

Algorithmus zur Bestimmung der Teilmengen beliebiger Polygone

Aus

$$A \cap B \cap C \cap D = ((A \cap B) \cap C) \cap D$$

ergibt sich, dass auch bei der Berechnung für n-Polygone zunächst der Schnitt von zwei der Polygone betrachtet werden kann. Ist der Schnitt aus A und B ermittelbar, so lässt sich durch die oben dargestellte Weise jede Zahl an Polygonen schneiden.

Näher betrachten wir den Schnitt der Polygone A und B .

Man wähle aus A einen Knoten k_0 auf der Kante K_0 . Wir iterieren solange durch die Kanten von A bis wir einen Schnittpunkt mit einer Kante aus B finden, oder wieder k_0 betrachten. Dies wäre in drei Fällen gegeben:

1. Polygon A liegt in $B \rightarrow A \subsetneq B$
2. Polygon B liegt in $A \rightarrow B \subsetneq A$
3. Die Polygone sind disjunkt

Mittels des Punkt-in-Polygon-Tests nach Jordan prüfen wir ob der Knoten k_0 des jeweiligen Polygons und somit das gesamte Polygon, innerhalb des jeweils anderen liegt. Die gemeinsame Schnittfläche ist dann das jeweils umschlossene Polygon und wird zurückgegeben.

Ermitteln von Schnittpunkten zweier Kanten

Ob eine Kante K_a eine andere Kante K_b schneidet wird mittels linearer Vektorgleichungen bestimmt. Hierbei geht eine Prüfung der Diskriminanten des sich aus der Schnittgleichung ergebenden Matrix voraus, um auszuschließen, dass die Kanten kollinear sind.

Aus der Modellierung ergibt sich als weitere Forderung an die Schnittpunkte, dass diese auf beide Kanten liegen müssen (nicht auf der verlängerten Geraden). Da sich der Richtungsvektor der jeweiligen Kante aus den begrenzenden Knoten der Kante bestimmt wird muss gelten:

$$0 < \lambda_a < 1 \ \&\& \ 0 < \lambda_b < 1$$

Hierdurch wird sichergestellt, dass der Schnittpunkt sowohl innerhalb der Kanten liegen muss und keiner der begrenzenden Punkte ist.

Wird der erste Schnittpunkt gefunden, wird mittels des Punkt-in-Polygon-Tests nach Jordan geprüft, ob der erste begrenzende Knoten der Kante aus A innerhalb des Polygons B liegt oder außerhalb. Liegt dieser außerhalb so wird die Kante, nach den oben beschriebenen Regeln, weiterverfolgt und der Schnittpunkt wird Liste der Schnittpolygon-Knoten als erster hinzugefügt.

Liegt dieser innerhalb, wird der Schnittpunkt der Schnittpolygon-Knoten als erster hinzugefügt. Darüber hinaus gilt für diesen Schnittpunkt dasselbe, was für jeden weiteren Schnittpunkt von A und B gilt.

Wird ein Schnittpunkt gefunden

Wenn ein Schnittpunkt gefunden wurde ist dieser Knoten des Schnittpolygons. Um dieses Schnittpolygon zu umlaufen, muss ab hier die Teil-Kante der geschnittenen Kante aus B weiterverfolgt werden. Diese setzt sich aus dem Schnittpunkt und dem (sich im Polygon A befindenden) zweiten begrenzenden Knoten zusammen. (Unter Berücksichtigung der oben genannten Bedingungen für Schnittpunkte, kann somit kein Schnittpunkt zweimal in einem Durchlauf gefunden werden.)

Die zu betrachtende Kante (und somit das Polygon) muss gewechselt werden, da die Kante, auf welcher ich zu diesem Schnittpunkt gelaufen bin, unter den oben genannten Bedingungen, das Schnittgebiet beider Polygone (rechts) begrenzt und somit nach einem Schnitt mit einer Kante aus B dieser Kante die begrenzende Funktion nun zukommt, da sich die Kante aus A nun aus dem Schnittgebiet herausbewegt.

Man vertausche nun Polygon A und B und führe die oben genannten Schritte solange aus (mit dem Zusatz, dass jeder Knoten, an welchem ich meine zu betrachtende Kante wechsele in die Liste der Schnittpolygon-Knoten aufgenommen wird {Schnittpunkte und "normale" Knoten}), der erste Schnittpunkt der Liste der Schnittpolygon-Knoten erneut gefunden wurde.

Das Schnittpolygon wurde vollständig durchlaufen und alle Knoten befinden sich in korrekter Reihenfolge in der Liste der Schnittpolygon-Knoten.

Sollte Schnittpolygon vollständig durchlaufen sein, wird das Polygon A erneut durchlaufen unter Berücksichtigung, dass alle bereits bekannten Schnittpunkte ignoriert werden müssen. Erst wenn ein kompletter Durchlauf des Polygons A ohne erneutes Finden eines Schnittpunktes erfolgt, wurden alle Schnittpolygone von A und B gefunden.

Dieser Prozess kann mit den neuen Polygonen und den n weiteren Polygonen beliebig oft durchgeführt werden, sodass ein Schnitt aller Polygone (bei disjunkten keiner) zurückgeliefert wird.