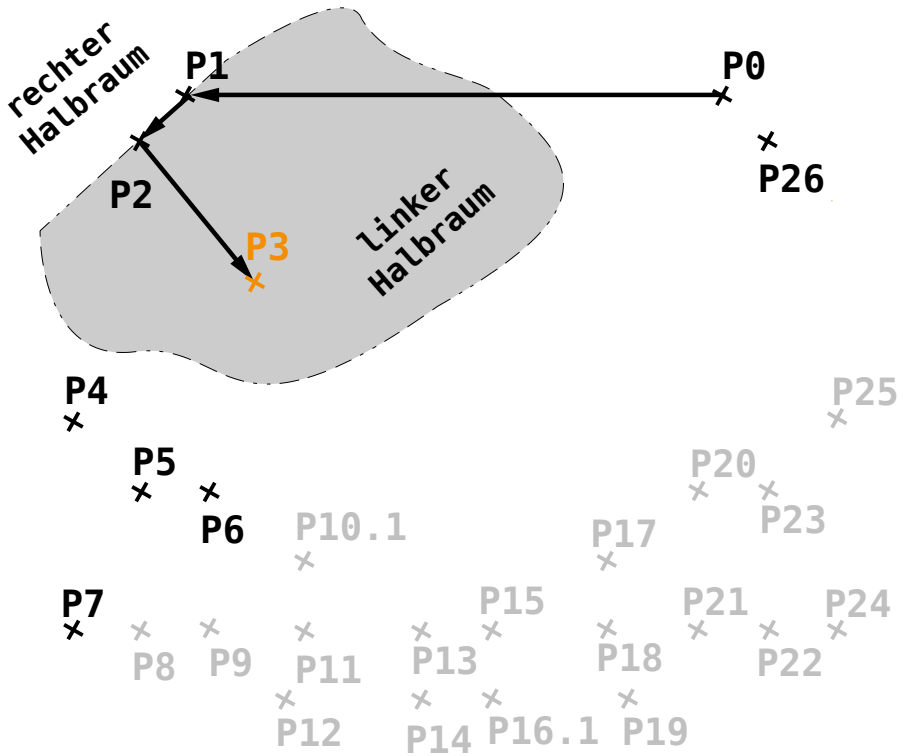


© Thomas Haenselmann, Prof. Dr. rer. nat. habil.

Komponenten u. Konturen

Konvexe Hülle

Ist der nächste Punkt im linken Halbraum, so erzeuge eine neue Kante und lege Sie auf den Stapel.



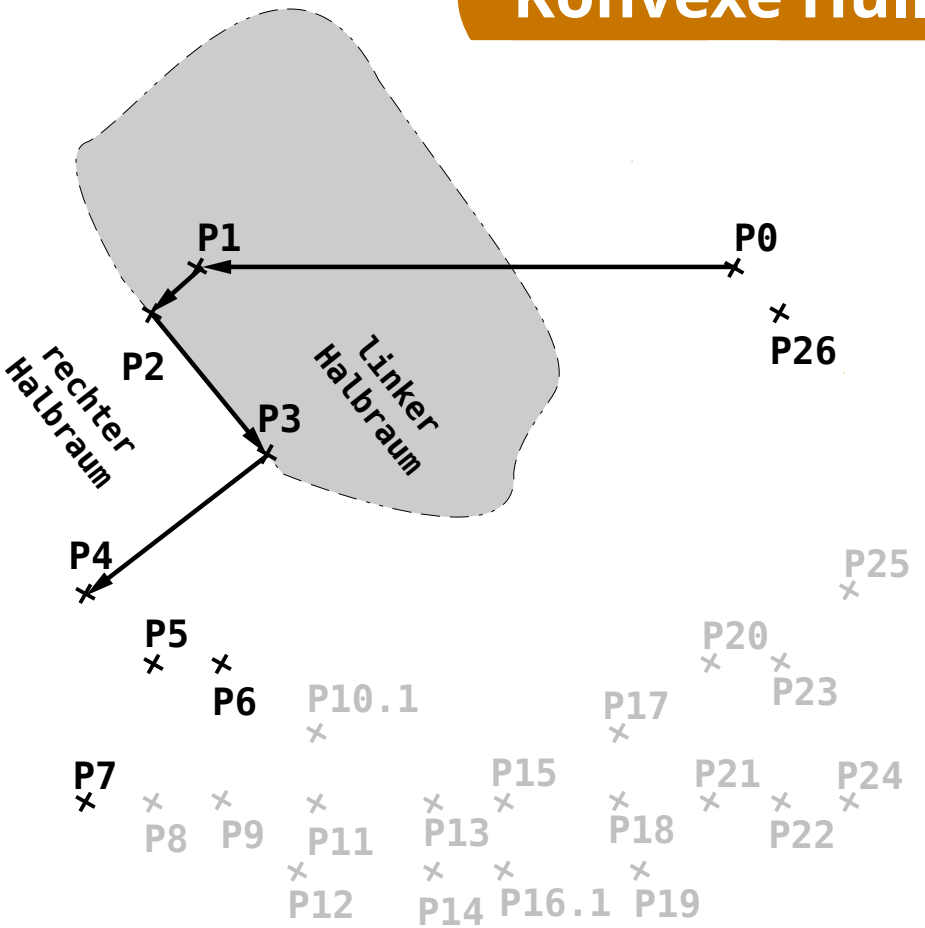
© Thomas Haenselmann, Prof. Dr. rer. nat. habil.

Komponenten u. Konturen

Konvexe Hülle

Befindet sich der nächste Punkt im rechten Halbraum, so verwirf die letzte Kante auf dem Stapel und ziehe eine neue Kante zum aktuellen Punkt.

Und so weiter…

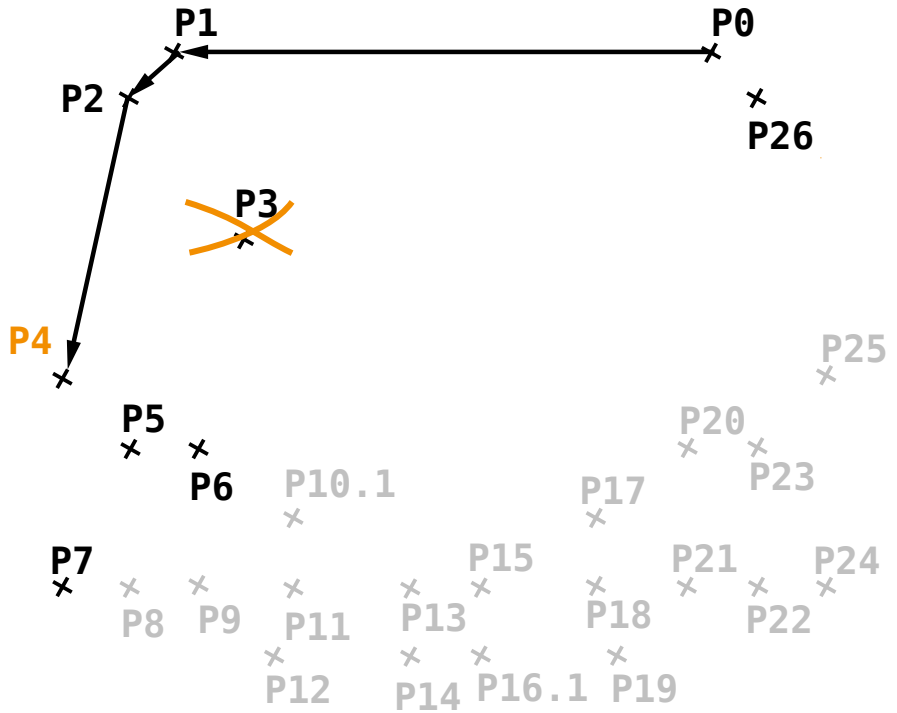


© Thomas Haenselmann, Prof. Dr. rer. nat. habil.

Komponenten u. Konturen

Konvexe Hülle

Kante P2→P3 entfernt. Ziehe dafür neue Kante P2→P4.



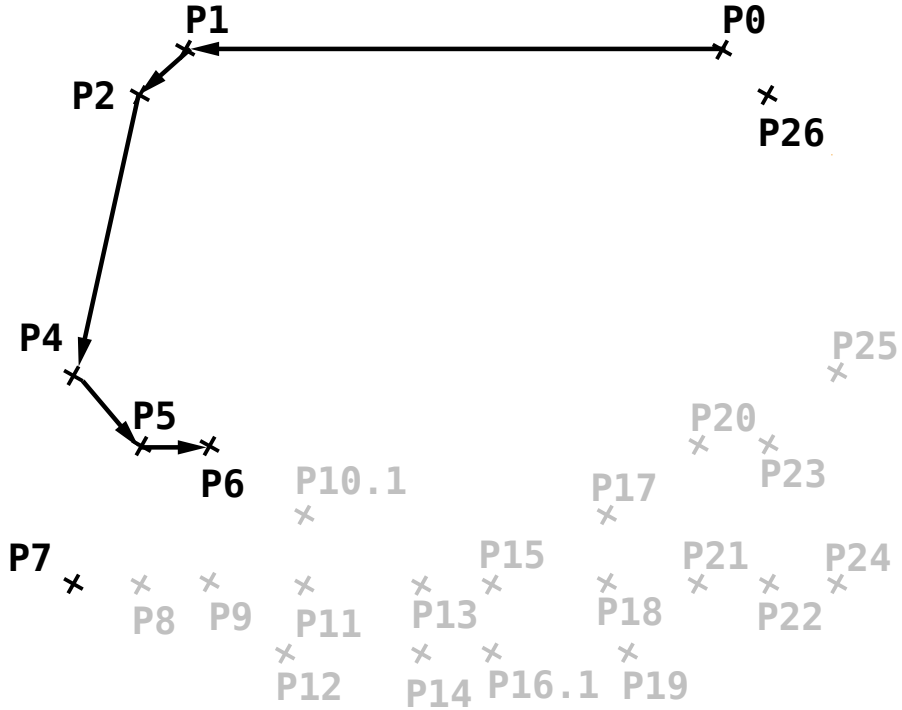
© Thomas Haenselmann, Prof. Dr. rer. nat. habil.

Komponenten u. Konturen

Konvexe Hülle

Füge Kante P4→P5 (erst mal unverdächtig) und dann P5→P6 hinzu.

Gleich merkt der Algorithmus, dass beide nicht auf der konvexen Füllen liegen und wieder entfernt werden müssen. Jetzt aber noch nicht.

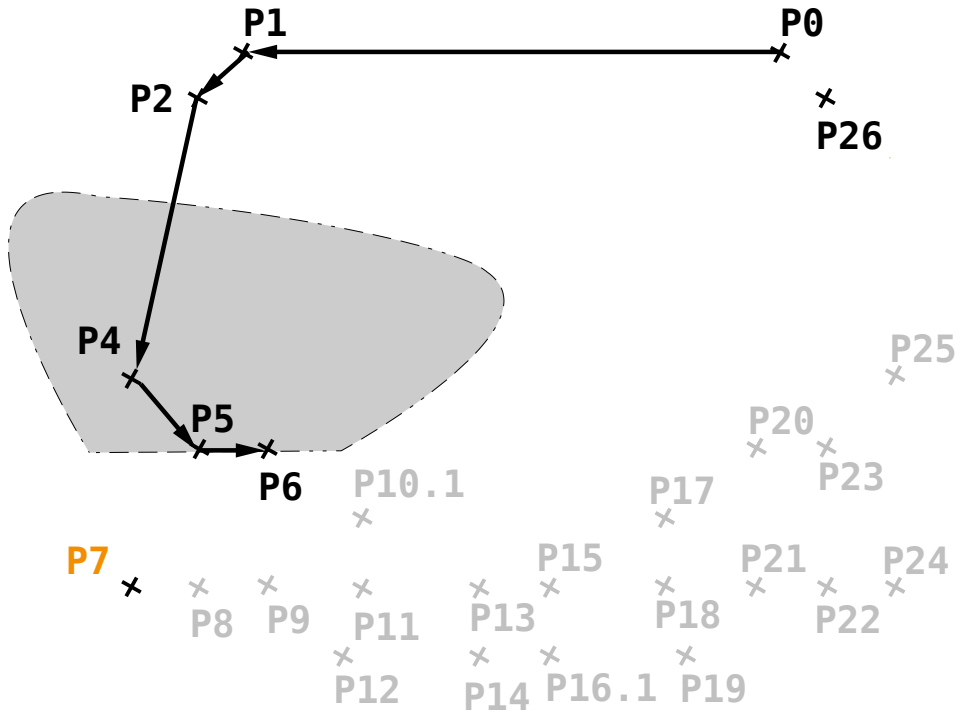


© Thomas Haenselmann, Prof. Dr. rer. nat. habil.

Komponenten u. Konturen

Konvexe Hülle

Algorithmus holt sich die nächste Ecke P7 und stellt fest: Ist im falschen Halbraum.

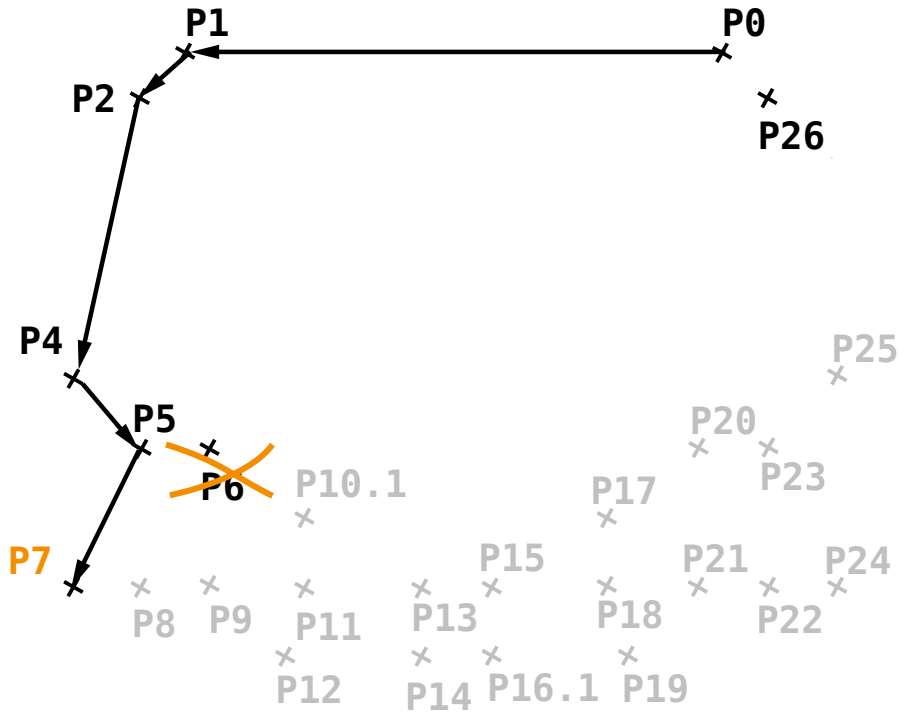


© Thomas Haenselmann, Prof. Dr. rer. nat. habil.

Komponenten u. Konturen

Konvexe Hülle

Folglich war mindestens die letzte Kante P5→P6 falsch und wird wieder entfernt.

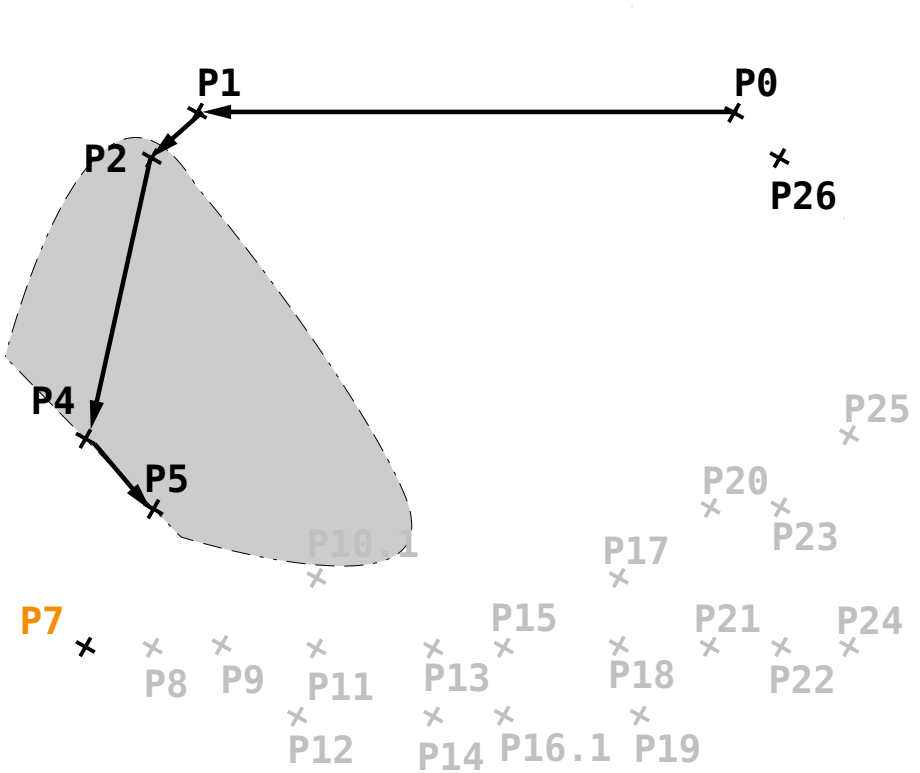


© Thomas Haenselmann, Prof. Dr. rer. nat. habil.

Komponenten u. Konturen

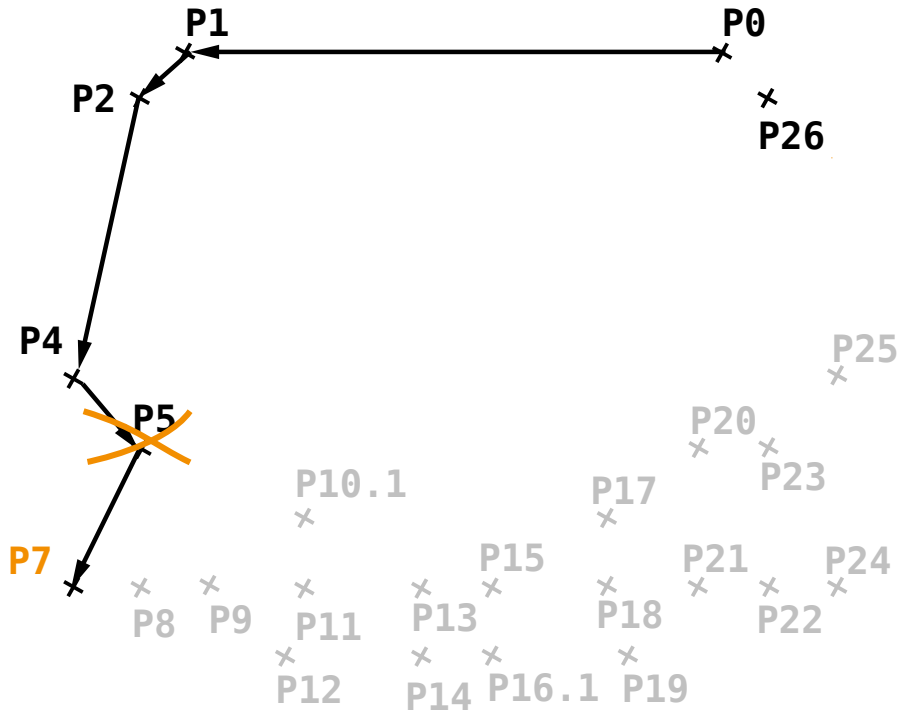
Konvexe Hülle

Jetzt liegt P7 aber immer noch im falschen Halbraum. Dem Algorithmus folgend wird auch jetzt wieder die letzte Kante P4→P5 entfernt...



© Thomas Haenselmann, Prof. Dr. rer. nat. habil.

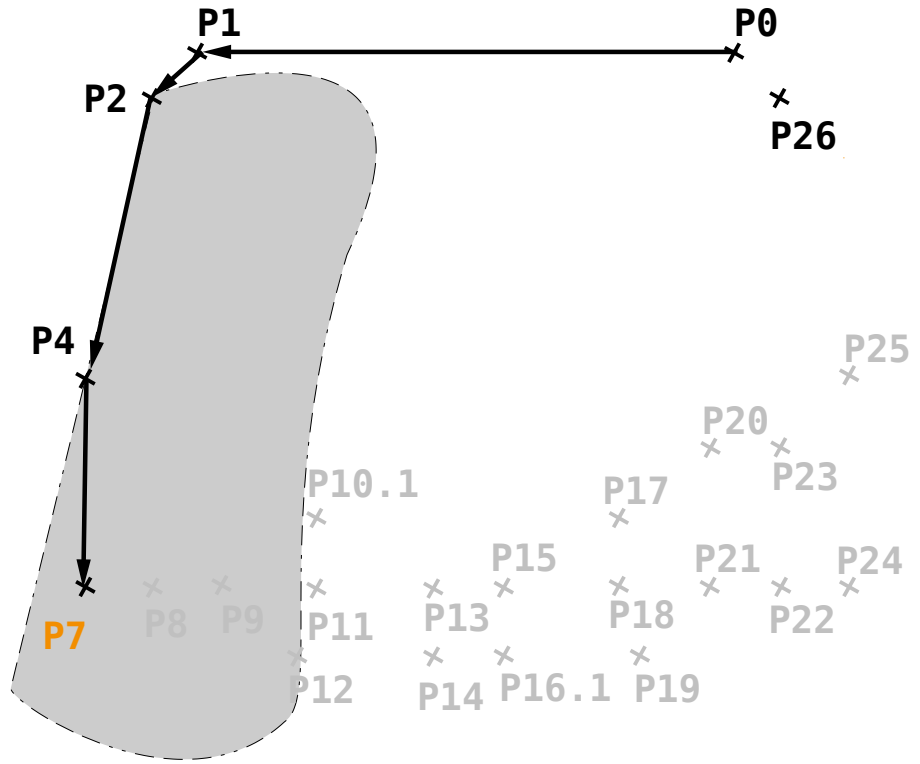
Kante P4→P5 entfernt



Komponenten u. Konturen

Konvexe Hülle

Nun liegt P7 endlich im richtigen Halbraum. Eine neue Kante P4→P7 kann hinzugefügt werden.

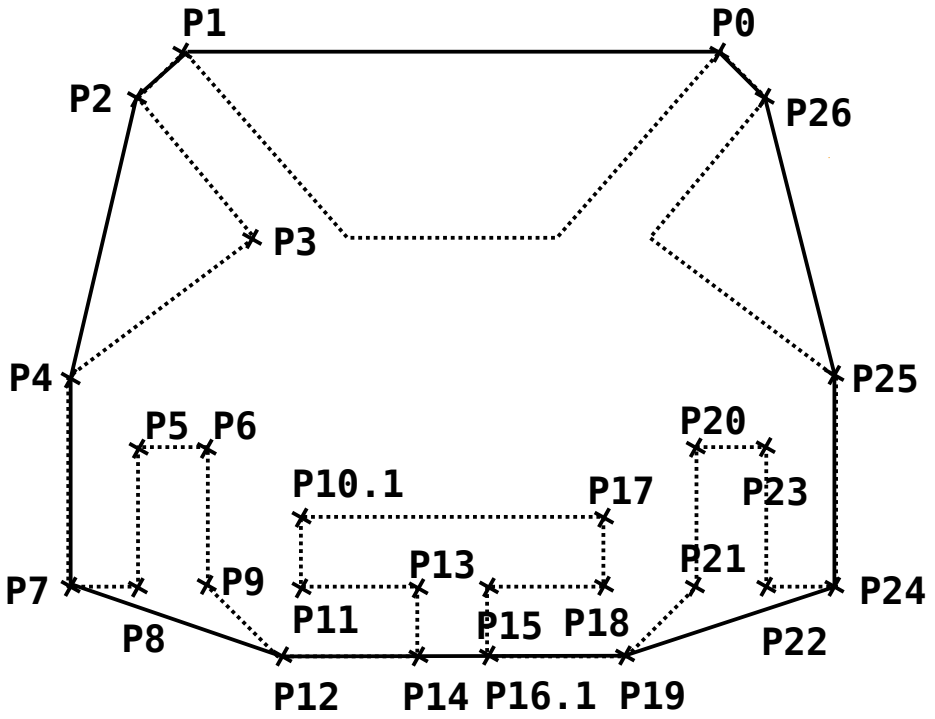


© Thomas Haenselmann, Prof. Dr. rer. nat. habil.

Komponenten u. Konturen

Konvexe Hülle

Konvexe Hülle nachdem der Algorithmus terminiert hat.



© Thomas Haenselmann, Prof. Dr. rer. nat. habil.