Kollisionserkennung mit dem Gilbert-Johnson-Keerthi Algorithmus

Julian Dierkes

Gymnasium St. Michael Ahlen

Schuljahr 2014/2015

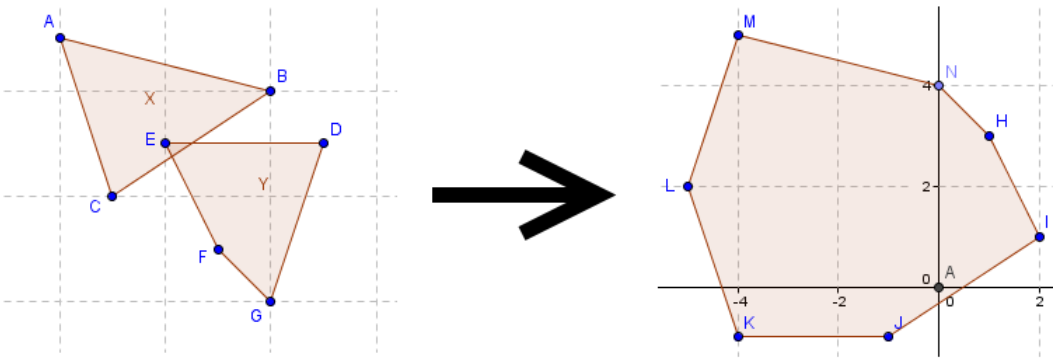
Inhaltsverzeichnis

1 Abstract

In dieser Arbeit stelle ich den GJK-Algorithmus vor, welcher zur effizienten Kollisionserkennung zwischen konvexen Menge verwendet werden kann. Dabei werden zunächst die mathematischen Grundlagen der konvexen Geometrie erläutert, um das mathematische Konzept, das hinter dem Algorithmus steht, besser verständlich zu machen. Danach findet ein Vergleich zwischen einer möglichen Brute-Force Methode und dem GJK-Algorithmus zur Lösung des Problems statt. Bei der Beschreibung des Algorithmus wird besonders auf den mathematischen Hintergrund des Algorithmus eingegangen und der Algorithmus schrittweise an einem Beispiel erläutert. Zur Unterstützung wird dem beigelegten Datenträger noch ein Programm hinzugefügt, dass die verschiedenen Arbeitsschritte des Algorithmus grafisch verdeutlicht.

2 Motivation

In der Informatik gibt es viele Einsatzgebiete, in denen echtzeitfähige Kollisionserkennungen benötigt werden. Zum Beispiel muss bei einem Fabrikroboter gewährleistet sein, dass dieser nicht mit einem Menschen kollidieren kann und diesen verletzen könnte. Eine weitere Verwendung finden moderne Algorithmen, um die echtzeitfähige Kollisionserkennung in physikalischen Simulation zu ermöglichen. Die Anforderungen an solche Algorithmen sind dabei sehr vielfältig und je nach Einsatzgebiet unterschiedlich. Ich bin durch die Entwicklung eines Computerspiels auf das Gebiet der Kollisionserkennung aufmerksam geworden. Am Anfang beschränkten sich meine Probleme noch auf die Erkennung einer Kollision zwischen zwei Kreisen oder Rechtecken, aber es stellte sich mir das Problem, eine Kollision zwischen zwei Asteroiden zu bestimmen, welche konvexe Körper waren. Nach einiger Recherche nach einer Lösung dieses Problems entdeckte ich schließlich den GJK-Algorithmus und schaffte es, mit diesem eine Kollision zwischen zwei konvexen Körpern zu bestimmen. Da mich die Einfachheit dieses Algorithmus begeistert hat, habe ich beschlossen, mich in meiner Projektarbeit näher mit diesem zu beschäftigen.



3 Konvexgeometrie

Der GJK-Algorithmus beschränkt sich auf die Kollisionserkennung zwischen konvexen Mengen. Deshalb werden in diesem Kapitel zum besseren Verständnis einige grundsätzliche Definitionen der konvexen Geometrie angegeben, vgl. hierzu auch [1] und [2].

3.1 Konvexe Menge

Sei $A \subset \mathbb{R}^n$ eine Menge und $x, y \in A$, dann bezeichnet:

$$[x, y] := \{\lambda x + (1 - \lambda)y;\ \lambda \in [0, 1]\}$$

die Strecke zwischen den Punkten x und y.

Definition 1 *Eine Menge $A \subset \mathbb{R}^n$ heißt konvex, wenn sie immer auch die Verbindungsstrecke zweier beliebiger Punkte der Menge enthält:*

$$x, y \in A \rightarrow [x, y] \subset A$$

Ein Beispiel für eine konvexe Menge ist eine Kugel. Nimmt man zwei beliebige Punkte aus einer Kugel, stellt man fest, dass immer auch die Verbindungsstrecke in der Kugel enthalten ist. Ein Gegenbeispiel ist ein Boomerang. Nimmt man die beiden Punkte der verschiedenen Enden des Boomerangs, ist die Verbindungsstrecke nicht in der Menge der Punkte des Boomerangs enthalten.

3.2 Konvexe Kombination

Definition 2 *Eine lineare Kombination $\lambda_1 x_1 + \cdots + \lambda_m x_m$ mit $\lambda_i \in \mathbb{R}$ wird als konvexe Kombination bezeichnet, wenn gilt:*

$$\lambda_i \geq 0 \ \textit{für alle } i \textit{ und } \lambda_1 + \cdots + \lambda_m = 1$$

Zum Beispiel ist jeder Punkt der Strecke $[x, y]$ eine konvexe Kombination der Punkte x und y.

3.3 Konvexe Hülle

Definition 3 *Die konvexe Hülle einer Menge $A \subset \mathbb{R}^n$ ist die Menge aller Punkte, die sich als konvexe Kombination von Punkten aus A ergeben:*

$$conv(A) := \{\lambda_1 x_1 + \cdots + \lambda_m x_m | x_1, ..., x_m \in A, \lambda_1, ..., \lambda_m \geq 0, \sum_{i=1}^{m} \lambda_i = 1, m \in \mathbb{N}\} \tag{1}$$

Aus Definition 3 geht hervor, dass sich jeder Punkt der konvexen Hülle einer Menge $A \subset \mathbb{R}^n$ als konvexe Kombination von Punkten aus A bestimmen lässt. Außerdem gilt für eine konvexe Hülle folgender Satz:

Satz 1 *Die konvexe Hülle einer Menge A ist die kleinste konvexe Menge, die A enthält.*

3.4 Konvexes Polytop

Definition 4 *Ein konvexes Polytop ist eine Teilmenge $A \subset \mathbb{R}^n$, die als konvexe Hülle endlich vieler Punkte dargestellt werden kann. D.h es gibt eine endliche Teilmenge $X \subset A$ mit $A = conv(X)$.*

Aus den Definitionen 3 und 4 lässt sich also folgender Satz herleiten:

Satz 2 *Die konvexe Hülle von einer endlichen Punktmenge $A \subset \mathbb{R}^n$ ist ein konvexes Polytop.*

3.5 Beispiele

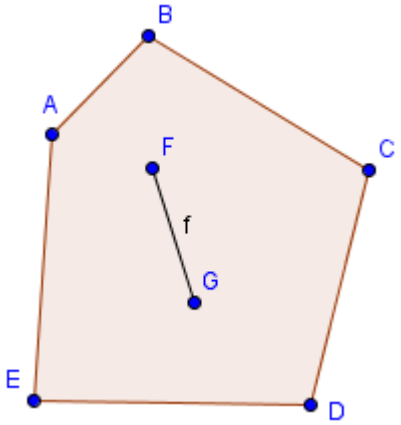


Abbildung 1: Beispiel einer konvexen Menge

In Abbildung 1 ist das Beispiel einer konvexen Menge zu sehen. Dies lässt sich mit Definition 1 belegen, denn die Menge enthält die Verbindungsstrecke zwischen beliebigen Punkten der Menge. In Abbildung 1 zum Beispiel die Verbindungsstrecke f zwischen F und G.

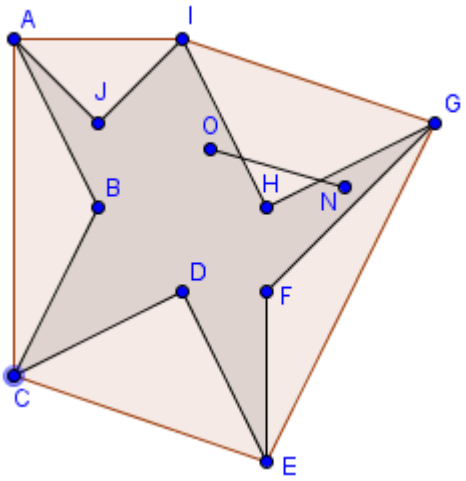


Abbildung 2: Beispiel einer nicht konvexen Menge mit konvexer Hülle

In Abbildung 2 ist eine nicht konvexe Menge zu sehen, die von einer konvexen Hülle eingeschlossen wird. Dass die Menge in Abbildung 2 nicht konvex ist, erschließt sich daraus, dass die Verbindungsstrecke der Punkte O und N nicht vollständig in der Menge enthalten ist.

4 Die Minkowski-Summe

Die Minkowski-Summe ist nach ihrem Entdecker Hermann Minkowski (*1864, +1909) benannt. Hermann Minkowski war ein deutscher Mathematiker und Physiker, dessen bedeutendste Entdeckung der Minkowski-Raum war, welcher eine Weiterentwicklung der Relativitätstheorie darstellt.

4.1 Berechnung der Minkowski-Summe

Definition 5 *Die Minkowski-Summe ist die Addition aller Punkte einer Menge $A \subset \mathbb{R}^n$ mit allen Punkten einer Menge $B \subset \mathbb{R}^n$:*

$$A + B := \{a + b \mid a \in A, b \in B\} \tag{2}$$

Die Minkowski-Summe funktioniert auch als Differenz, genannt Minkowski-Differenz.

Definition 6 *Die Minkowski-Differenz ist die Subtraktion aller Punkte einer Menge $A \subset \mathbb{R}^n$ mit allen Punkten einer Menge $B \subset \mathbb{R}^n$:*

$$A - B := \{a - b \mid a \in A, b \in B\} \tag{3}$$

Zur besseren Veranschaulichung der Minkowski Summe dient die folgende Grafik:

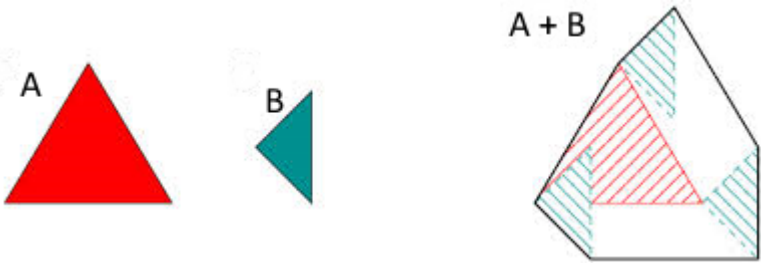


Abbildung 3: Menge A wird mit Menge B addiert. Der schwarze Umriss stellt die resultierende Minkowski-Summe dar.

Auf Abbildung 3 ist zu erkennen, wie alle Elemente der Menge B auf die Ecken der Menge A addiert werden. Dabei ergibt sich aus den entstehenden Punkten ein neue Menge, deren konvexe Hülle die Minkowski-Summe der Mengen A und B bildet. Außerdem gilt folgender Satz:

Satz 3 *Die Minkowski-Summe oder Differenz zweier konvexer Mengen ist wieder konvex.*

Die Minkowski-Summe von zwei konvexen Mengen muss also im $\mathbb{R}^n$ ein Polytop sein. Das ergibt sich daraus, dass sich die Minkowski-Summe aus zwei Mengen mit endlich vielen Eckpunkten zusammensetzt und daher auch nur endlich viele Eckpunkte enthalten kann. Außerdem gilt durch Satz 3, dass die Minkowski-Summe konvex ist. Folglich kann sie als konvexe Hülle endlich vieler Punkte dargestellt werden, was nach Satz 2 einem konvexen Polytop entspricht.

4.2 Kollisionserkennung mit Hilfe der Minkowski-Differenz

Wenn zwei Mengen kollidieren, bedeutet das, dass sich einige Punkte der Menge A mit einigen Punkten der Menge B überschneiden. Durch Anwenden der Minkowski-Differenz auf diese beiden Mengen werden alle Punkte der Menge A von den Punkten der Menge B subtrahiert. Also werden auch die Punkte voneinander subtrahiert, die sich überschneiden und somit gleich sind. Eine Subtraktion zwischen zwei gleichen Punkten ergibt null und daher muss die Minkowski-Differenz den Ursprung beinhalten. Es lässt sich also folgender Satz aufstellen:

Satz 4 *Sind $B, C \subset \mathbb{R}^n$ konvexe Mengen, so gilt: B und C überschneiden sich (also $B \cap C \neq \varnothing$) genau dann, wenn für die Minkowski-Differenz $A = B - C$ gilt: $0 \in A$, d.h. A enthält den Ursprung.*

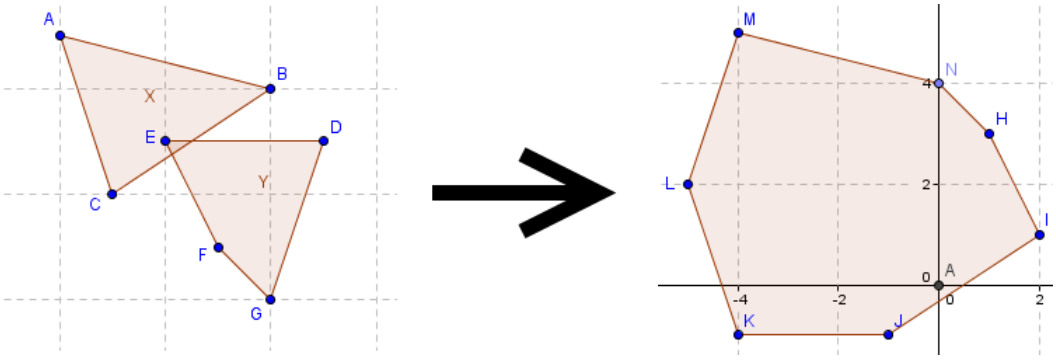


Abbildung 4: Ein Beispiel für die Minkowski-Differenz, bei der eine Kollision vorliegt und die Ergebnismenge den Ursprung beinhaltet

Ein Algorithmus, der überprüft, ob zwei konvexe Mengen kollidieren, muss also nur prüfen, ob der Ursprung in der Minkowski-Differenz, also in dem Polytop, enthalten ist.

Carathéodory hat einen bedeutenden Satz formuliert und bewiesen, der Folgendes besagt:

Satz 5 *(Carathéodory) Sei $A \subset \mathbb{R}^n, A \neq \varnothing$. Dann gilt: jedes $x \in conv(A)$ läßt sich als Konvexkombination von höchstens (n+1) Elementen aus A darstellen.*

Für eine konvexe Hülle $A \subset \mathbb{R}^2$ gilt folglich, dass jeder Punkt der konvexen Hülle in einem Simplex mit 3 Ecken in A liegt. Um zwei Mengen auf Kollision zu überprüfen, reicht es demnach, ein Dreieck mit den Eckpunkten der Menge der Minkowski-Differenz zu bilden, bei dem sich der Ursprung innerhalb dieses Dreiecks befindet.

Da hier nur konvexe Mengen im $\mathbb{R}^2$ auf Kollision überprüft werden, finden die folgenden Betrachtungen im $\mathbb{R}^2$ statt.

5 Rechnungen am Simplex

Um zu berechnen, ob der Ursprung innerhalb eines Dreiecks liegt, fasst man die Seiten als „Teile“ von Geraden (Strecken) durch die Eckpunkte der Seiten auf. Eine solche Geraden im $\mathbb{R}^2$ lässt sich in der Normalenform

$$g : (\vec{x} - \vec{p}) \cdot \vec{n} = 0$$

schreiben. Dabei ist $\vec{x}$ Ortsvektor eines beliebigen Punktes im $\mathbb{R}^2$, $\vec{p} = \begin{pmatrix} p_1 \\ p_2 \end{pmatrix}$ Ortsvektor eines (Auf-)Punktes der Geraden und $\vec{n} = \begin{pmatrix} n_1 \\ n_2 \end{pmatrix}$ bezeichne einen Normalenvektor.

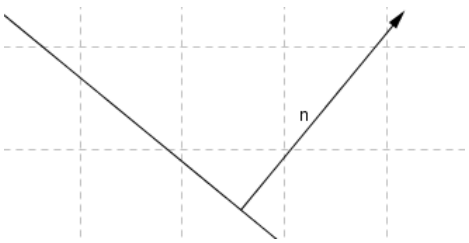


Abbildung 5: Beispiel einer Geraden mit Normalenvektor

Die Gerade g zerlegt die Ebene in zwei Halbebenen. Bezeichnet

$$\vec{n_0} = \frac{\vec{n}}{|\vec{n}|} = \frac{1}{\sqrt{n_1^2 + n_2^2}} \begin{pmatrix} n_1 \\ n_2 \end{pmatrix}$$

den Normaleneinheitsvektor von $\vec{n}$, der so orientiert ist, dass er in die Halbebene zeigt, die nicht den Ursprung enthält, so gilt nach der Hesseschen Normalenform:

Satz 6 *Ein Punkt Q mit Ortsvektor* $\vec{q} = \begin{pmatrix} q_1 \\ q_2 \end{pmatrix}$ *hat von der Geraden g den mit Vorzeichen versehenen Abstand*

$$d := d(Q) = (\vec{q} - \vec{p}) \cdot \vec{n_0} = \frac{q_1 n_1 + q_2 n_2 + c}{\sqrt{n_1^2 + n_2^2}}$$

Dabei ist $c = -(p_1 n_1 + p_2 n_2) < 0$.

Ein Punkt Q, der die Gleichung $d(Q) = 0$ erfüllt, liegt damit auf der Geraden. Ist Q aber kein Punkt der Geraden, so gilt $d \neq 0$ und:

- Ist $d > 0$, so liegen der Ursprung O und Q in verschiedenen Halbebenen.
- Ist $d < 0$, so liegen der Ursprung O und Q in derselben Halbebene.

Da $\sqrt{n_1^2 + n_2^2} > 0$, folgt also für jeden Punkt Q, der nicht auf der Geraden liegt:

- Ist $(\vec{q} - \vec{p}) \cdot \vec{n} > 0$, so liegt Q auf der entgegengesetzten Seite des Ursprungs und damit auf der Seite, in die der Normalenvektor zeigt.
- Ist $d < 0$, so liegt Q auf der Seite des Ursprungs, also auf der Seite, die in der entgegengesetzten Richtung des Normalenvektors liegt.

Ist g eine Gerade durch zwei Punkte A und B und $(\vec{b} - \vec{a}) = \begin{pmatrix} x \\ y \end{pmatrix}$, so ist $\vec{n} = \begin{pmatrix} y \\ -x \end{pmatrix}$ der Normalenvektor, der so orientiert ist, dass er in die Halbebene zeigt, die nicht den Ursprung enthält und somit der für Satz 6 richtige Normalenvektor. Um den in Satz 6 - im folgenden häufig gebrauchten - Sachverhalt einfacher ausdrücken zu können, führe ich folgende Sprechweise an:

Definition 7 *Ein Punkt A einer Geraden g passiert einen Punkt B in Richtung des Normalenvektors $\vec{n}$ der Geraden g, wenn gilt*

$$(\vec{b} - \vec{a}) \cdot \vec{n} < 0$$

dabei bezeichnen $\vec{a}$ und $\vec{b}$ die Ortsvektoren der Punkte A und B.

Ist insbesondere B der Ursprung O, so folgt damit aus der Gleichung $-\vec{a} \cdot \vec{n} < 0$, dass der Punkt A einer Geraden g den Ursprung in Richtung des Normalenvektors $\vec{n}$ passiert.

Damit ist klar, dass folgender Satz gilt:

Satz 7 *Es bezeichnen $\vec{a}, \vec{b}$ und $\vec{c}$ die Ortsvektoren der Punkte A, B, C eines Dreiecks und $\vec{n}_{AB}$, $\vec{n}_{BC}$, $\vec{n}_{CA}$ die Normalenvektoren auf den Seiten, die nach außen zeigen. Dann folgt, dass der Ursprung im Dreieck liegt, wenn gilt:*

$$-\vec{a} \cdot \vec{n}_{AB} < 0 \ \wedge \ -\vec{b} \cdot \vec{n}_{BC} < 0 \ \wedge \ -\vec{c} \cdot \vec{n}_{CA} < 0 \qquad (4)$$

Denn alle Ecken des Dreiecks passieren dann den Ursprung.

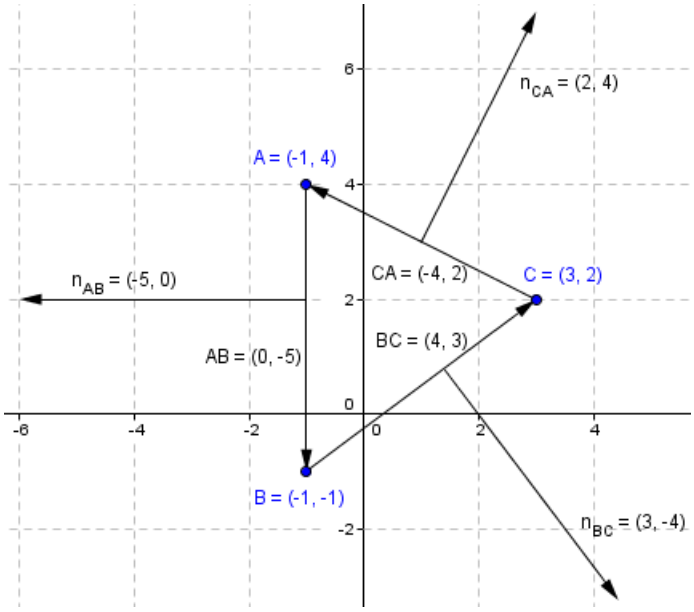


Abbildung 6: Beispiel eines Dreiecks mit Normalenvektoren, die nach außen zeigen.

Dieser Satz wird nun einmal an dem Beispiel aus Abbildung 6 überprüft. In Abbildung 6 sind die Normalenvektoren zu den drei Seiten des Dreiecks angegeben. Auf diese Normalenvektoren können nun die Rechnungen 4 angewendet werden, um zu überprüfen, ob sich der Ursprung innerhalb des Dreiecks befindet:

$$-\vec{a} \cdot \vec{n}_{AB} = \begin{pmatrix} 1 \\ -4 \end{pmatrix} \cdot \begin{pmatrix} -5 \\ 0 \end{pmatrix} = -5 < 0 \tag{5}$$

$$-\vec{b} \cdot \vec{n}_{BC} = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \cdot \begin{pmatrix} 3 \\ -4 \end{pmatrix} = -1 < 0 \tag{6}$$

$$-\vec{c} \cdot \vec{n}_{CA} = \begin{pmatrix} -3 \\ -2 \end{pmatrix} \cdot \begin{pmatrix} 2 \\ 4 \end{pmatrix} = -14 < 0 \tag{7}$$

Alle Ergebnisse sind kleiner als null und damit ist bewiesen, dass der Ursprung sich innerhalb des Dreiecks aus Abbildung 6 befindet.

6 Erkennung einer Kollision per Brute-Force

Eine mögliche Brute-Force Methode, um eine Kollision zwischen zwei konvexen Mengen zu überprüfen, könnte für das untenstehende Beispiel wie folgt aussehen:

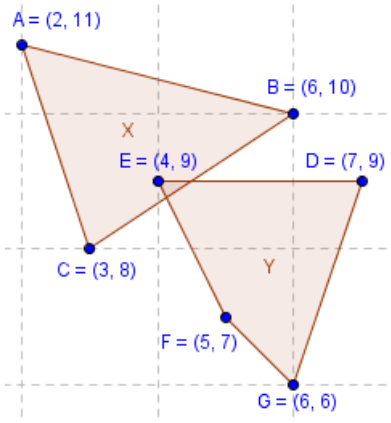


Abbildung 7

Zuerst werden alle 12 Differenzpunkte der beiden Mengen X und Y in Abbildung 7 ausgerechnet. Dazu wird jede Ecke der Menge X mit jeder Ecke der Menge Y subtrahiert. Für den Punkt A der Menge X bedeutet das:

$$H = A - D = (2|11) - (7|9) = (-5|2) \tag{8}$$
$$I = A - E = (2|11) - (4|9) = (-2|2) \tag{9}$$
$$J = A - F = (2|11) - (5|7) = (-3|4) \tag{10}$$
$$K = A - G = (2|11) - (6|6) = (-4|5) \tag{11}$$

Nach dem selben Verfahren müssen noch die Differenzpunkte für B und C der Menge X ausgerechnet werden. Wenn dies geschehen ist, ergibt sich folgende Menge an Punkten:

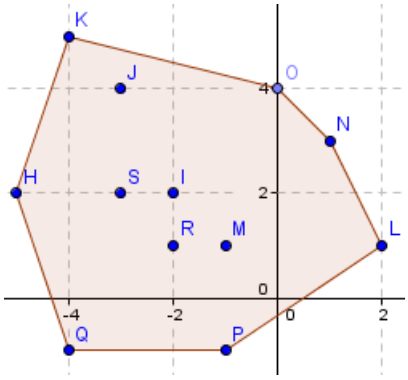


Abbildung 8

Die Minkowski-Differenz ist die konvexe Hülle der Menge in Abbildung 8. Um festzustellen, ob der Ursprung in dieser Menge enthalten ist, werden alle Dreiecke, die aus je 3 Punkten der 12 Differenzpunkte gebildet werden können, darauf überprüft, ob sie den Ursprung beinhalten. Wenn dies der Fall ist und eines der Dreiecke den Ursprung beinhaltet, ist eine Kollision vorhanden.

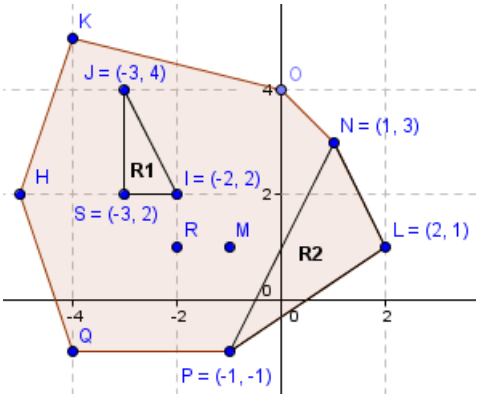


Abbildung 9: Beispiel von zwei Dreiecken, die durch jeweils 3 der 12 Differenzpunkte gebildet wurden

In Abbildung 9 sind zwei Beispiele von Dreiecken zu sehen, die durch 3 der 12 Differenzpunkte gebildet wurden. Das Dreieck R1 enthält den Ursprung nicht und somit kann keine Aussage gemacht werden, ob eine Kollision vorliegt. Das Dreieck R2 enthält den Ursprung und somit ist bewiesen, dass die beiden Ausgangsmengen X und Y kollidieren.

Damit eine Aussage getroffen werden kann, ob eine Kollision vorhanden ist, muss also ein mathematisches Verfahren vorliegen, mit dem ausgerechnet werden kann, ob der Ursprung sich innerhalb eines Dreiecks befindet. Einen offensichtlichen Lösungsansatz bietet hierzu folgender Satz:

Satz 8 *Ein Punkt liegt genau dann innerhalb eines Dreiecks, wenn der Punkt als konvexe Kombination der Dreieckspunkte A,B und C dargestellt werden kann.*

Dieser Satz bedeutet für uns, dass versucht werden kann, eine konvexe Kombination aus den Dreieckspunkten der jeweiligen Dreiecke zum Ursprung zu bilden. Wenn für diese konvexe Kombination gilt, dass $\lambda_1, \lambda_2, \lambda_3 \geq 0$ ist, dann ist bewiesen, dass der Ursprung im überprüften Dreieck liegt. Zu diesem Zweck löst man ein Gleichungssystem mit drei Gleichungen und drei Unbekannten, wobei die dritte Gleichung die Bedingung

$$\lambda_1 + \lambda_2 + \lambda_3 = 1$$

an der konvexen Kombination sein muss. Für das Dreieck R1 in Abbildung 9 erhält man mit λ_1, λ_2, $\lambda_3 \in \mathbb{R}$ und den Punkten I, J und S das lineare Gleichungssystem:

$$\begin{aligned} 0 = &\quad -3\lambda_1 - 2\lambda_2 - 3\lambda_3 \\ 0 = &\quad 4\lambda_1 + 2\lambda_2 + 2\lambda_3 \\ 1 = &\quad \lambda_1 + \lambda_2 + \lambda_3 \end{aligned} \tag{12}$$

Dieses kann nun aufgelöst werden und es folgt:

$$\begin{aligned} \lambda_1 = &\quad -1 \\ \lambda_2 = &\quad 3 \\ \lambda_3 = &\quad -1 \end{aligned} \tag{13}$$

λ_1 und λ_3 sind kleiner als null, daher kann der Ursprung nicht innerhalb des Dreiecks R1 liegen.

Eine nicht so offensichtliche Möglichkeit, um zu überprüfen, ob der Ursprung innerhalb des Dreiecks liegt, wurde bereits in Kapitel 5 vorgestellt. Das Verfahren aus Kapitel 5 ist deutlich schneller als die Überprüfung über die konvexe Kombination, da keine Gleichungssysteme aufgelöst werden müssen.

Wir wenden das Verfahren aus Kapitel 5 einmal auf das Dreieck R2 in Abbildung 9 an. Die Verbindungsvektoren zwischen den Ecken des Dreiecks sind gegeben durch:

$$\vec{NP} = \begin{pmatrix} -2 \\ -4 \end{pmatrix} \ \wedge \ \vec{PL} = \begin{pmatrix} 3 \\ 2 \end{pmatrix} \ \wedge \ \vec{LN} = \begin{pmatrix} -1 \\ 2 \end{pmatrix} \tag{14}$$

Aus diesen Verbindungsvektoren ergeben sich die Normalenvektoren der Dreiecksseiten, welche vom Simplex wegzeigen:

$$\vec{n}_{NP} = \begin{pmatrix} -4 \\ 2 \end{pmatrix} \ \wedge \ \vec{n}_{PL} = \begin{pmatrix} 2 \\ -3 \end{pmatrix} \ \wedge \ \vec{n}_{LN} = \begin{pmatrix} 2 \\ 1 \end{pmatrix} \tag{15}$$

Durch Satz 7 kann schließlich mit den Normalenvektoren überprüft werden, ob der Ursprung innerhalb des Dreiecks liegt:

$$-\vec{n} \cdot \vec{n}_{NP} = \begin{pmatrix} -1 \\ -3 \end{pmatrix} \cdot \begin{pmatrix} -4 \\ 2 \end{pmatrix} = -2 < 0 \tag{16}$$

$$-\vec{p} \cdot \vec{n}_{PL} = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \cdot \begin{pmatrix} 2 \\ -3 \end{pmatrix} = -1 < 0 \tag{17}$$

$$-\vec{l} \cdot \vec{n}_{LN} = \begin{pmatrix} -2 \\ -1 \end{pmatrix} \cdot \begin{pmatrix} 2 \\ 1 \end{pmatrix} = -5 < 0 \tag{18}$$

Alle Skalare sind kleiner als null und damit ist bewiesen, dass der Ursprung sich innerhalb des Simplex befindet.

Die vorgestellte Brute-Force Methode zur Überprüfung einer Kollision zwischen zwei konvexen Polygonen hat große Nachteile. Viele Dreiecke, die zwischen den Differenzpunkten der Ausgangsmengen gebildet werden, überschneiden sich und somit wird beim Test, ob der Ursprung in diesen enthalten ist, viel Rechenarbeit doppelt verrichtet. Außerdem kann es bei Mengen mit sehr vielen Differenzpunkten, wie zum Beispiel einem Kreis, passieren, dass eine sehr große Anzahl an Dreiecken gebildet werden muss. Dies kann es dem Computer unmöglich machen, in einer akzeptablen Zeit zu testen, ob der Ursprung in einem Dreieck enthalten ist. Ein deutlich effizienteres Verfahren stellt der GJK-Algorithmus dar, welcher im folgenden Kapitel erläutert wird.

7 Der GJK-Algorithmus

Der GJK-Algorithmus wurde von den Wissenschaftlern Gilbert, Johnson und Keerthi entwickelt und diente ursprünglich dazu, den Abstand zwischen zwei konvexen Mengen zu bestimmen. Ich beziehe mich hierbei auf die Ausarbeitungen von William Bittle [3].

7.1 Ablauf des GJK-Algorithmus

Der Algorithmus versucht, das größte Problem der vorgestellten Brute-Force Methode (Kapitel 6) zu umgehen. Anstatt alle möglichen Dreiecke der Differenzpunkte zu bilden und auf den Ursprung zu überprüfen, wird gezielt versucht, ein einzelnes Simplex innerhalb der Minkowski-Differenz um den Ursprung zu erstellen. Liegt der Ursprung nicht innerhalb des gebildeten Simplex, wird ein neues Dreieck in Richtung des Ursprungs vom alten Simplex gebildet. Dies wird so lange wiederholt, bis ein Simplex gefunden ist, welches den Ursprung enthält, oder bewiesen ist, dass der Ursprung nicht in der Minkowski-Differenz liegen kann.

Um ein Dreieck innerhalb der Minkowski-Differenz zu bilden, muss nicht die gesamte Minkowski-Differenz berechnet werden, sondern es genügt, einzelne Ecken dieser berechnen zu können.

Vorgehensweise 1 *Um eine Ecke P in eine Richtung $\vec{r}$ einer Minkowski-Differenz aus zwei konvexen Mengen $A \subset \mathbb{R}^2$ und $B \subset \mathbb{R}^2$ zu berechnen, wird die Ecke der konvexen Mengen A berechnet, die am weitesten in Richtung $\vec{r}$ liegt und die Ecke, die am weitesten in Richtung $-\vec{r}$ von der Menge B liegt. Durch Subtraktion der Ecken ergibt sich die Ecke der Minkowski-Differenz. Bezeichnen also $\vec{a}_i$ bzw. $\vec{b}_i$ die Ortsvektoren der Eckpunkte der Mengen A und B, so findet man ein s und ein t mit:*

$$\vec{a}_s \cdot \vec{r} = max\{\vec{a_i} \cdot \vec{r}; i = 0...m\} \quad (0 \leq s \leq m) \tag{19}$$

$$\vec{b}_t \cdot (-\vec{r}) = max\{\vec{b_i} \cdot (-\vec{r}); i = 0...n\} \quad (0 \leq t \leq n) \tag{20}$$

$$und\ setzt:\ \vec{OP} = \vec{a}_s - \vec{b}_t \tag{21}$$

Für das Beispiel in Abbildung 10 berechnen wir einen Ausgangspunkt für das Simplex innerhalb der Minkowski-Differenz in Richtung $\vec{r} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$.

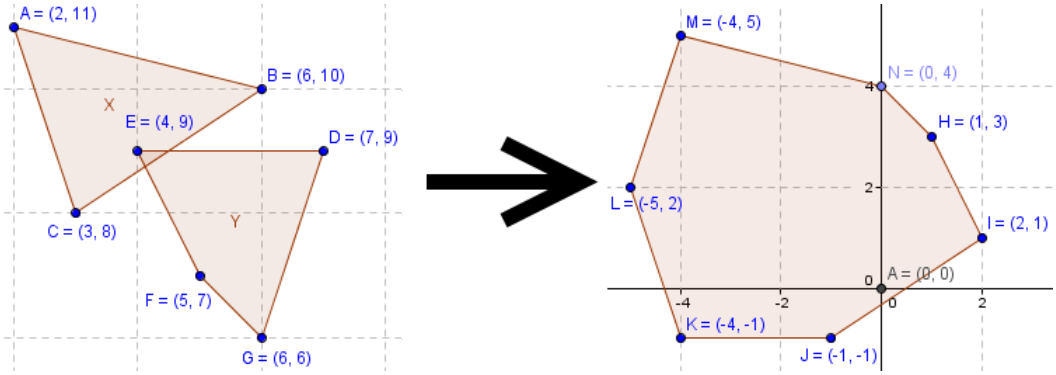


Abbildung 10: Beispiel einer Kollision zweier konvexer Mengen und der resultierenden Minkowski-Differenz

Zuerst wird die Ecke errechnet, welche am weitesten in der Richtung von Vektor $\vec{r}$ der Menge X entfernt ist. Dazu wird die Rechnung 19 aus Vorgehensweise 1 angewendet:

$$\vec{r} \cdot \vec{v}_A = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \cdot \begin{pmatrix} 2 \\ 11 \end{pmatrix} = 11 \tag{22}$$

$$\vec{r} \cdot \vec{v}_B = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \cdot \begin{pmatrix} 6 \\ 10 \end{pmatrix} = 10 \tag{23}$$

$$\vec{r} \cdot \vec{v}_C = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \cdot \begin{pmatrix} 3 \\ 8 \end{pmatrix} = 8 \tag{24}$$

Das Produkt mit dem Ortsvektor von A ist am größten und damit ist belegt, dass Punkt A sich am weitesten in Richtung $\vec{r}$ befindet. Zur Berechnung des Punktes der Menge Y, muss die Rechnung 20 aus Vorgehensweise 1 angewandt werden:

$$-\vec{r} \cdot \vec{v}_D = \begin{pmatrix} 0 \\ -1 \end{pmatrix} \cdot \begin{pmatrix} 7 \\ 9 \end{pmatrix} = -9 \tag{25}$$

$$-\vec{r} \cdot \vec{v}_E = \begin{pmatrix} 0 \\ -1 \end{pmatrix} \cdot \begin{pmatrix} 4 \\ 9 \end{pmatrix} = -9 \tag{26}$$

$$-\vec{r} \cdot \vec{v}_F = \begin{pmatrix} 0 \\ -1 \end{pmatrix} \cdot \begin{pmatrix} 5 \\ 7 \end{pmatrix} = -7 \tag{27}$$

$$-\vec{r} \cdot \vec{v}_G = \begin{pmatrix} 0 \\ -1 \end{pmatrix} \cdot \begin{pmatrix} 6 \\ 6 \end{pmatrix} = -6 \tag{28}$$

Der Ortsvektor von Punkt G hat das größte Skalar und somit befindet dieser

sich am weitesten in der gesuchten Richtung. Die Ecke, die am weitesten in Richtung $\vec{r}$ der Minkowski-Differenz liegt, wird schließlich durch die folgende Subtraktion der Ortsvektoren von A und G errechnet:

$$\vec{m} = \vec{a} - \vec{g} = \begin{pmatrix} 2 \\ 11 \end{pmatrix} - \begin{pmatrix} 6 \\ 6 \end{pmatrix} = \begin{pmatrix} -4 \\ 5 \end{pmatrix} \tag{29}$$

Folglich liegt die Ecke $M(-4|5)$ der Minkowski-Differenz am weitesten in der gesuchten Richtung. Es wurde also ein Ausgangspunkt für das Simplex gefunden.

Um einen weiteren Punkt zum Simplex hinzuzufügen, wird nun vom schon vorhandenen Simplex, also dem Punkt M, eine neue Ecke der Minkowski-Differenz in Richtung Ursprung gesucht. Es müssen daher die Rechnungen aus Vorgehensweise 1 ein weiteres mal ausgeführt werden, nur, dass dieses mal $\vec{r}$ durch den Vektor von Punkt M zum Ursprung gegeben ist. Rechnen wir dies noch einmal für die Menge X:

$$\vec{r} \cdot \vec{v}_A = \begin{pmatrix} 4 \\ -5 \end{pmatrix} \cdot \begin{pmatrix} 2 \\ 11 \end{pmatrix} = -47 \tag{30}$$

$$\vec{r} \cdot \vec{v}_B = \begin{pmatrix} 4 \\ -5 \end{pmatrix} \cdot \begin{pmatrix} 6 \\ 10 \end{pmatrix} = -26 \tag{31}$$

$$\vec{r} \cdot \vec{v}_C = \begin{pmatrix} 4 \\ -5 \end{pmatrix} \cdot \begin{pmatrix} 3 \\ 8 \end{pmatrix} = -28 \tag{32}$$

In diesem Fall hat $\vec{v}_B$ das größte Skalar und daher befindet sich der Punkt B am weitesten in der gesuchten Richtung. Rechnet man dies auch wieder für Menge Y aus, so erhält man den Punkt E. Durch Subtrahieren beider Punkte erhält man schließlich die neue Ecke auf der Minkowski-Differenz:

$$\vec{i} = \vec{b} - \vec{e} = \begin{pmatrix} 6 \\ 10 \end{pmatrix} - \begin{pmatrix} 4 \\ 9 \end{pmatrix} = \begin{pmatrix} 2 \\ 1 \end{pmatrix} \tag{33}$$

Der neue Punkt des Simplex ist also $I(2|1)$. Fügen wir diesen zum aktuellen Simplex, also dem Punkt M hinzu, ergibt sich folgende Gerade:

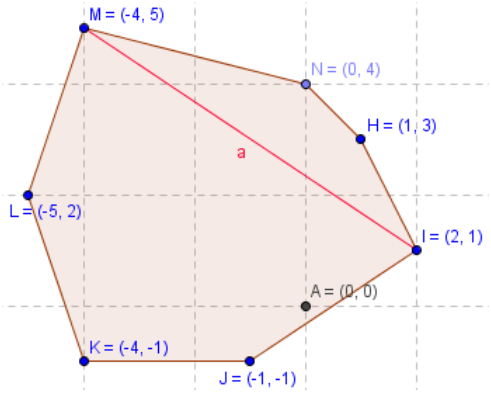


Abbildung 11: Das aktuelle Simplex

Immer wenn ein neuer Punkt zum Simplex hinzugefügt wird, kann die Abbruchbedingung des Algorithmus überprüft werden. Der Ursprung kann nur in der Minkowski-Differenz liegen, wenn dieser in dem Feld zwischen den beiden Geraden liegt, die orthogonal zum aktuellen Richtungsvektor stehen und durch die beiden schon vorhandenen Ecken des Simplex gehen. Die folgende Abbildung veranschaulicht dieses Feld:

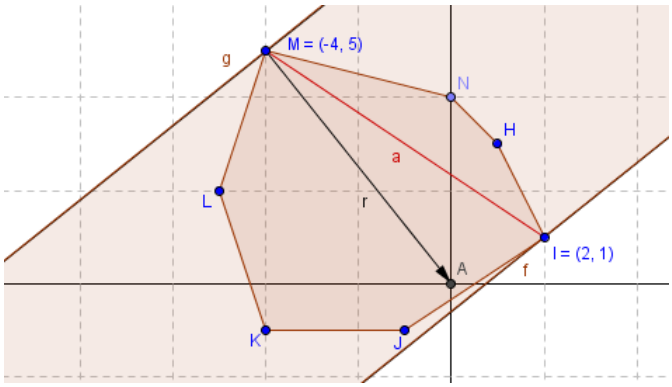


Abbildung 12: Der Ursprung muss im braunen Bereich liegen, damit eine Kollision weiterhin möglich ist.

Nur wenn der Ursprung innerhalb des braunen Bereichs von Abbildung 12 und damit zwischen den Geraden f und g liegt, kann er in der Minkowski-Differenz enthalten sein. Es muss also überprüft werden, auf welcher Seite sich der neue Punkt des Simplex von der Geraden befindet, die in Richtung des aktuellen Richtungsvektors liegt. Befindet sich der Punkt auf der Seite, in die der Richtungsvektor nicht zeigt, liegt er im braunen Bereich. Der Richtungsvektor ist der Normalenvektor der zu überprüfenden Geraden. Es wird also überprüft, ob der neue Punkt des Simplex den Ursprung passiert.

Wir überprüfen dies an dem neuen Punkt I in Abbildung 12 und berechnen, ob dieser den Ursprung passiert hat:

$$\lambda = -\vec{i} \cdot r \tag{34}$$

$$\Leftrightarrow \lambda = \begin{pmatrix} -2 \\ -1 \end{pmatrix} \cdot \begin{pmatrix} 4 \\ -5 \end{pmatrix} \tag{35}$$

$$\Leftrightarrow \lambda = -3 \tag{36}$$

Der Skalar λ ist negativ und damit wurde der Ursprung passiert. Das bedeutet, dass dieser immer noch innerhalb der Minkowski-Differenz liegen kann. Daher muss der Algorithmus weiter ausgeführt werden. Wäre λ nicht negativ, wäre der Algorithmus fertig, denn der neue Punkt liegt am weitesten in Richtung Ursprung vom aktuellen Simplex. Allerdings passiert dieser den Ursprung nicht, womit der Ursprung außerhalb des braunen Bereichs aus Abbildung 12 liegt und daher nicht in der Minkowski-Differenz enthalten sein kann.

Nun muss ein nächster Punkt zum Simplex hinzugefügt werden, um das erste Mal ein vollständiges Dreieck zu erhalten. Dazu wird der Vektor berechnet, der orthogonal zum aktuellen Simplex (hier eine Strecke) steht und in Richtung Ursprung zeigt. Dieser Vektor lässt sich leicht ablesen und auch berechnen. Auf Abbildung 11 bezogen gilt $\vec{r} = \begin{pmatrix} -4 \\ -6 \end{pmatrix}$. Mit dem neuen Vektor in Richtung Ursprung wiederholen wir das Verfahren von Vorgehensweise 1 und berechnen die Ecke der Minkowski-Differenz von Abbildung 11, die am weitesten in dieser Richtung liegt. Dabei ergibt sich folgende Subtraktion:

$$\vec{k} = \vec{c} - \vec{d} = \begin{pmatrix} 3 \\ 8 \end{pmatrix} - \begin{pmatrix} 7 \\ 9 \end{pmatrix} = \begin{pmatrix} -4 \\ -1 \end{pmatrix} \tag{37}$$

Der Punkt K ist also die neue Ecke des Simplex. Bevor das Simplex aber mit Punkt K erweitert werden kann, muss getestet werden, ob K den Ursprung passiert:

$$\lambda = -\vec{k} \cdot r \tag{38}$$

$$\Leftrightarrow \lambda = \begin{pmatrix} 4 \\ 1 \end{pmatrix} \cdot \begin{pmatrix} -4 \\ -6 \end{pmatrix} \tag{39}$$

$$\Leftrightarrow \lambda = -22 \tag{40}$$

Der Punkt $K(-4|-1)$ passiert den Ursprung und kann dem Simplex angehängt werden. Dies geschieht, indem dieser Punkt mit den schon errechneten Punkten M und I des Simplex verbunden wird.

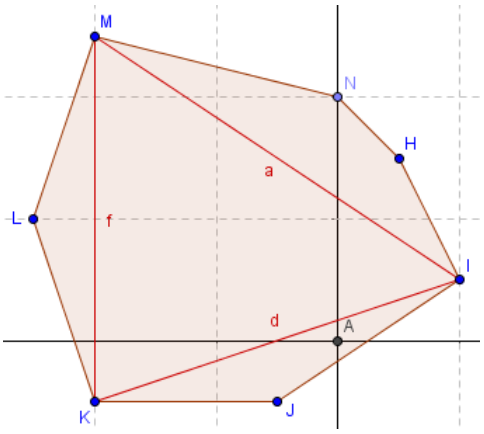


Abbildung 13: Vollständiges Simplex

Nun kann durch Anwendung von Satz 7 überprüft werden, ob der Ursprung sich innerhalb des gebildeten Simplex befindet oder nicht. Hierzu müssen die Normalenvektoren der Dreiecksseiten bestimmt werden, welche auf Abbildung 13 durch folgende Vektoren gegeben sind:

$$\vec{n}_f = \begin{pmatrix} -6 \\ 0 \end{pmatrix} \wedge \vec{n}_d = \begin{pmatrix} 2 \\ -6 \end{pmatrix} \wedge \vec{n}_a = \begin{pmatrix} 4 \\ 6 \end{pmatrix} \tag{41}$$

Mit den Normalenvektoren kann schließlich berechnet werden, ob der Ursprung innerhalb des Simplex ist:

$$-\vec{m} \cdot \vec{n}_f = \begin{pmatrix} 4 \\ -5 \end{pmatrix} \cdot \begin{pmatrix} -6 \\ 0 \end{pmatrix} = -24 \tag{42}$$

$$-\vec{k} \cdot \vec{n}_d = \cdot \begin{pmatrix} 4 \\ 1 \end{pmatrix} \cdot \begin{pmatrix} 2 \\ -6 \end{pmatrix} = 2 \tag{43}$$

$$-\vec{i} \cdot \vec{n}_a = \begin{pmatrix} -2 \\ -1 \end{pmatrix} \cdot \begin{pmatrix} 4 \\ 6 \end{pmatrix} = -14 \tag{44}$$

Der Skalar von Ortsvektor $-\vec{k}$ ist positiv und daher kann der Ursprung nicht innerhalb des gebildeten Simplex liegen, sondern befindet sich in Richtung des Normalenvektors der Strecke d. Es wird also ein neues Simplex mit dem neuen Richtungsvektor $\vec{n_d}$ gebildet, da in dieser Richtung der Ursprung vom aktuellen Simplex liegt. Daher muss wieder die Ecke der Minkowski-Differenz mit dem neuen Richtungsvektor gesucht werden und zur schon

vorhandenen Strecke d hinzugefügt werden.

Die Ecke, die am weitesten in Richtung $\vec{n}_d$ von der Strecke d liegt, ist die Ecke $J(-1|-1)$ der Minkowski-Differenz. Es ergibt sich also folgendes neues Simplex:

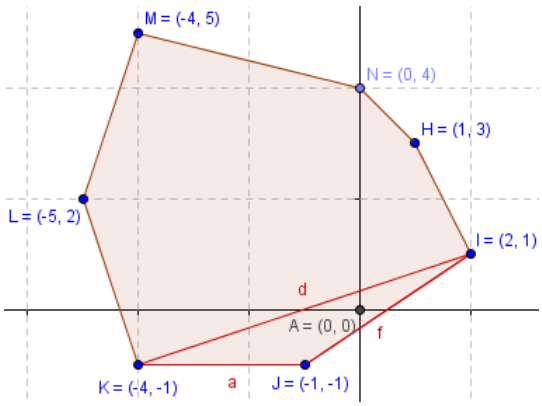


Abbildung 14: Zweites vollständiges Simplex

Das neue Simplex wird nun wieder darauf überprüft, ob es den Ursprung beinhaltet. Dies ist der Fall und damit ist bewiesen, dass der Ursprung sich in der Minkowski-Differenz befindet und eine Kollision zwischen den zwei Ausgangsmengen vorliegt.

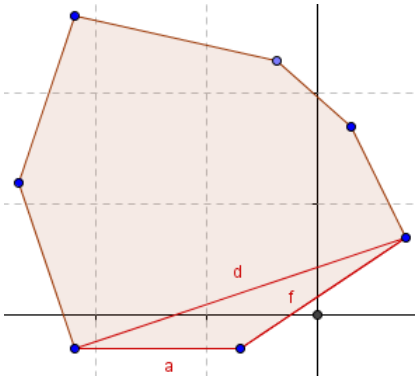


Abbildung 15: Vollständiges Simplex

Würde die Minkowski-Differenz den Ursprung nicht beinhalten, wie in Abbildung 15, würde der Algorithmus dies auch während der nächsten Rechnungen feststellen. Denn die Ecke, die am weitesten in Richtung Ursprung von der Strecke f liegt, passiert den Ursprung nicht mehr.

7.2 Pseudocode des GJK-Algorithmus

input : Polygon A, Polygon B
output: Kollision Wahr oder Falsch

```
Punkt P = punktAufMinkowskidifferenz(A, B, richtung);
Simplex simplex = [P];
Vektor richtung = richtungZumUrsprung(simplex);

while sucheNachKollision do
    P = punktAufMinkowskidifferenz(A, B, richtung);
    if Skalar(richtung, −P) > 0 then
        return False;
    else
        simplex = simplex ∪ P;
        if ursprungImSimplex(simplex) = Wahr then
            return Wahr;
        else
            richtung = richtungZumUrsprung(simplex);
        end
    end
end
```

Algorithm 1:

Der Algorithmus 1 veranschaulicht den Ablauf des GJK-Algorithmus als Pseudocode und verdeutlicht die iterative Suche nach einem Simplex in der Minkowski-Differenz.

7.3 Das Programm „Visual-GJK“

Der GJK-Algorithmus wurde in einem von mir entwickelten Programm in der Programmiersprache Java umgesetzt. Das Programm besitzt eine grafische Benutzeroberfläche, damit eine einfache Bedienung gewährleistet ist:

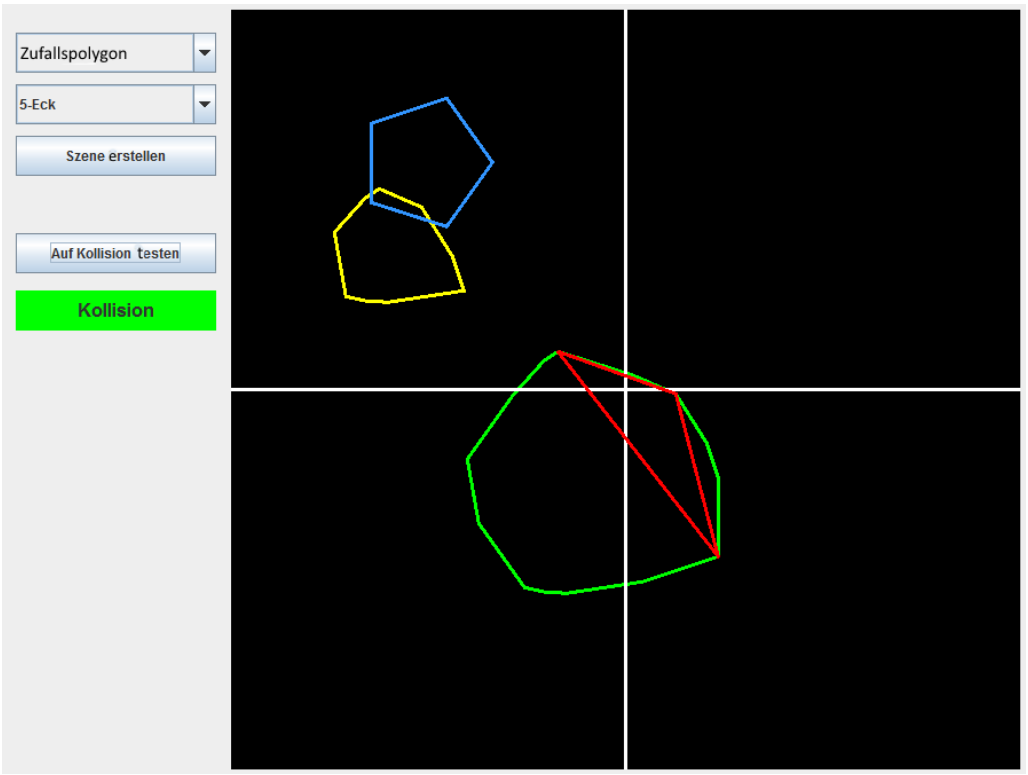


Abbildung 16: Grafische Benutzerobefläche des Progammes „Visual-GJK“

Mit den beiden Dropdown-Listen der Benutzeroberfläche kann eingestellt werden, was für Polygone erstellt werden sollen, um diese auf eine Kollision zu überprüfen. Durch den Button „Szene erstellen“ werden die ausgewählten Polygone schließlich in dem schwarzen Feld an einer zufälligen Position eingefügt. Die Polygone können per Drag and Drop im schwarzen Feld verschoben werden. Das grüne Polygon stellt immer die Minkowski-Differenz der beiden erstellten Polygonen dar. Beim Klicken auf den Button „Auf Kollision testen“ beginnt eine Animation, welche grafisch veranschaulicht, wie der GJK-Algorithmus versucht ein Simplex um den Ursprung zu finden.

8 Ausblick

Wenn sehr viele Körper auf Kollision überprüft werden müssen, dann kann dies sehr schnell eine Menge Performance des Computers in Anspruch nehmen. Ich habe zwei Möglichkeiten gefunden, vgl. hierzu [4] und [5], mit denen es möglich ist, Kollisionserkennungen zu beschleunigen. Diese werde ich einmal kurz vorstellen und in Ausblick zur Erweiterung des Themas Kollisionserkennung stellen.

8.1 Optimierung durch Hüllkörper

In den meisten Fällen, in denen Kollisionen zwischen sehr vielen Körpern bestimmt werden müssen, tritt der Fall ein, dass nur wenige Kollisionen stattfinden. Das bedeutet, dass ein komplexer Algorithmus wie der GJK-Algorithmus häufig angewendet wird, obwohl überhaupt keine Kollision vorhanden ist. Um dies zu verhindern, können so genannte Hüllkörper eingesetzt werden. Ein Hüllkörper ist ein einfacher geometrischer Körper, der einen komplexeren Körper so genau wie möglich einschließt. Hierzu bieten sich vor allem Kreise und Rechtecke an, denn sie lassen sich sehr schnell untereinander auf Kollisionen untersuchen. Es wird so vorgegangen, dass immer zuerst die Hüllkörper auf eine Kollision überprüft werden, denn dies ist schneller, als direkt den GJK-Algorithmus anzuwenden. Besteht keine Kollision der Hüllkörper, kann auch keine Kollision der eigentlichen Körper bestehen, denn diese befinden sich innerhalb der Hüllkörper. Kollidieren die Hüllkörper, muss mit dem GJK-Algorithmus genau getestet werden, ob auch wirklich eine Kollision der Ausgangskörper vorhanden ist.

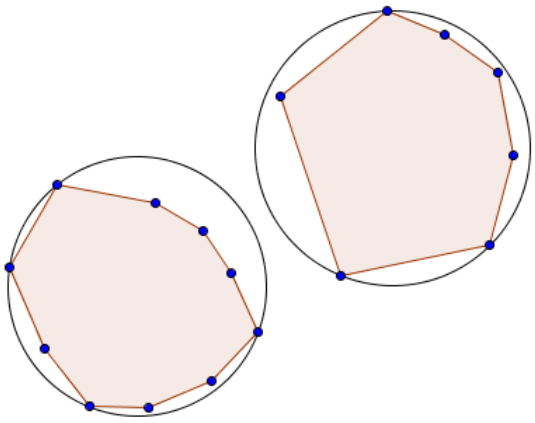


Abbildung 17: Zwei Körper mit Kreisen als konvexe Hülle

In Abbildung 17 ist ein Beispiel für zwei komplexe Körper zu sehen, deren

Hüllkörper nicht kollidieren. Somit muss auch nicht genauer getestet werden, ob eine Kollision zwischen den Ausgangskörpern vorhanden ist.

8.2 Optimierung durch Raumpartitionierung

Während die Optimierung durch Hüllkörper versucht, die Kollisionserkennungen zu beschleunigen, versucht die Optimierung durch Raumpartitionierung, die Anzahl an Kollisionserkennungen soweit wie möglich zu reduzieren. Befinden sich zwei Körper in zwei verschiedenen Ecken eines Raumes, ist es für einen menschlichen Betrachter sofort ersichtlich, dass keine Kollision zwischen den beiden Körpern vorhanden ist, denn eine offensichtliche räumliche Distanz trennt sie. Für einen Computer ist dies leider nicht so eindeutig, denn er muss alle Kollisionen rechnerisch belegen. Bei dieser Optimierung wird versucht, durch geschickte Raumpartitionierung festzustellen, welche Körper überhaupt für eine Kollision infrage kommen und welche nicht.

Ein Quadtree ist eine Baum-Datenstruktur, bei der jeder innere Knoten vier Blätter hat.

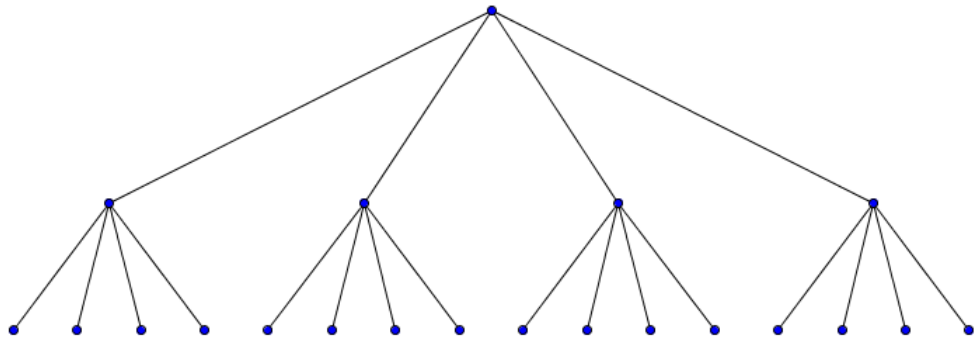


Abbildung 18: Beispiel eines Quadtree

Um nun diese Baumstruktur zur Raumpartitionierung zu verwenden, wird der zweidimensionale Raum, in dem sich alle Körper befinden, in vier gleich große Teile unterteilt. Die entstandenen Räume werden wiederum nach bestimmten Regeln in vier neue Teile unterteilt oder nicht mehr unterteilt.

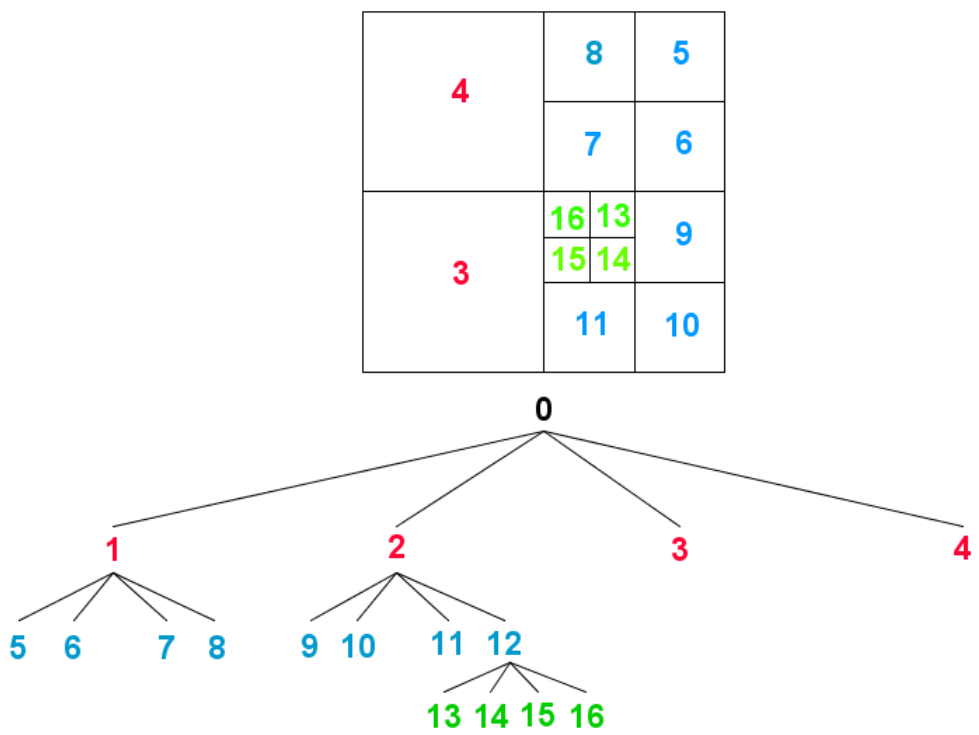


Abbildung 19: Beispiel einer Unterteilung eines zweidimensionalen Raumes mit dazugehörigem Quadtree

Bei der Unterteilung des Raumes wird nach folgenden Regeln vorgegangen:

- Hat ein Blatt keinen Kontakt mit einem Körper, wird es nicht weiter unterteilt.
- Liegt ein Blatt komplett in einem Körper, wird das Blatt nicht weiter unterteilt und der Körper wird im entsprechenden Blatt gespeichert.
- Hat ein Blatt nur teilweise Kontakt mit einem Körper, wird das Blatt ein weiteres mal unterteilt. Danach werden die neu entstandenen Blätter auf Kollision mit dem Körper überprüft.
- Erreicht ein Blatt eine vorgegebene maximale Höhe, wird es nicht weiter unterteilt und alle mit diesem kollidierenden Körper werden in diesem gespeichert.

Werden diese Regeln zur Einordnung eines Rechtecks in einen zweidimensionalen Raum angewandt, kann sich folgendes Bild ergeben:

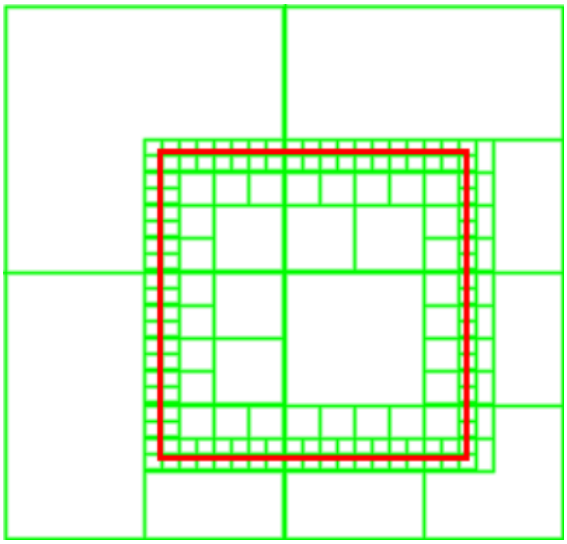


Abbildung 20: Beispiel einer Einordnung eines Rechtecks in einen zweidimensionalen Raum durch einen Quadtree mit der maximalen Höhe 5

Durch die Einsortierung in den Quadtree müssen nur noch die Körper auf eine Kollision überprüft werden, welche sich in dem gleichen Blatt befinden. Dadurch kann die Kollisionserkennung wesentlich beschleunigt werden. Grundsätzlich lässt sich sagen, dass je höher der Baum ist, desto weniger Kollisionen müssen berechnet werden, aber desto länger dauert die Erstellung des Quadtree. Es muss also für jede Anwendung eines Quadtrees eine möglichst optimale Höhe gefunden werden, damit die Baumerstellung und die anschließende Kollisionserkennung insgesamt möglichst wenig Perfomance verbrauchen. Des Weiteren sind die angegebenen Regeln nicht die einzige Möglichkeit Körper in einen Quadtree einzusortieren, sondern es können je nach Einsatzgebiet verschiedene verwendet werden.

9 Zusammenfassender Abschluss

9.1 Zusammenfassung

Durch die Minkowski-Differenz, kann das Problem einer Kollisionserkennung zwischen zwei konvexen Mengen, auf eine Kollisionserkennung zwischen dem Ursprung und einer konvexen Menge reduziert werden. Dies legt den Grundstein für den GJK-Algorithmus, welcher es darauf aufbauend schafft, in einer sehr kurzen Zeit eine Kollision nachzuweisen oder zu widerlegen. Ermöglicht wird dies, durch die Eigenschaft, nicht die gesamte Minkowski-Differenz berechnen zu müssen und durch den Satz von Carathéodory. Die Überlegenheit des GJK-Algorithmus wurde besonders im direkten Vergleich zu einer Brute-Force Methode deutlich.

9.2 Eigene Bewertung des Lösungskonzepts

Ich habe beim Erarbeiten meiner Projektarbeit viel Neues gelernt und es hat mir sehr viel Spaß gemacht, mich in die Strukturen des Algorithmus hineinzudenken. Besonders spannend finde ich die Mathematik, die hinter diesem Algorithmus steht und in diesem einen praktischen Bezug zum Lösen aktueller Problemen aus der Informatik erhält. Mit dem gefundenen Lösungsansatz bin ich sehr zufrieden, denn der vorgestellte Algorithmus ist schnell und schafft es ohne aufwändige Rechnungen, wie Gleichungssysteme, auszukommen. Außerdem befindet sich auf dem mitgelieferten Datenträger ein von mir entwickeltes Programm zur Veranschaulichung des GJK-Algorithmus unter dem Namen „Visual-GJK“. Um diesen Algorithmus in einer praktischen Anwendung einzusetzen, habe ich außerdem noch eine Physiksimulation von Festkörpern im zweidimensionalen Raum erstellt, welche sich unter dem Namen „RigidBody-Simulation“ auf dem mitgelieferten Datenträger befindet.

9.3 Kritische Reflexion

Kritisch an meiner Projektarbeit sehe ich, dass es mir an manchen Stellen nicht gelungen ist, meine Sätze genau nach den mathematischen Konventionen zu formulieren. So musste ich mir die Grundlagen der analytischen Geometrie selber erarbeiten, da dieses Gebiet erst Thema des nächsten Schuljahres ist. Es fehlte mir daher zum Teil das mathematische Hintergrundwissen und außerdem die Erfahrung um alle Sätze mathematisch präzise, korrekt und verständlich zu schreiben.

Quellen

[1] Rudolf Scharlau. *Diskrete Geometrie-Version(3)*. 12.12.2011

[2] Ivan Izmestiev. *Einführung in die Konvexgeometrie*. FU Berlin, WS 2003/2004

[3] William Bittle. *GJK (Gilbert–Johnson–Keerthi)*. 13.4.2010, unter: http://www.dyn4j.org/2010/04/gjk-gilbert-johnson-keerthi/, Eingesehen am: 23.3.2015

[4] *Hüllkörper*. 30. 6 2008, unter: http://wiki.delphigl.com/index.php/H%C3%BCllk%C3%B6rper, Eingesehen am: 29.4.2015

[5] Jonas Tihon. *QuadTree Implementierung und Visualisierung*. WS 2007/2008, unter: http://www.informatik.uni-trier.de/ naeher/Professur/PROJECTS/WS07/pro5/index.htm, Eingesehen am: 29.4.2015

Abbildungsverzeichnis

Titelbild: Newtons cradle with eight balls - panorama 40377563 © electriceye - fotolia.com

Danksagung

Mein besonderer Dank gilt meiner Projektkurslehrerin Frau Dr. S. Terveer für anregende und kritische Gespäche sowie für Hinweise, wenn bei mathematischen Problemen das Schulwissen nicht ausreichte. Weiterhin bedanke ich mich bei allen, die sich die Zeit genommen haben, mir bei der Beseitigung von Tippfehlern zu helfen.

Versicherung

Hiermit versichere ich, dass ich die Arbeit selbstständig angefertigt, keine anderen als die angegebenen Hilfsmittel benutzt und die Stellen der Projektarbeit, die im Wortlaut oder im wesentlichen Inhalt aus anderen Werken oder Quellen entnommen wurden, mit genauer Quellenangabe kenntlich gemacht habe. Ich bin damit einverstanden, dass die von mir verfasste Facharbeit in der Schulbibliothek anderen zugänglich gemacht wird.

Ort, Datum

Unterzeichner