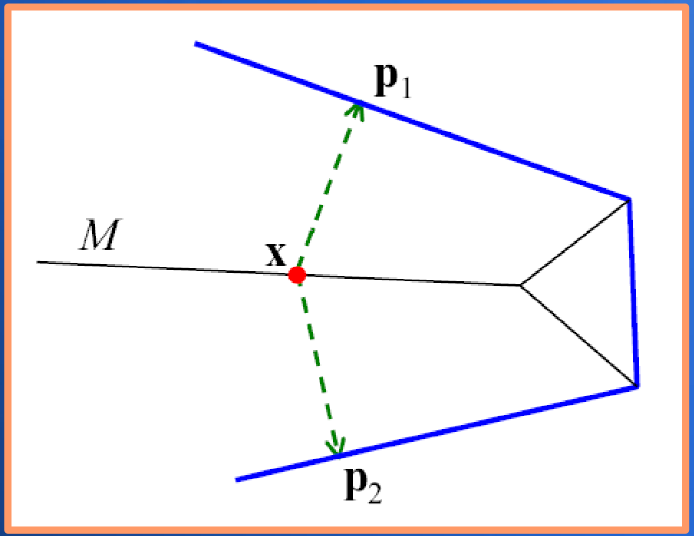
Index

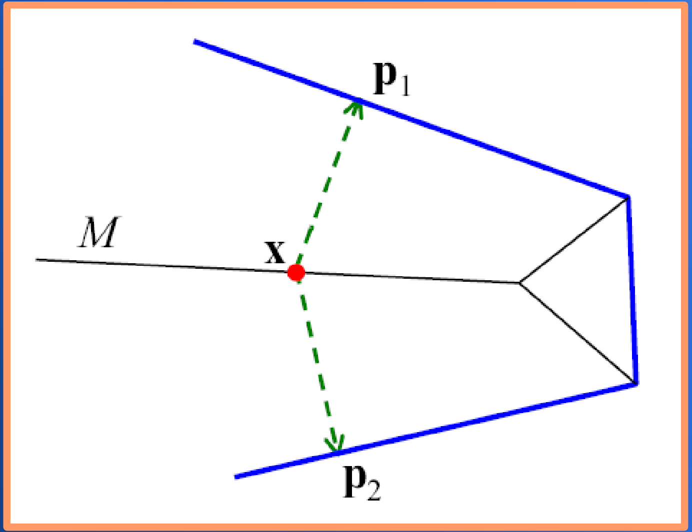
Mediale Achse

- Ist eine Punktmenge
- mehr als 1 kürzester Weg zur Oberfläche



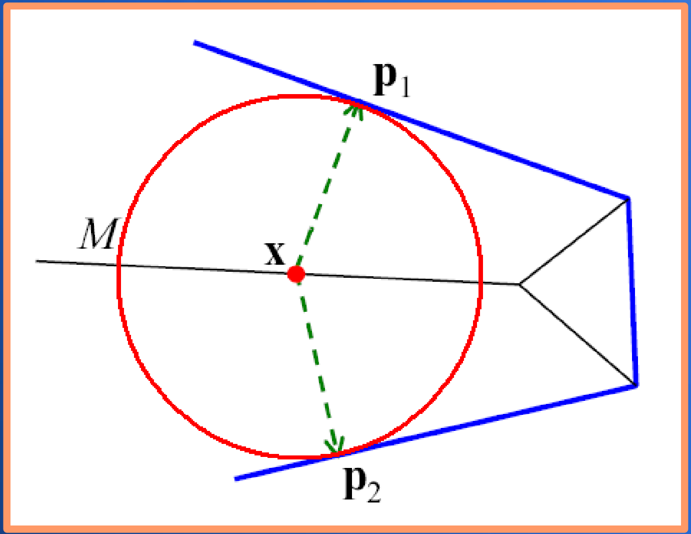
Mediale Achse

- Ist eine Punktmenge
- mehr als 1 kürzester Weg zur Oberfläche



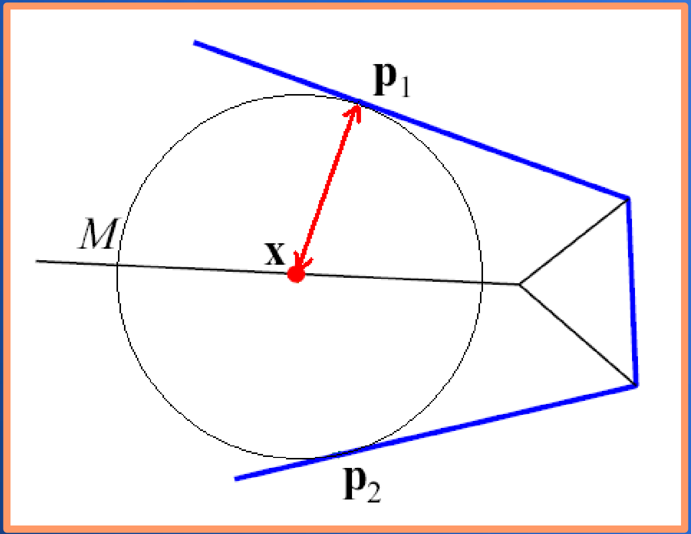
Mediale Achse

- Ist eine Punktmenge
- mehr als 1 kürzester Weg zur Oberfläche
- *Alternativ:* Zentralpunkte der medialen Kreise, Kugeln die S tangential in mehr als einem Punkt berühren



Mediale Achse

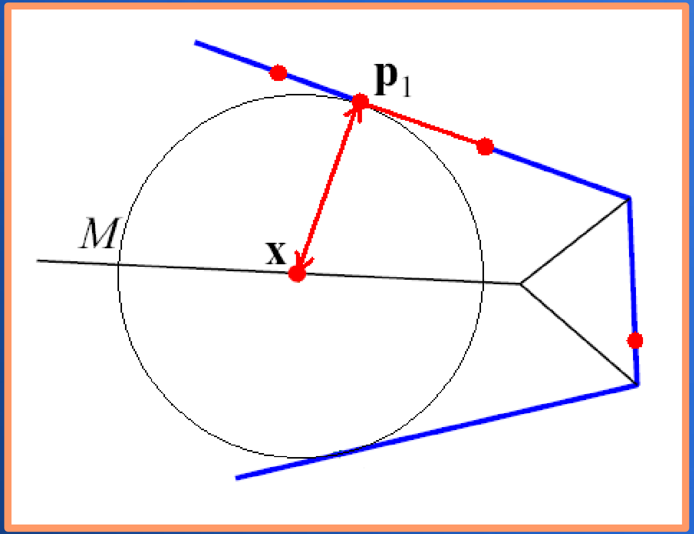
- Local feature size: f(p)
  kleinster Abstand vom Punkt p zur Medialen Achse



- *Alternativ:*
  Zentralpunkte der medialen Kreise, Kugeln die S tangential in mehr als einem Punkt berühren

ε - Sample

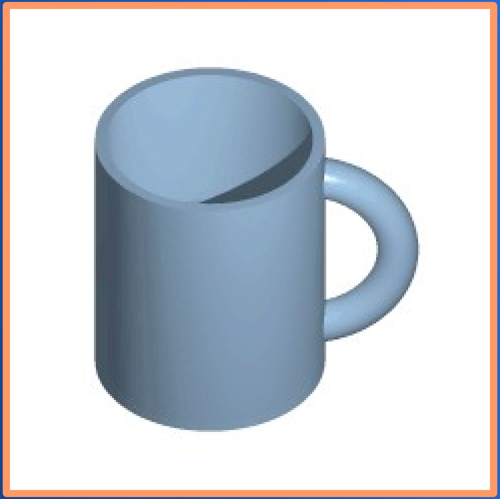
- Local feature size: f(p)
  kleinster Abstand vom Punkt p zur Medialen Achse



- ε = Konstante,
- ε ≤ 0.06 notwendig für den Beweis von Homeomorphism (eng.) zwischen der Triangulation und der Oberfläche
- eine Punktmenge P für die gilt: jeder Punkt $\underbrace{p \in S}$ hat mindestens einen anderen Punkt im Abstand ε f(p)

Homeomorphism

- Begriff in der Topologie
- bijektive, stetige Abbildung zwischen zwei Objekten
- durch Dehnen, Stauchen, Verbiegen, Verzerren, Verdrillen kann man ein Objekt zu einem anderen umwandeln



Homeomorphism

- Conditions des Algorithmus
    I. restricted delaunay condition
    II. small triangle condition
    III. flat triangle condition

Homeomorphism

- Conditions des Algorithmus
    I. restricted delaunay condition
    II. small triangle condition
    III. flat triangle condition
- T ist die Menge der Dreiecke die alle Abtastpunkte überspannt
- Condition I erlaubt die Bildung eines 2-manifold aus T, der alle Vertices beinhaltet, also selbe Eigenschaft wie T
- Conditions II und III erlauben den beweis des Homöomorphismus zw. S und jedem 2-manifold der aus T gebildet wurde und alle vertices enthält

Homeomorphism

I. restricted delaunay condition:

für $\varepsilon \leq 0.1$ gilt, alle Dreiecke aus der restricted delaunay triangulation müssen in der Dreieckmenge T enthalten sein

T beinhaltet alle Dreiecke dessen duale Voronoi Kanten S schneiden

Homeomorphism

II. small triangle condition:

für den Radius r von einem beliebigen Dreieck $\underbrace{t \in T}$ gilt $\underbrace{r \leqslant \frac{1.3\,\epsilon}{1-\epsilon} \cdot f(p)}$, wobei dies für jeden Vertex p im Dreieck t der Fall ist

Der Umkreis von den Dreiecken ist kleiner als die local feature size von dem Punkt p

Homeomorphism

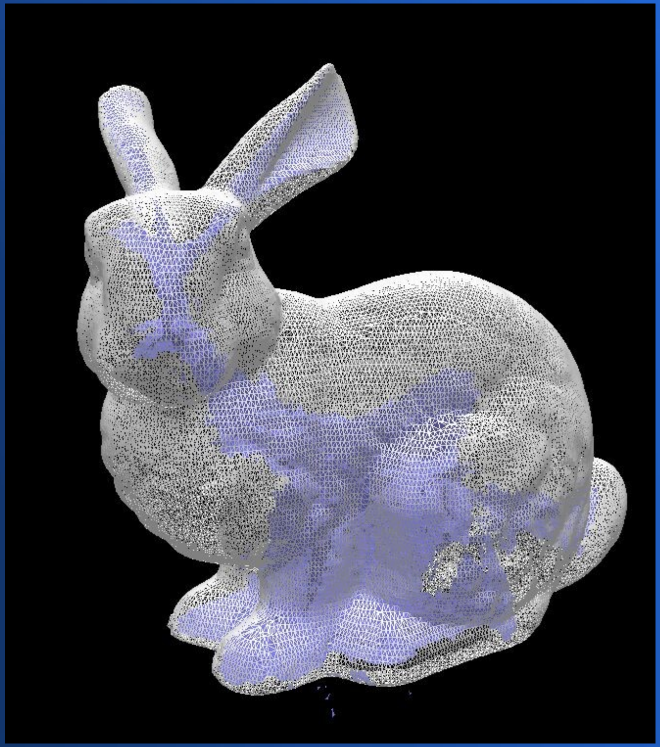
III. flat triangle condition:

für $\varepsilon \leq 0.09$ gilt, die normale von jedem Dreieck $\underbrace{t \in T}$ und die normale von dem Vertex p mit dem grössten Winkel α innerhalb von t bilden einen Spitzwinkel der nicht grösser als $\underbrace{\alpha + \arcsin(2/\sqrt{3} \cdot \sin(2 \cdot \alpha))}$ ist

Die Dreiecke sind flach bezogen auf S, ihre Normalen und die Normalen der Punkte von S stehen in sehr kleinen Winkeln zueinander

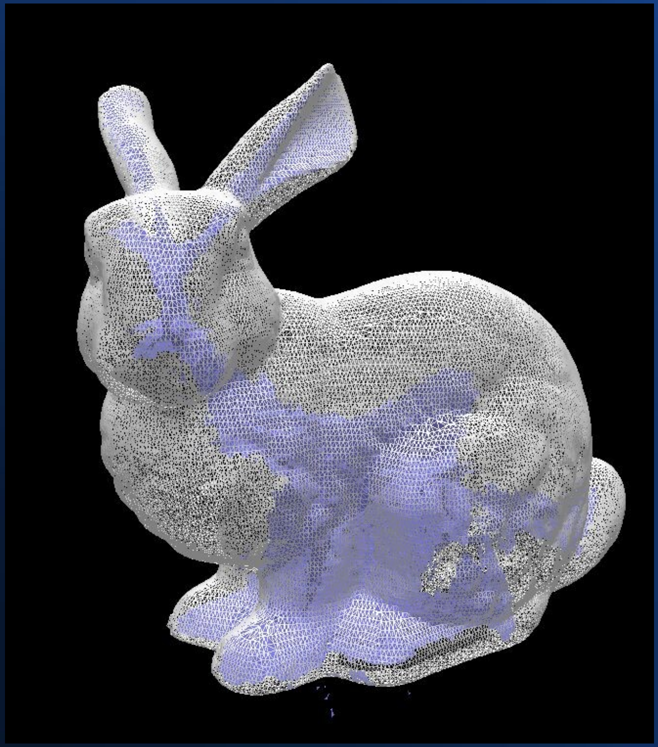
Homeomorphism

Homeomorphism zwischen Cocone Output N und Oberfläche S



Homeomorphism

Homeomorphism zwischen Cocone Output N und Oberfläche S

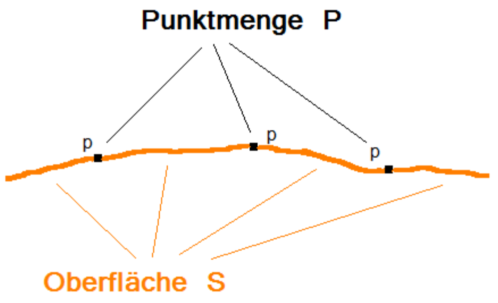


Cocone Algorithmus

- Abtastung
- Delaunay Triangulation
- Voronoi Diagramm
- Pole
- Geschätzte Normale
- Cocone
- Kanten markieren
- Manifold extrahieren

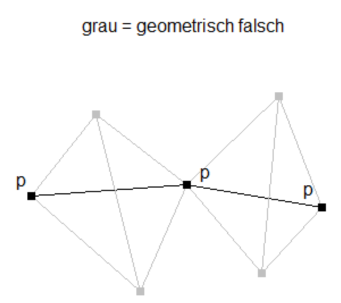
Cocone Algorithmus

- Abtastung
- Delaunay Triangulation
- Voronoi Diagramm
- Pole
- Geschätzte Normale
- Cocone
- Kanten markieren
- Manifold extrahieren



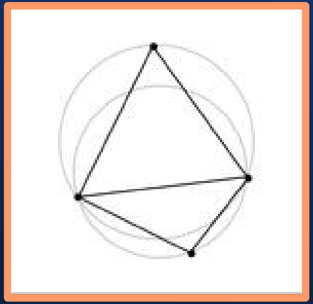
Cocone Algorithmus

- Abtastung
- Delaunay Triangulation
- Voronoi Diagramm
- Pole
- Geschätzte Normale
- Cocone
- Kanten markieren
- Manifold extrahieren

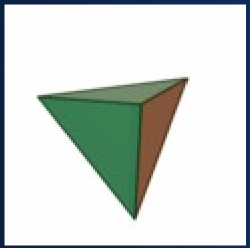


Cocone Algorithmus

- Delaunay Triangulation: D(P) beinhaltet Tetrahedra
- 2D: Umkreisbedingung



- 3D: Umkugelbedingung äquivalent zur Umkreisbedingung über Tetraeder



Cocone Algorithmus

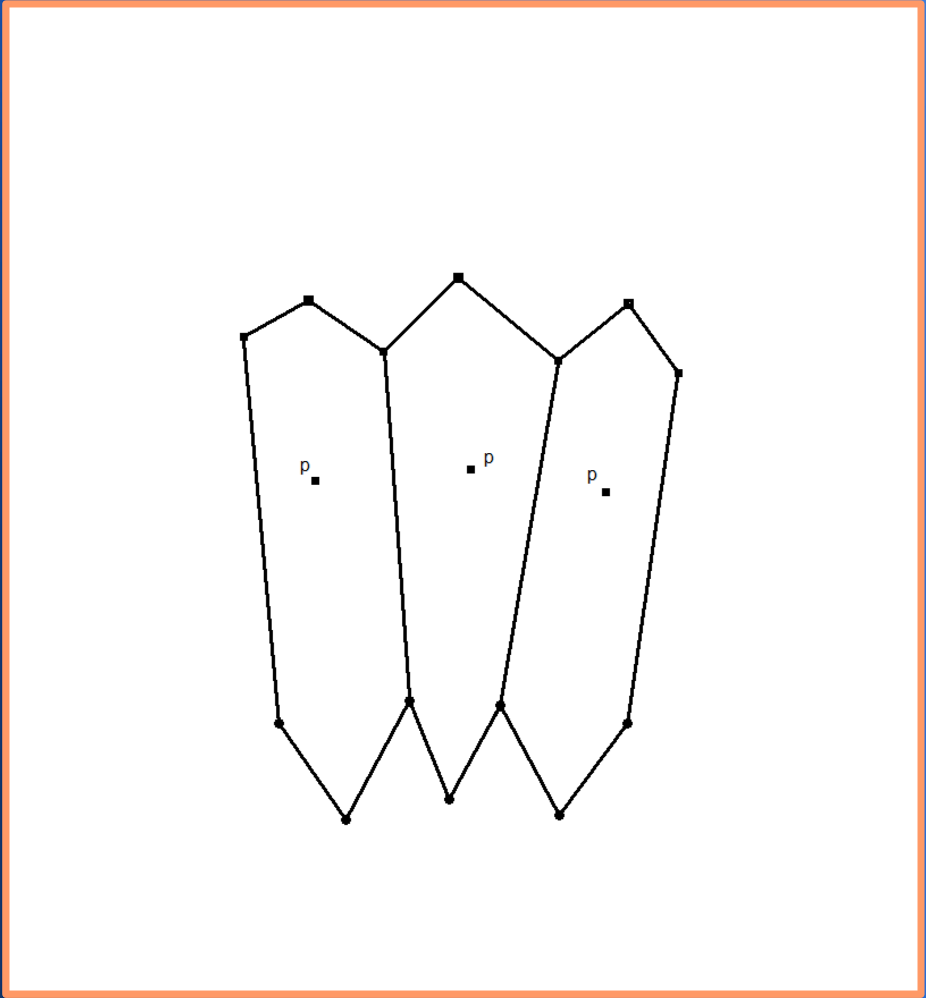
- eingeschränktes Voronoi Diagramm: $V(P,S)$
  - Beinhaltet eingeschränkte Voronoi Zellen: $V(p,S) = V(p) \cap S$

Cocone Algorithmus

- eingeschränktes Voronoi Diagramm: V(P,S)
  - Beinhaltet eingeschränkte Voronoi Zellen: V(p,S) = V(p) ∩ S
- eingeschränkte Delaunay Triangulation: D(P,S)
  - Besteht aus Dreiecken die dual zu V(p,S) sind
  - Kante pq ist in D(P,S), wenn V(p,S)∩ V(q,S) nicht leer
  - Dreieck pqr ist in D(P,S), wenn V(p,S)∩ V(q,S)∩ V(r,S) nicht leer
  - Wenn S keine Voronoi Vertices schneidet sind in D(P,S) keine Tetraeder
  - Homeomorphism zwischen D(P,S) und S (Bedingungen notwendig)

Cocone Algorithmus

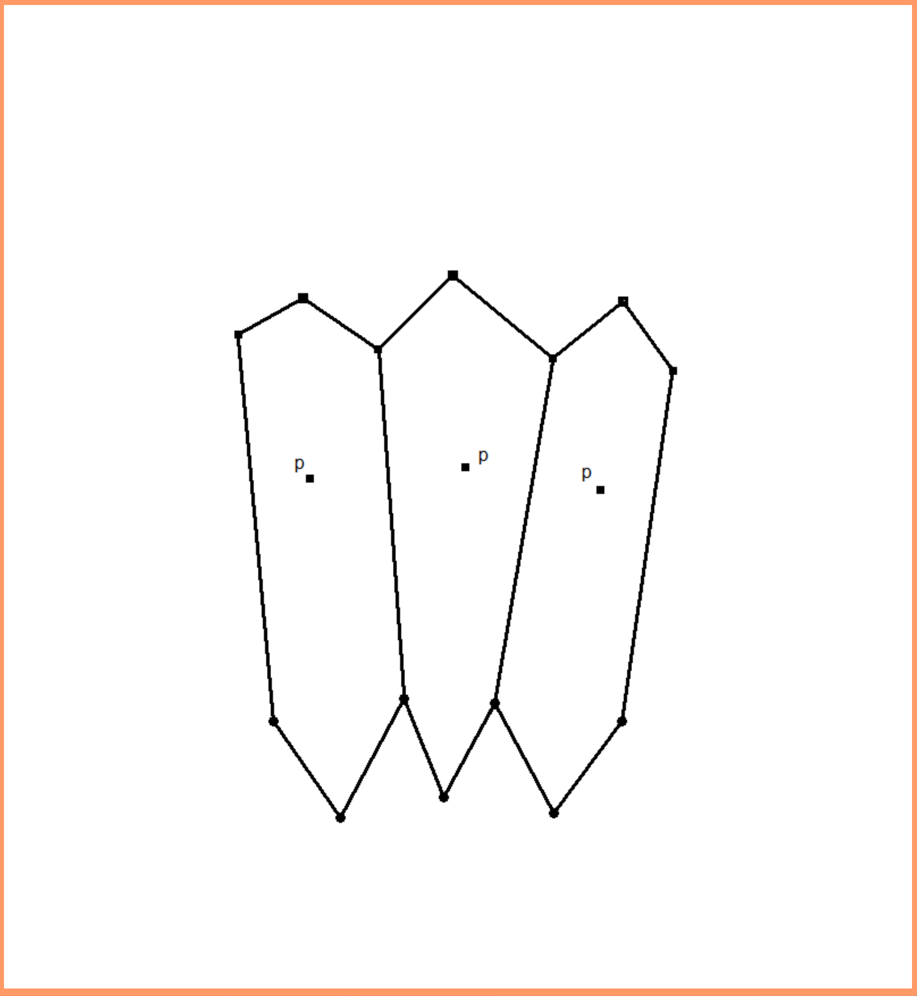
- Abtastung
- Delaunay Triangulation
- Voronoi Diagramm
- Pole
- Geschätzte Normale
- Cocone
- Kanten markieren
- Manifold extrahieren



Cocone Algorithmus

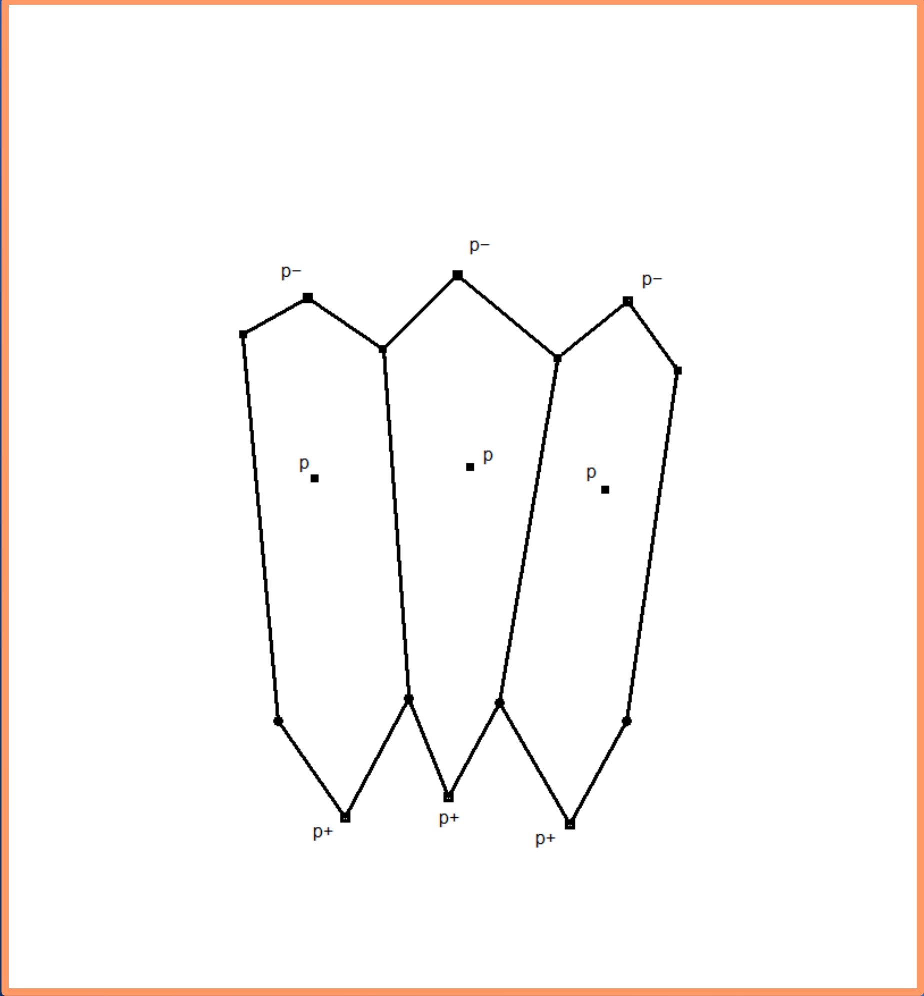
- Voronoi Diagramm: V(P)

  dual zu D(P)
- Voronoi Zelle: V(p)
  - Punktmenge $(\mathbb{R}^3)$
  - Zentrum p
  - $\underbrace{|px| \leq |qx| \quad x \in \mathbb{R}^3 \quad p, q \in P \quad p \neq q}$
- der Vector vom punkt p zu irgendeinem punkt $\underbrace{x \in V(p)}$ wird x' genannt



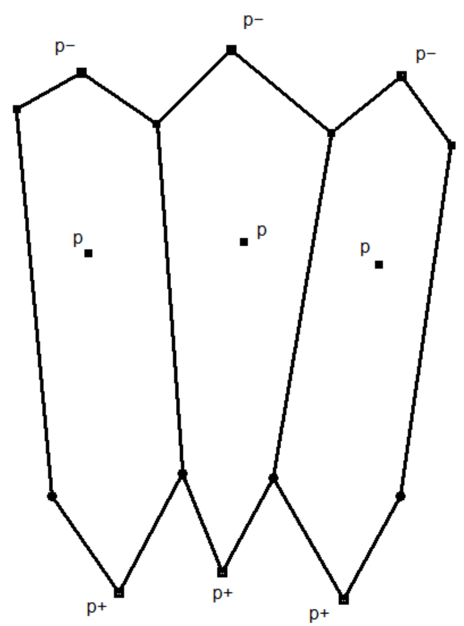
Cocone Algorithmus

- Abtastung
- Delaunay Triangulation
- Voronoi Diagramm
- Pole
- Geschätzte Normale
- Cocone
- Kanten markieren
- Manifold extrahieren



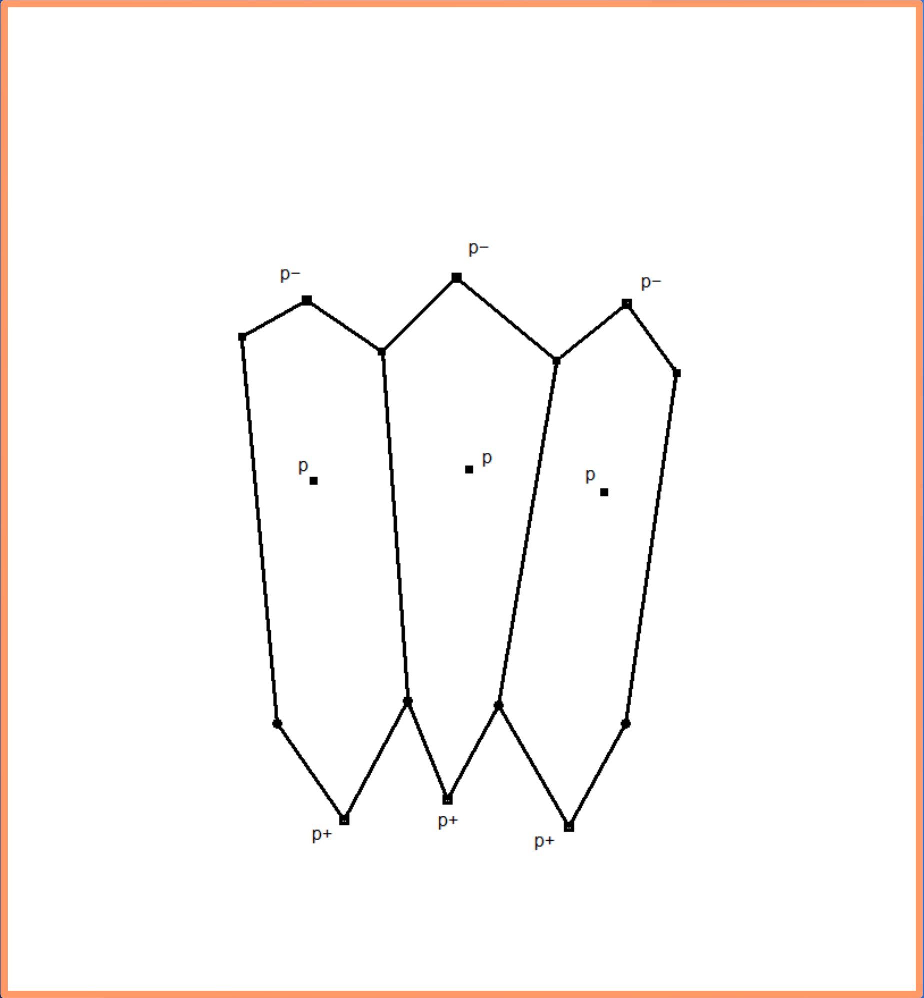
Cocone Algorithmus

- der am weitesten vom Punkt p entfernte Punkt p+ ist der positive Pol



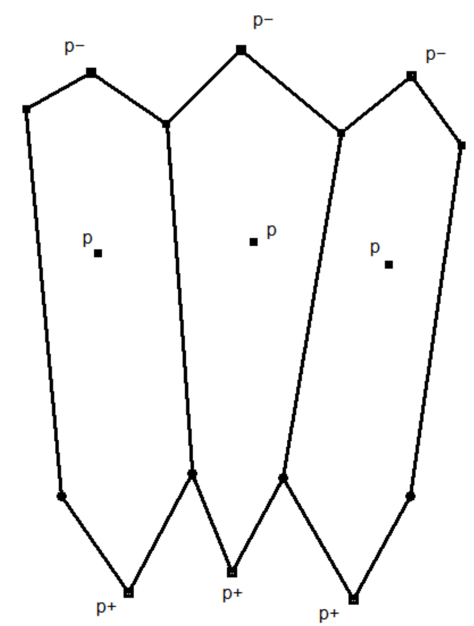
Cocone Algorithmus

- der am weitesten vom Punkt p entfernte Punkt p+ ist der positive Pol
- p− ist der negative Pol, und der Winkel zwischen p−' und p+' ist mindestens $\pi/2$ und er ist auch (am weitesten) entfernt von p
- $\underbrace{p+, p- \in V(p)}$



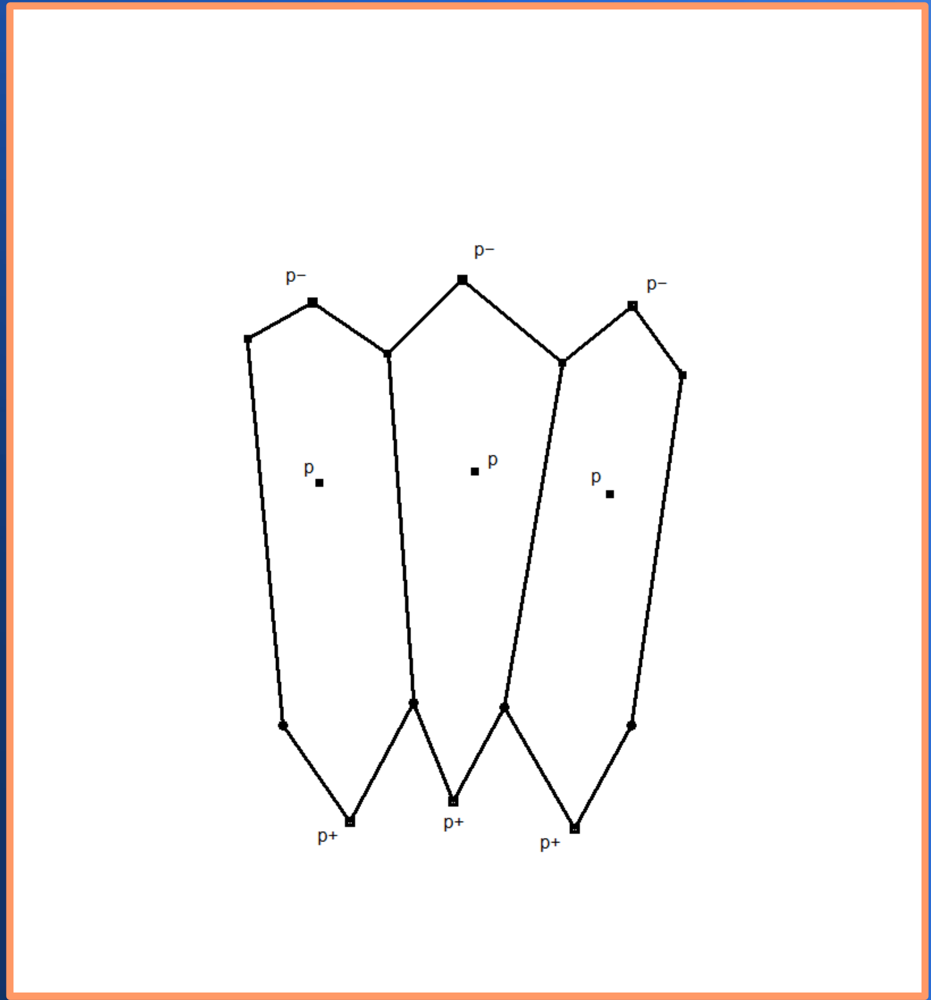
Cocone Algorithmus

- der am weitesten vom Punkt p entfernte Punkt p+ ist der positive Pol
- p− ist der negative Pol, und der Winkel zwischen p−' und p+' ist mindenstens $\pi/2$ und er ist auch (am weitesten) entfernt von p
- $\underbrace{p+, p- \in V(p)}$
- Schleife über alle Vertices



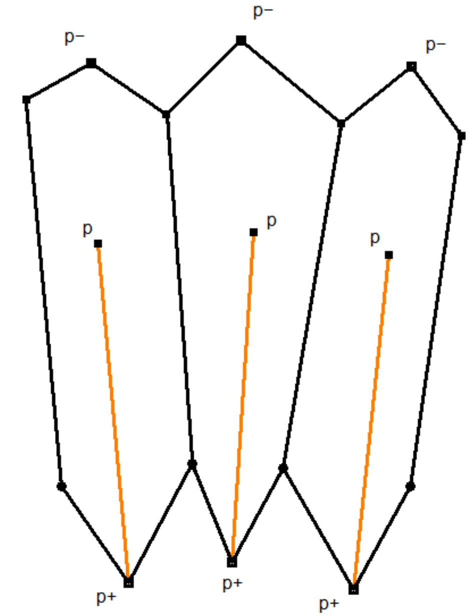
Cocone Algorithmus

- Wenn V(p) unbegrenzt ist liegt p+ im unendlichen und der Vektor p+' ist der Durchschnitt aller Voronoi Kanten-Vektoren



Cocone Algorithmus

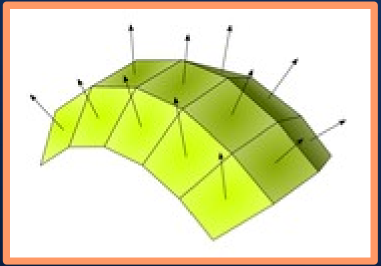
- Abtastung
- Delaunay Triangulation
- Voronoi Diagramm
- Pole
- Geschätzte Normale
- Cocone
- Kanten markieren
- Manifold extrahieren

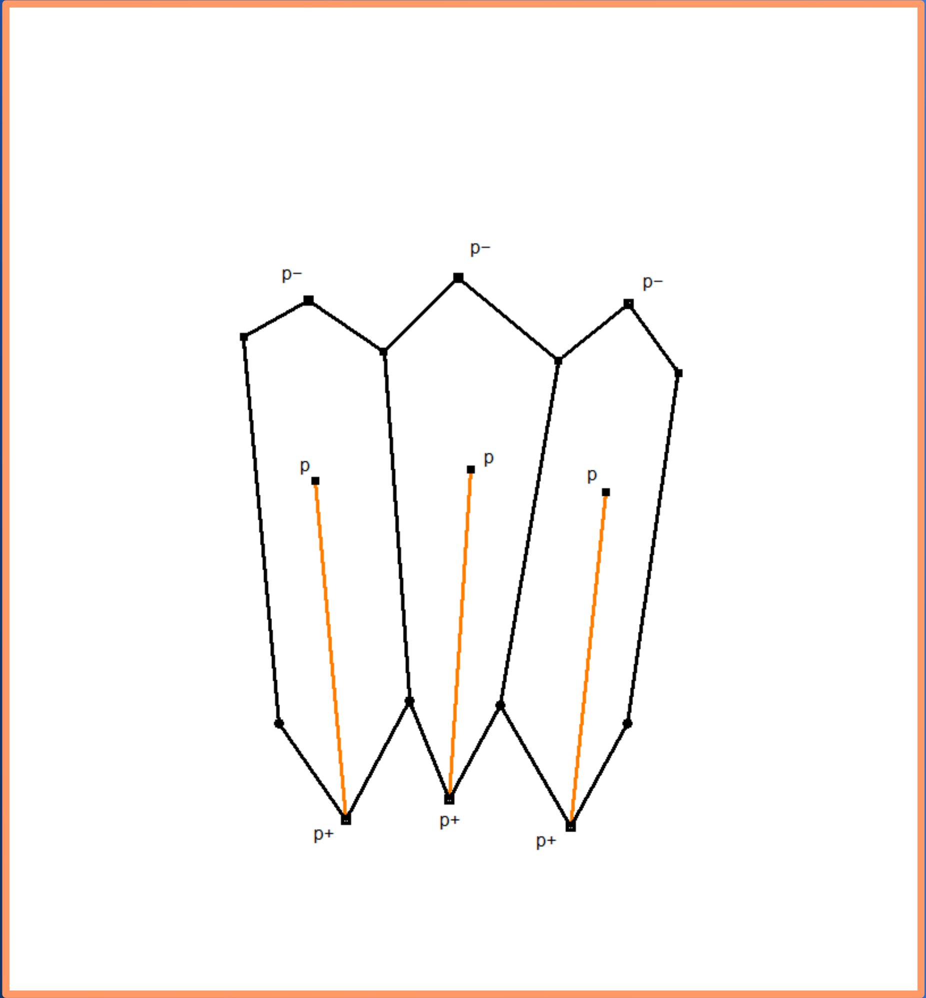


Cocone Algorithmus

- Echte Normale: n(p)

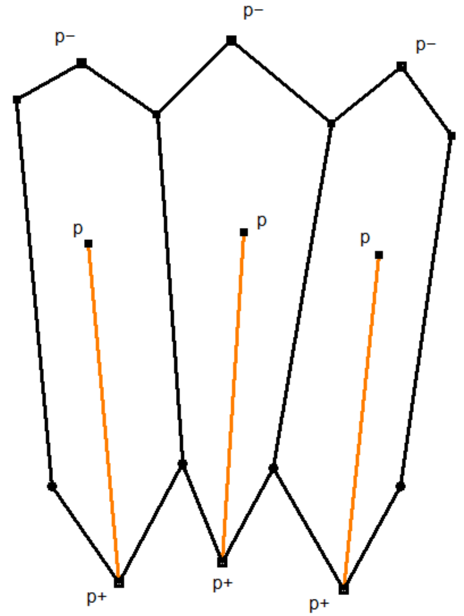
  Normale am Punkt p steht senkrecht zur Oberflächen-tangente am Punkt p





Cocone Algorithmus

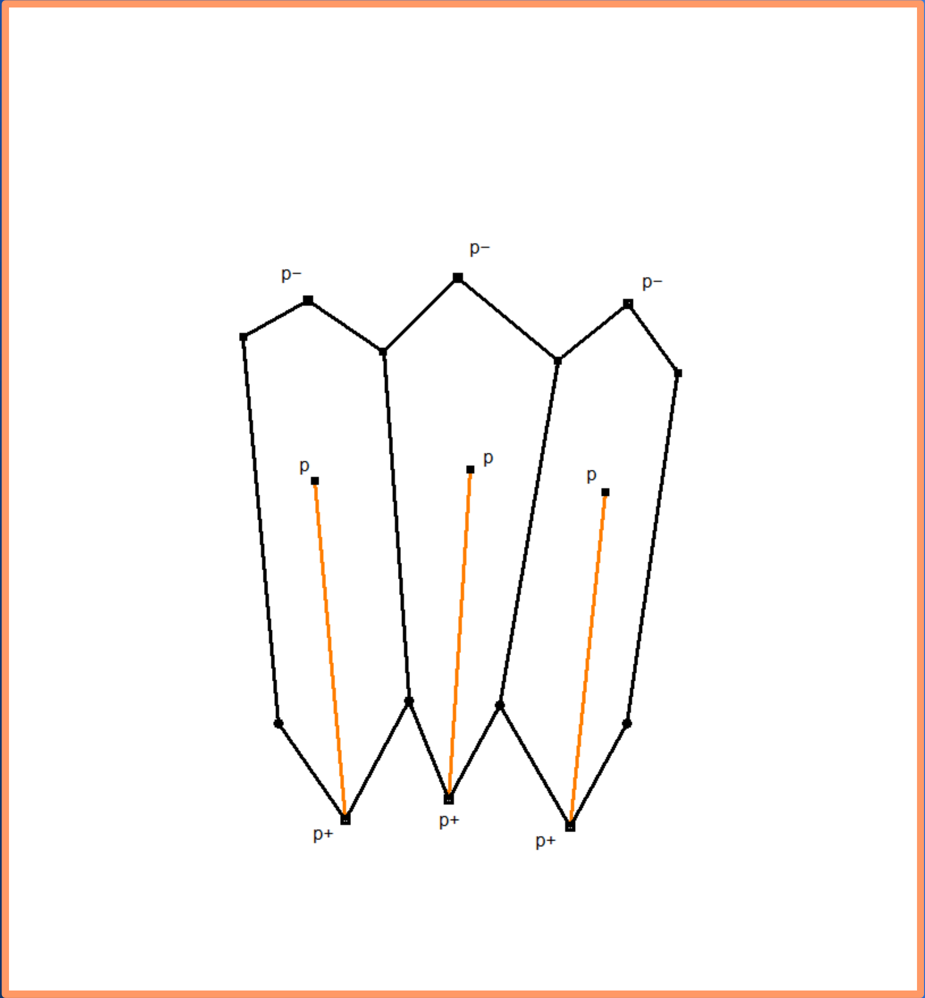
- Geschätzte Normale: gn(p)
  die Linie zwischen p und p+



Cocone Algorithmus

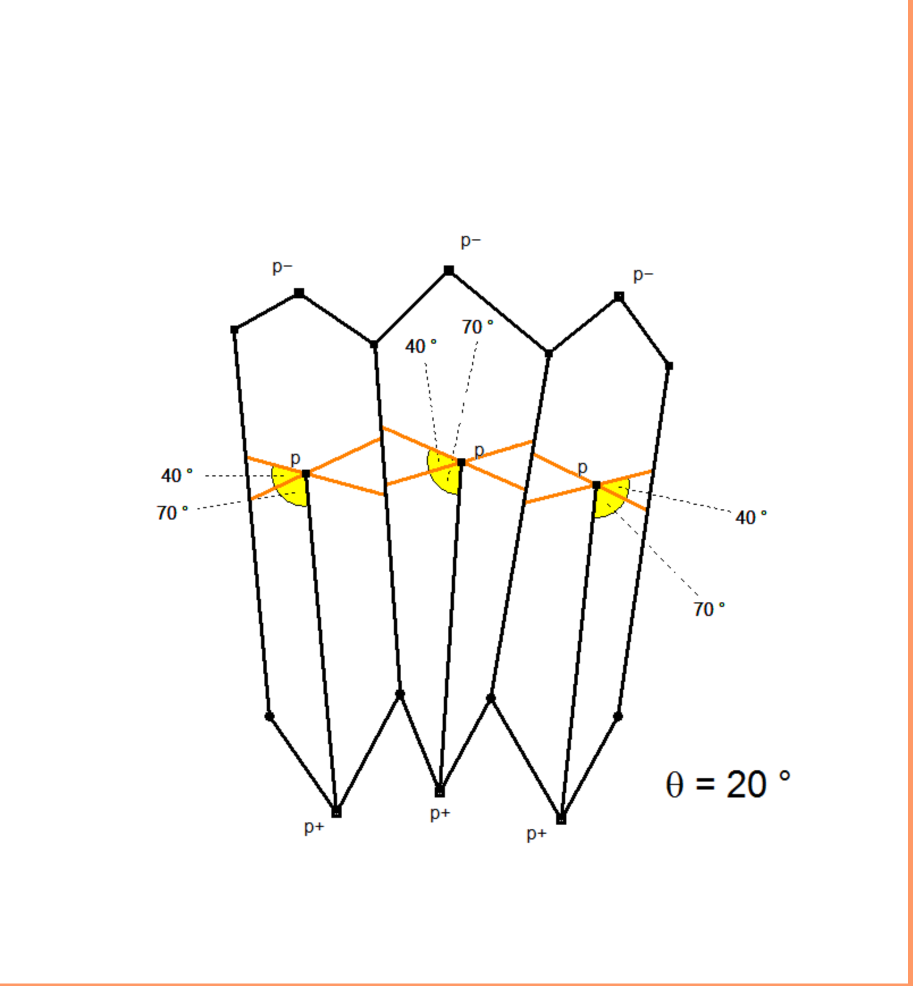
- Geschätzte Normale: gn(p)
  die Linie zwischen p und p+

- identisch mit dem Vektor p+'
- Berechnung durch Differenz der Einheitsvektoren p und p+



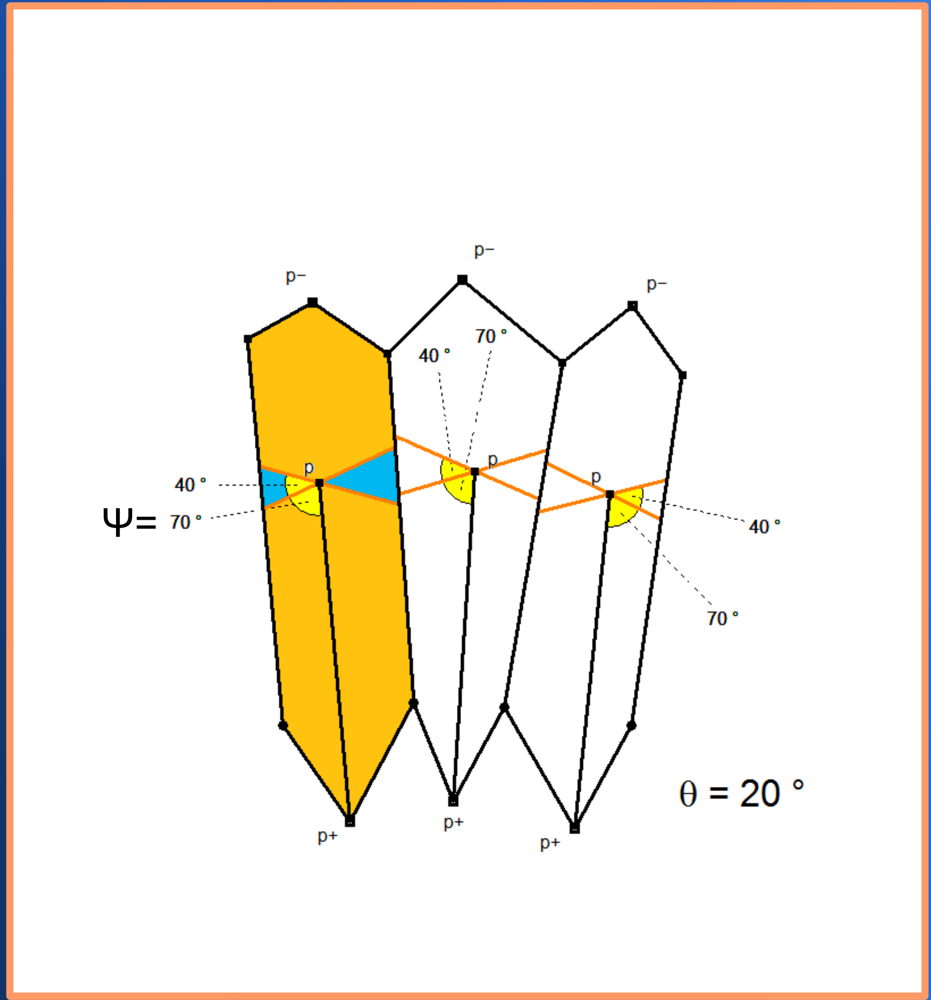
Cocone Algorithmus

- Abtastung
- Delaunay Triangulation
- Voronoi Diagramm
- Pole
- Geschätzte Normale
- Cocone
- Kanten markieren
- Manifold extrahieren



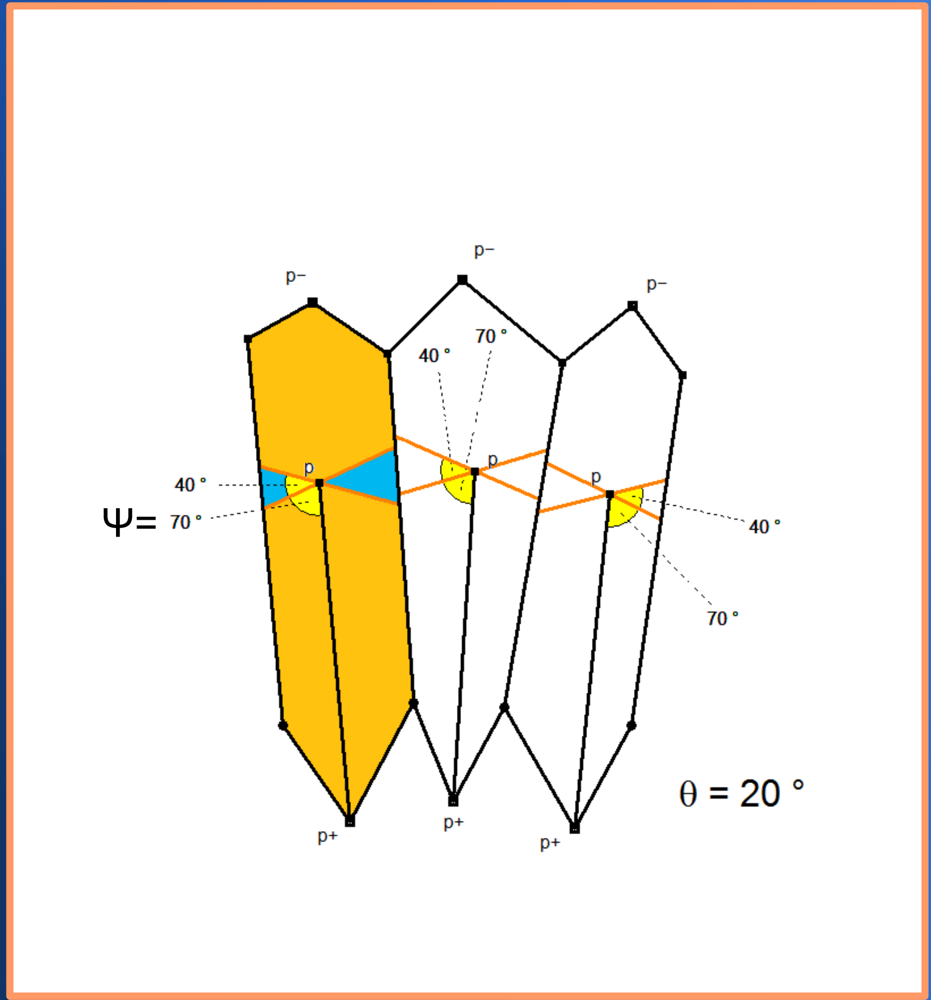
Algorithmus

- Cocone C(p)
  - complemented cones
  - Komplement von dem Doppelkonus mit dem Zentrum p
- Punktmenge C(p)=(x ∍ V(p); der Winkel Ψ zwischen x' und p+' ist $\geq$ $3\pi/8$)



Cocone Algorithmus

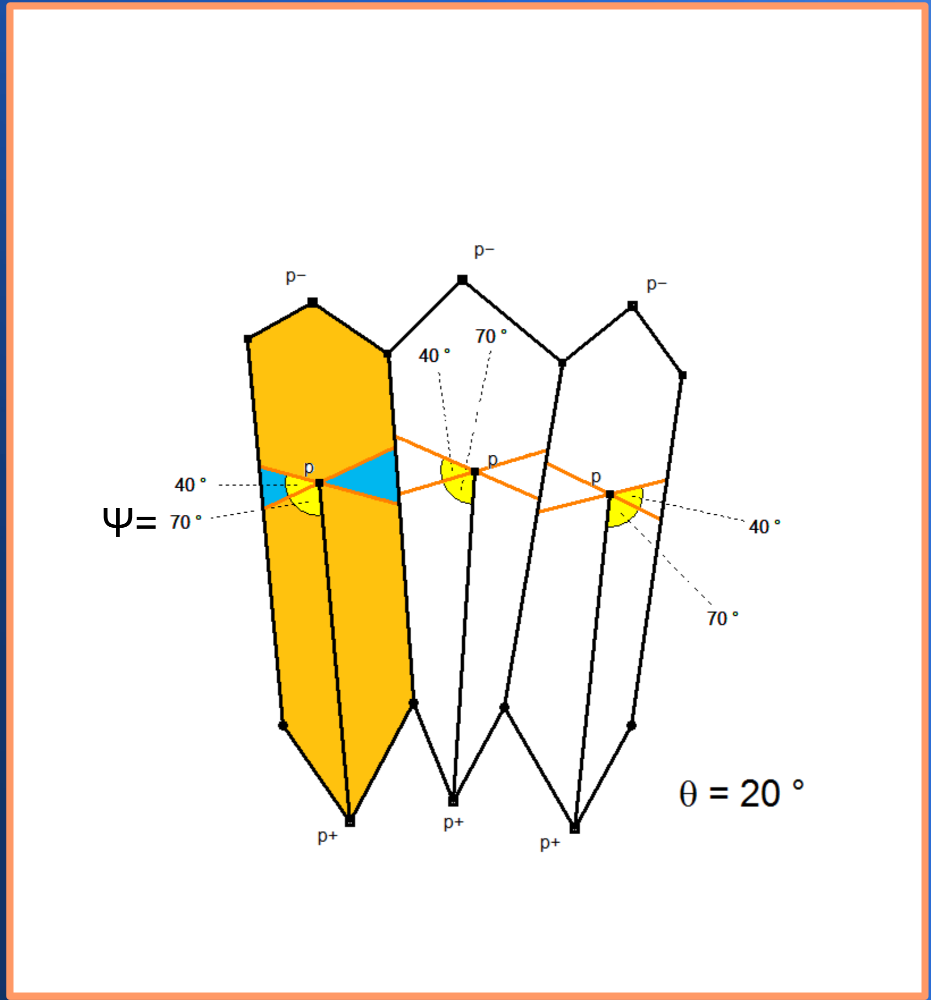
- Θ muss klein genug sein, maximal $\leq \pi/8 = 22.5°$



Cocone Algorithmus

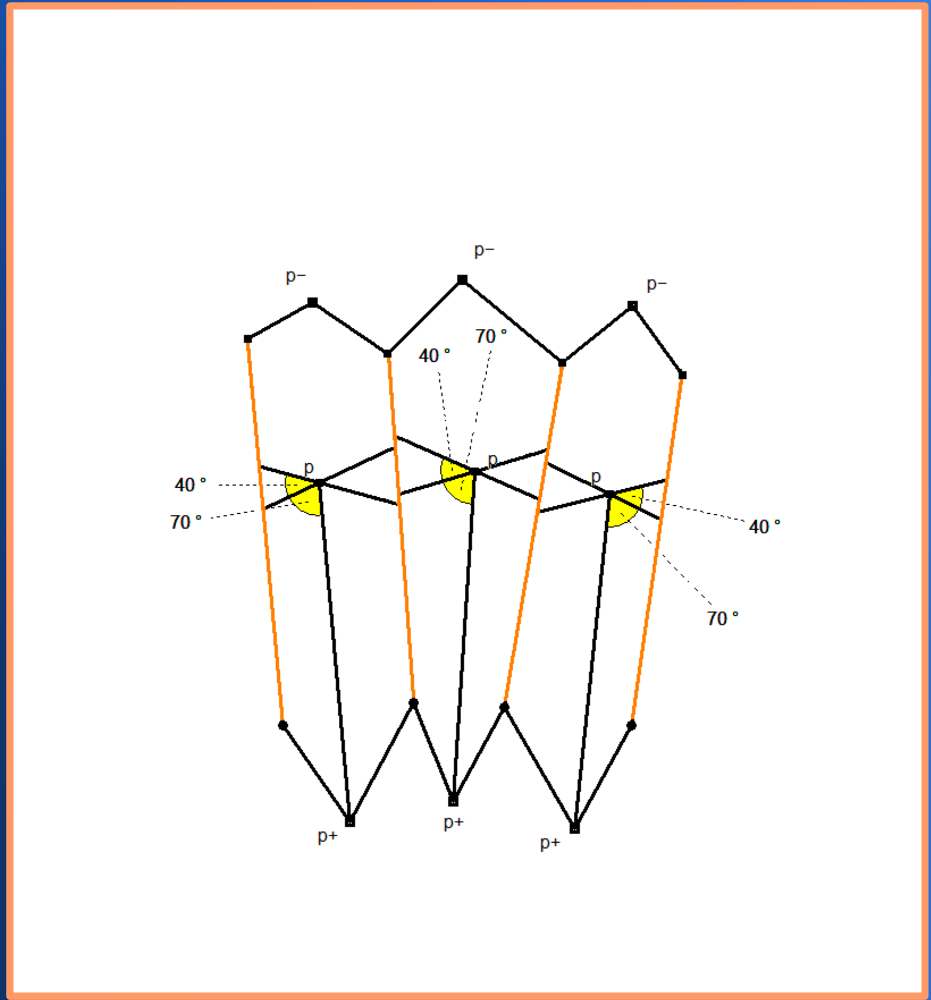
- Θ muss klein genug sein, maximal $\leq \pi/8 = 22.5°$

- Es gilt $\Psi = \pi/2 - \Theta$
- für $\Theta = \pi/8$ ist $\Psi = 3\pi/8$
- für $\Theta = 20°$ ist $\Psi = 70°$



Cocone Algorithmus

- Abtastung
- Delaunay Triangulation
- Voronoi Diagramm
- Pole
- Geschätzte Normale
- Cocone
- Kanten markieren
- Manifold extrahieren



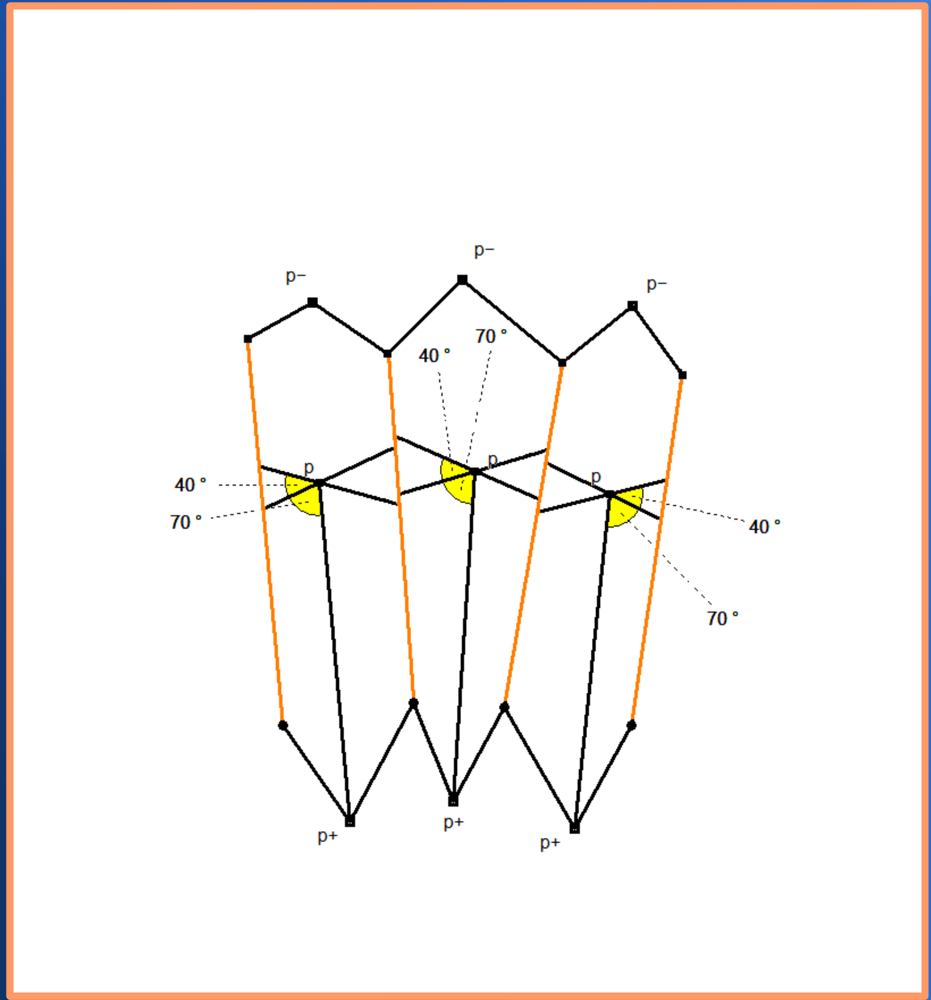
Cocone Algorithmus

- Kanten markieren

  alle Kanten, die Cocone schneiden werden markiert

- Gesucht sind:

  Punkte x ϶ V(P), der Winkel zwischen x' und gn(p) ist im Bereich von A=(π/2−Θ bis π/2+Θ)

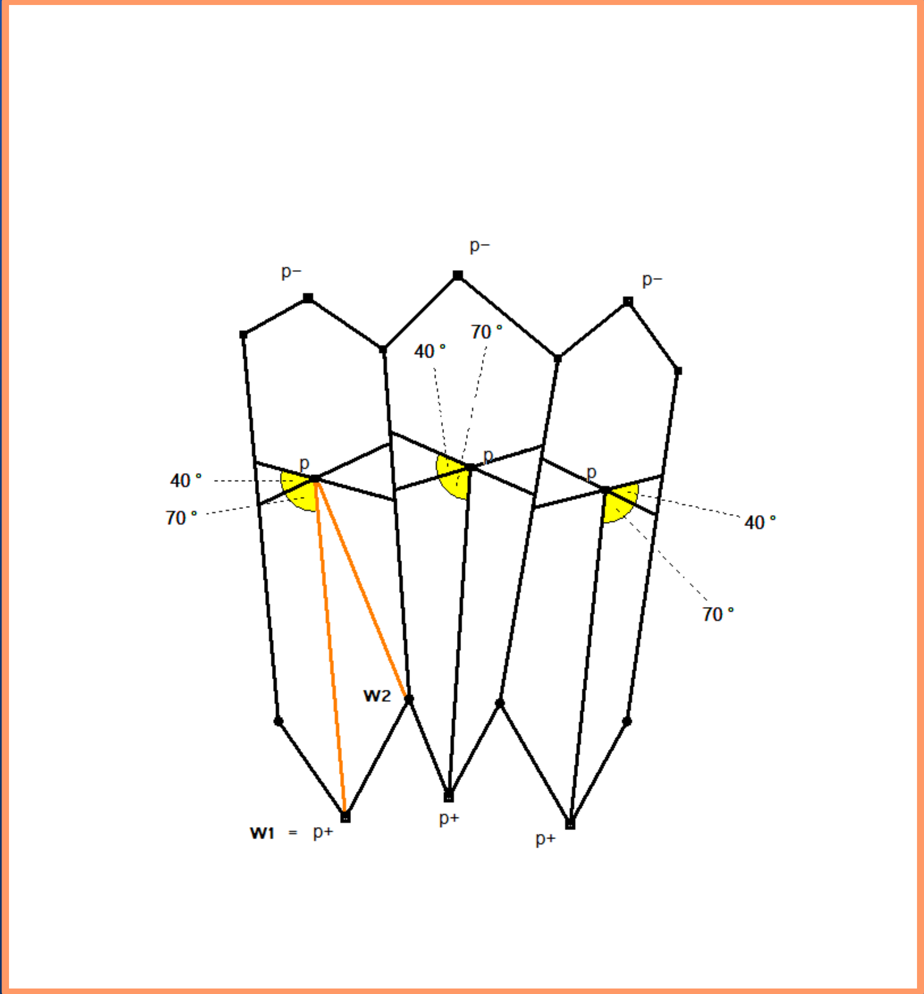


Cocone Algorithmus

- Berechnung:

  Input: 2 Eckpunkte （w1, w2）der Voronoi Kante e die getestet werden soll,

  die Winkel zwischen w1',w2' und gn（p） bilden den Bereich I2



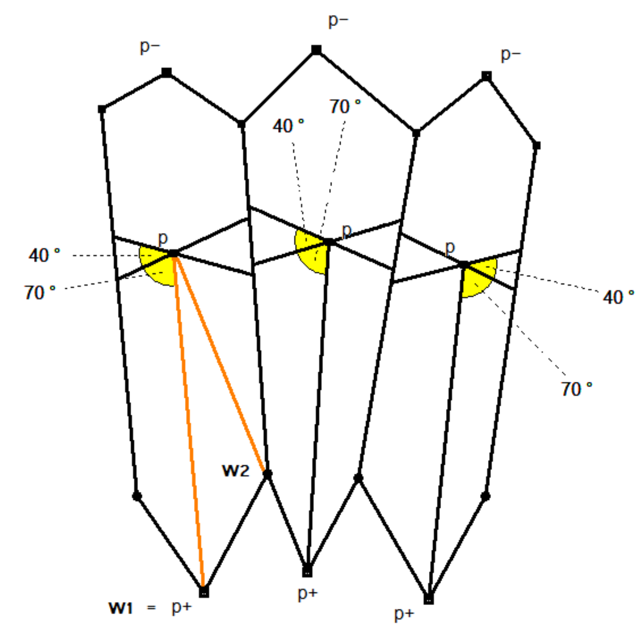
Cocone Algorithmus

- Berechnung:

  Input: 2 Eckpunkte （w1, w2）der Voronoi Kante e die getestet werden soll,

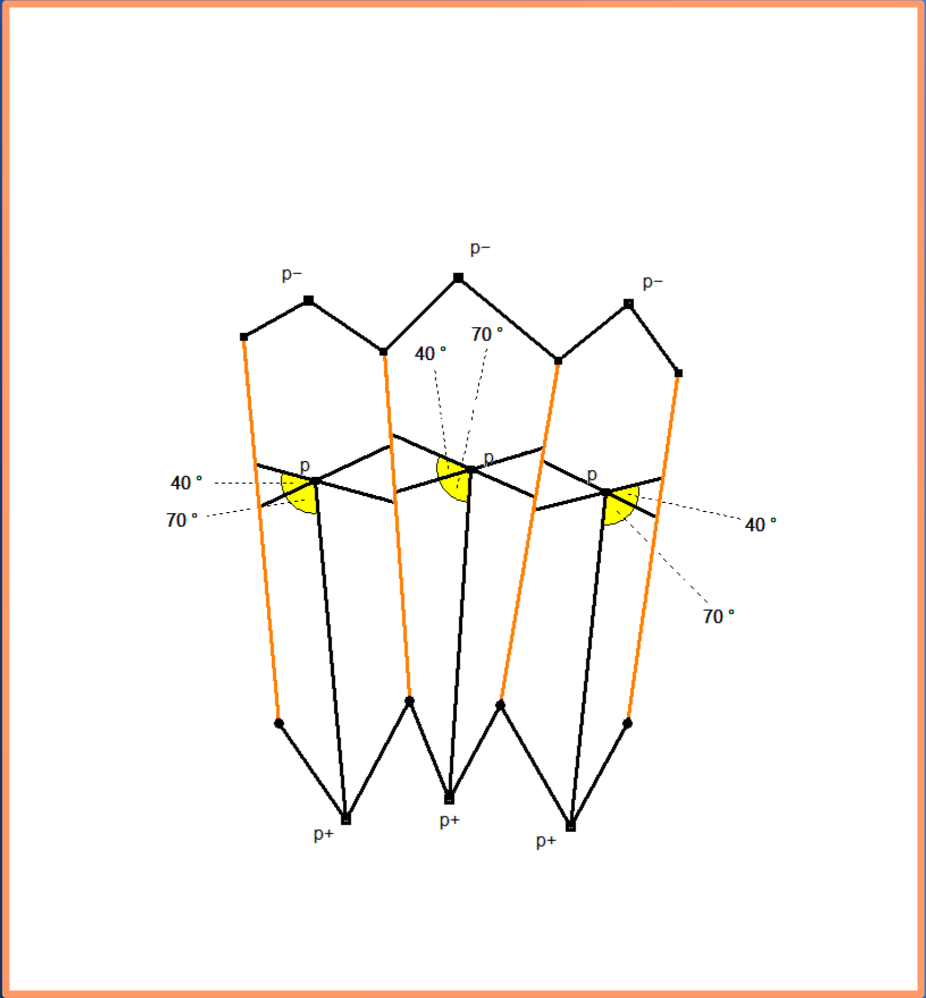
  die Winkel zwischen w1',w2' und gn（p） bilden den Bereich I2

  Markieren bei Überschneidung von I und I2



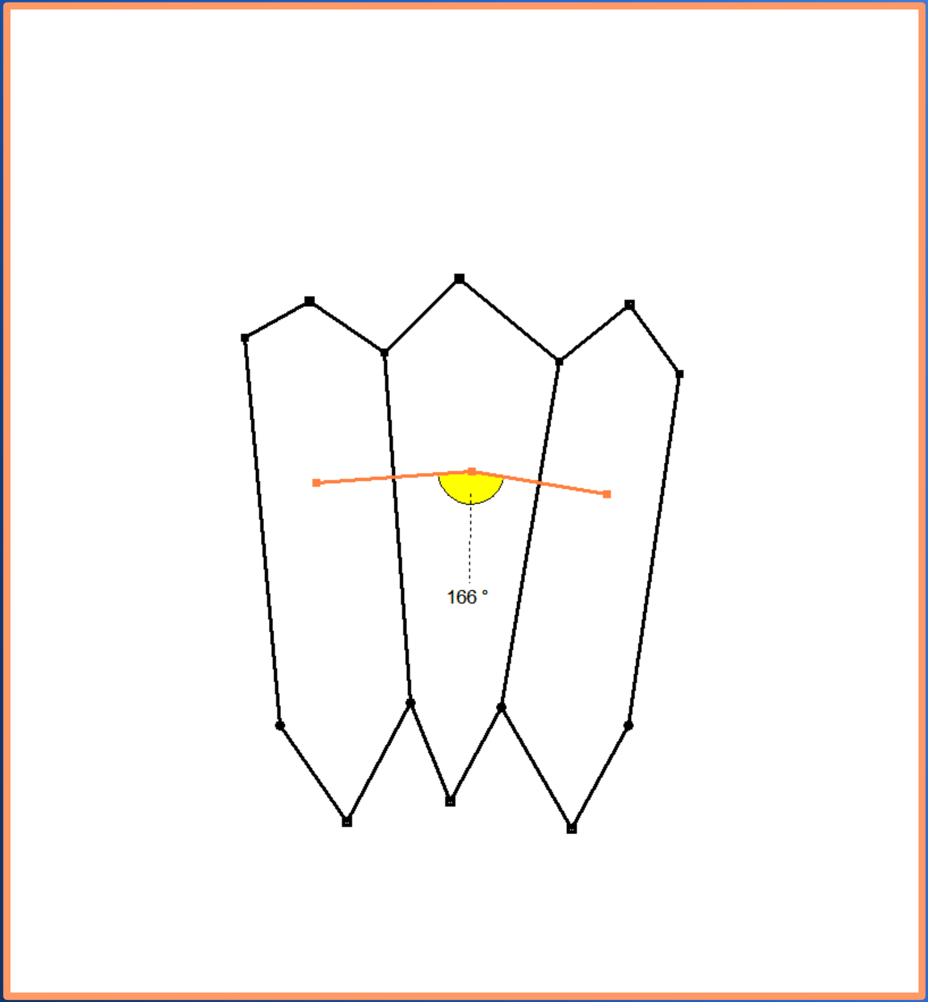
Cocone Algorithmus

- Eine Kante e die von allen Zellen markiert wurde, die sie beinhalten wird in die Kantenmenge E eingefügt
- Die zu E duale Dreiecke bilden die Menge der Kandidatendreiecke T



Cocone Algorithmus

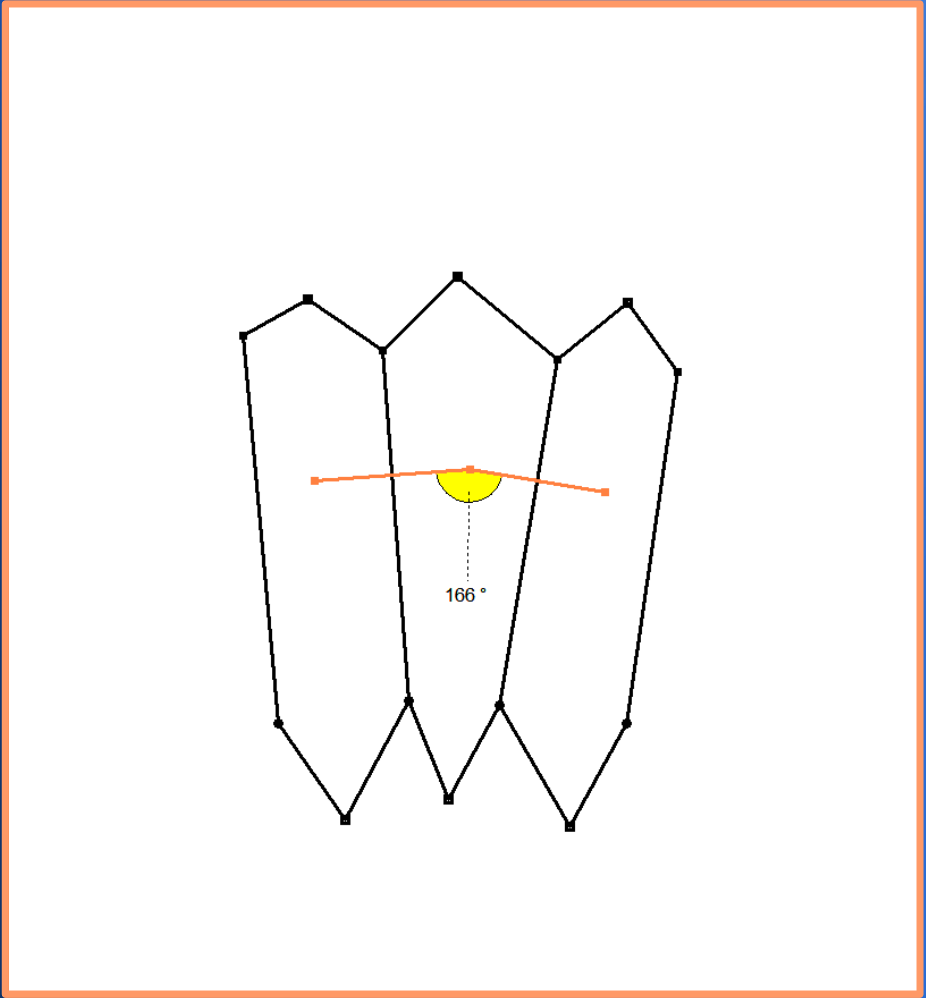
- Abtastung
- Delaunay Triangulation
- Voronoi Diagramm
- Pole
- Geschätzte Normale
- Cocone
- Kanten markieren
- Manifold extrahieren



Cocone Algorithmus

- Manifold N:

  komplexe Objekte,
  allgemein x−manifold,
  x ist die Dimension,
  3−manifold = 3D−manifold



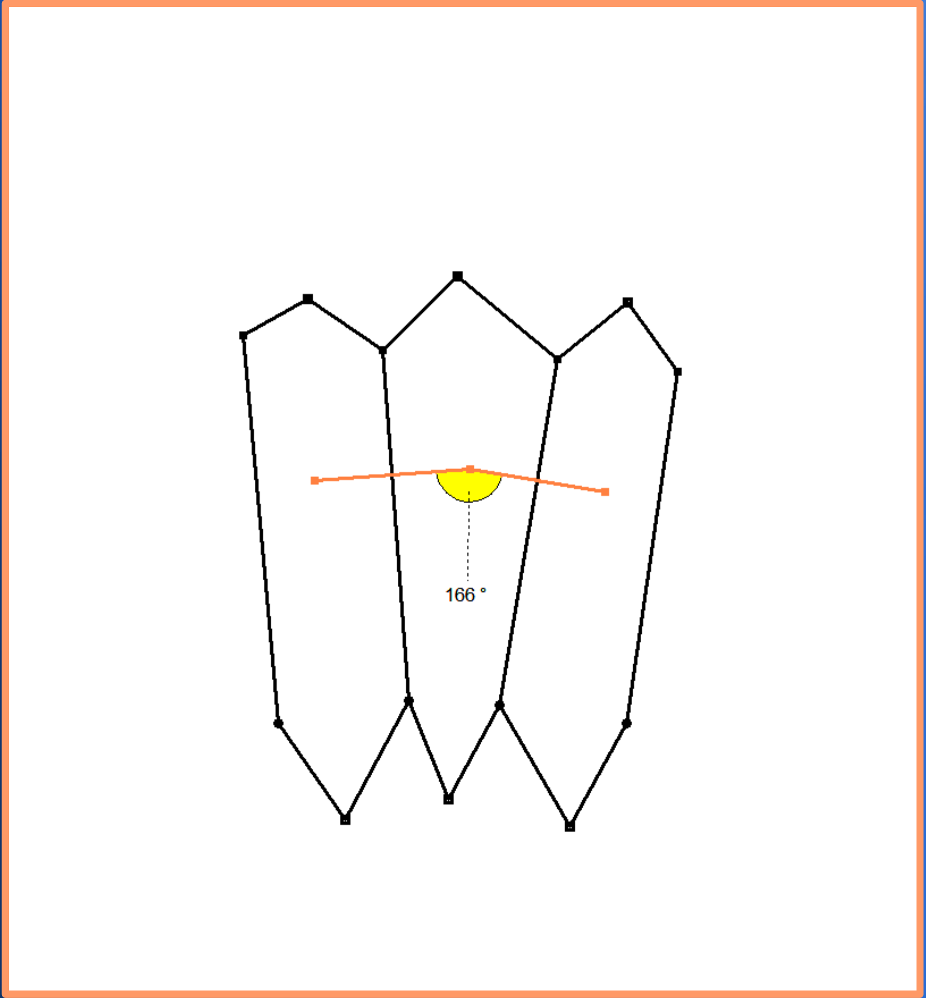
Cocone Algorithmus

- Manifold N:

  komplexe Objekte,
  allgemein x−manifold,
  x ist die Dimension,
  3−manifold = 3D−manifold

- jeder Punkt hat eine ganz bestimmte Nachbarschaft

  1−manifold (Linie, Kreis,...)
  2−manifold (Disk, Sphäre, Ebene Fläche, ...)



Cocone Algorithmus

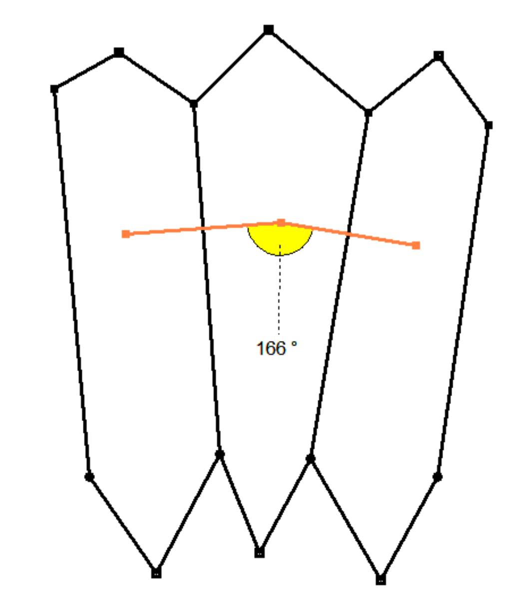
- Manifold N Extraktion:

  N wird aus T extrahiert , Dreiecke an scharfen Kanten werden gelöscht

- Scharfe Kanten:

  der Winkel zwischen 2 einfallenden, aufeinanderfolgenden Deiecken ist $\leq 3\pi/2$

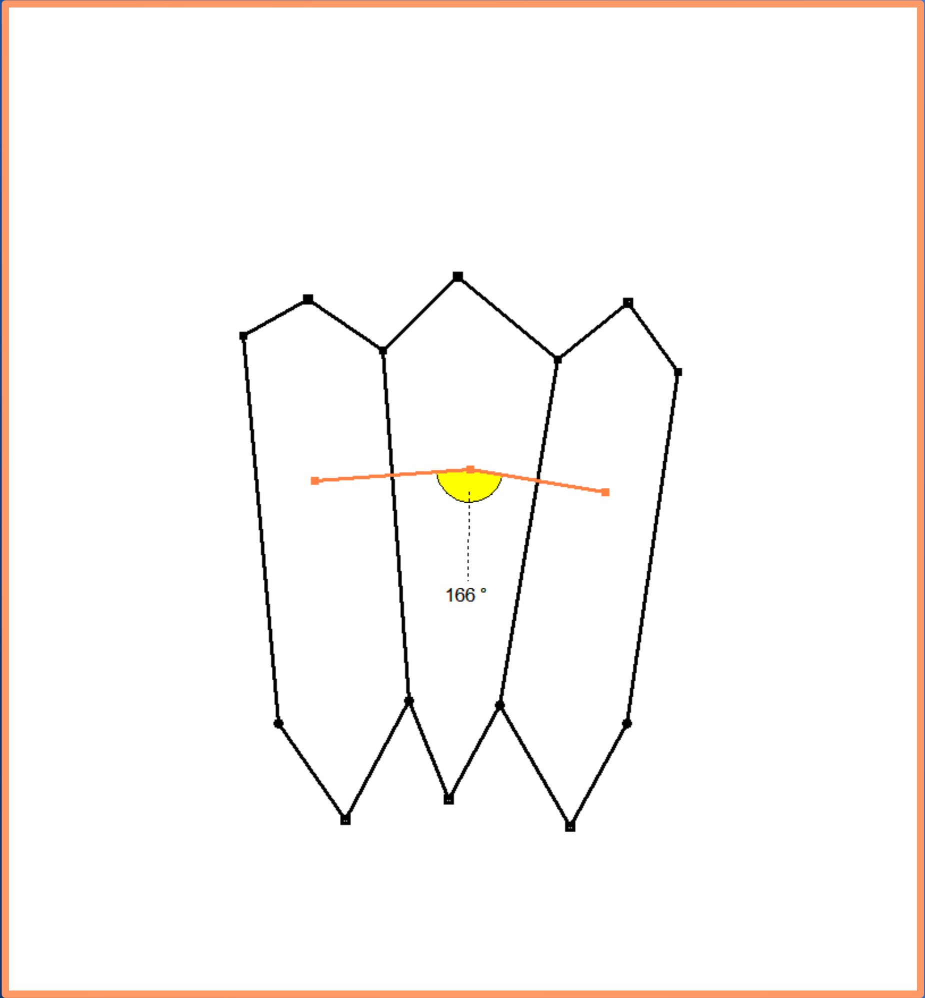
  nur 1 Dreieck einfallend



Cocone Algorithmus

- Manifold N Extraktion:

  Löschen ist sicher, weil alle Dreiecke aus D(P,S) diese Bedingung nicht erfüllen



Cocone Algorithmus

- Manifold N Extraktion:

  Tiefendurchgang über D(P)

  das 2-manifold N wird mit Hilfe von der Kandidatenmenge T extrahiert