Modelle für virtuelle Realitäten
Kontaktmechanik

Rigid Body

Üblicherweise nimmt man in den meisten Simulationen an, dass starre Körper simuliert werden. Für starre Körper gibt es in der Physik ausgereifte Modelle derer wir uns bedienen möchten. Man kann die Bewegung eines starren Körpers komplett beschreiben, wenn man die Position und Geschwindigkeit, sowie die momentane Rotation und die Rotationsgeschwindigkeit weiß.

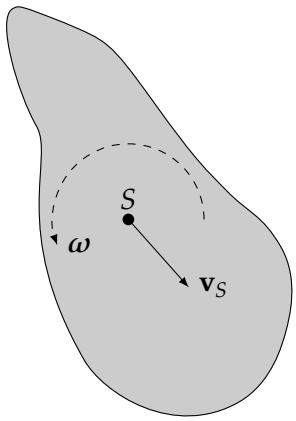

Abbildung 1: Vollständige Beschreibung der Rigid Body Bewegung

Für die Bewegung des Schwerpunktes gilt genau das Selbe wie für jede Partikelbewegung. Bei der Rotationsbewegung muss man ein wenig mehr Aufwand treiben. In Abbildung 1 ist ein Rigid Body dargestellt mit dem Schwerpunkt S der Schwerpunktsgeschwindigkeit $\mathbf{v}_S$ und der Drehgeschwindigkeit $\boldsymbol{\omega}$. Nicht dargestellt ist die aktuelle Rotation des Objektes gegenüber der Initialposition.
Eine übliche Form die Rotation eines Objektes zu speichern ist die Rotationsmatrix, jedoch ist diese für unsere Zwecke eher unhandlich und die Verwendung einer Rotationsmatrix fürht in den meisten Fällen zum Gimbal Lock, der besonders bei physikalischen Simulationen äußerst unschöne Effekte hervorruft. Deswegen verwenden wir Quaternionen.

Quaternion

Ein Quaternion ist eine 4-dimensionale Zahl (Realteil + 3 Imaginäranteile). Er beschreibt eine Scherdrehung im Raum und kann leicht für reine Drehungen verwendet werden. Ein Quaternion $\mathbf{q} = w + i \cdot x + j \cdot y + k \cdot z$ (i, j, k sind die Imaginären Zahlen) wird üblicherweise als Skalar-Vektorpaar geschrieben $\mathbf{q} = (w, \mathbf{v}) = w + \mathbf{v}$ mit: $\mathbf{v} = \begin{pmatrix} x & y & z \end{pmatrix}^T$. Da wir keine Scherungen mit den Quaternionen modellieren wollen beschränken wir uns auf die Einheitsquaternionen, also

$$|\mathbf{q}| = \sqrt{w^2 + x^2 + y^2 + z^2} = 1.$$

Ein Quaternion kann als Drehung um eine Achse verstanden werden. Jede Drehung, im 3-dimensionalen, um mehrere Achsen, kann als Drehung um eine Achse mit einem bestimmten Winkel aufgefasst werden. Wenn man

einen Punkt $\mathbf{x}$ im Raum um die Achse $\mathbf{v}$ mit Winkel α drehen möchte, kann er mit der zur Drehung gehörigen Quaternion $\mathbf{q} = \cos(\frac{\alpha}{2}) + \mathbf{v}\sin(\frac{\alpha}{2})$ abgebildet werden:

$$\mathbf{x}_R = \mathbf{q}\,\mathbf{x}\,\bar{\mathbf{q}}\ ,$$

wobei $\mathbf{x}_R$ der Punkt $\mathbf{x}$ nach der Rotation ist und $\bar{\mathbf{q}}$ der konjurgierte Quaternion von $\mathbf{q}$.
Um Drehungen nacheinander durchzuführen kann man die Quaternionen multiplizieren, wobei $\mathbf{q}_2 \times \mathbf{q}_1$ zunächst um $\mathbf{q}_1$ rotiert und anschließend um $\mathbf{q}_2$.

Rotationen in physikalischen Simulationen

Wirken in Rigid-Body Simulationen Kräfte, so können diese auch ein Drehmoment (Drehkraft, engl. torque) $\mathbf{M}$ hervorrufen, wenn die Kraft nicht direkt in Richtung des Massenschwerpunktes zeigt. Genauer muss man eine Kraft $\mathbf{F}$ die auf ein Rigid-Body in Punkt $\mathbf{p}$ wirkt folgendermaßen aufteilen:

$$\begin{aligned} \mathbf{F}_S &= \mathbf{F}\frac{\mathbf{F}\cdot\mathbf{r}}{|\mathbf{F}||\mathbf{r}|} \\ \mathbf{M} &= r \times \mathbf{F}, \end{aligned}$$

dabei ist $\mathbf{F}_S$ die Kraft auf den Schwerpunkt, die die Beschleunigung bewirkt, $\mathbf{F}\cdot\mathbf{r}$ ist die Projektion der Gesamtkraft auf die Radiusrichtung. Der Radiusvektor $\mathbf{r} = \mathbf{p} - \mathbf{s}$ mit dem Schwerpunkt $\mathbf{s}$ gibt die Verbindung von Schwerpunkt und dem Kraftansatzpunkt in dem die Kraft wirkt an. Das Drehmoment bewirkt, ähnlich den normalen Kräften, eine Drehbeschleunigung:

$$\boldsymbol{\tau} = I^{-1}(t)M$$

Wobei $I(t)$ der Trägheitstensor (Inertiatensor) ist, der den Widerstand gegenüber Rotationsbewegungen angibt. Da unterschiedliche Achsen unterschiedlich schwer drehbar sein können, reicht hier kein Skalar. Wir verwenden für unsere Simulationen eine Trägheitsmatrix $I = I(t = 0)$, die für die Initialrotation berechnet wird. Diese Trägheitsmatrix gilt aber nur für das lokale Koordinatensystem, weswegen wir die globale Trägheitsmatrix $I(t)$ zum Zeitpunkt t über die momentane Rotation $R(t)$ des starren Körpers errechnen:

$$I(t) = R(t)^{-1}I(0)R(t).$$

Anschaulich wird dabei die Drehbeschleunigung zunächst mit $R(t)$ in das lokale Koordinatensystem transformiert, dort kann die initiale Trägheitsmatrix verwendet werden um das lokale Drehmoment zu erhalten. Schließlich wird das Drehmoment wieder in die globalen Koordinaten transformiert. Um nun über das Drehmoment die Drehbeschleunigung zu erhalten benötigen wir nur noch $I(t)^{-1} = R(t)I(0)R(t)^{-1}$. Mittels der Drehbeschleunigung kann nun die Drehgeschwindigkeit $\boldsymbol{\omega}$ aktualisiert werden, es gilt einfach $\dot{\boldsymbol{\omega}} = \boldsymbol{\tau}$. Für die Änderung der Rotation muss man nur noch um die Rotationsachse $\boldsymbol{\omega}$ drehen, dafür wird üblicherweise ein Quaternion $d\mathbf{q}$ erzeugt, der gerade die Rotation um diese Achse darstellt. Also:

$$d\mathbf{q} = \cos\left(\frac{\theta}{2}\right) + \hat{\boldsymbol{\omega}}\sin\left(\frac{\theta}{2}\right),$$

mit $\theta = |\boldsymbol{\omega}| \cdot dt$ und $\hat{\boldsymbol{\omega}}$ dem normalisierten Vektor $\boldsymbol{\omega}$. Damit ergibt sich die neue Rotation:

$$\mathbf{q}(t + dt) = d\mathbf{q} \times \mathbf{q}(t).$$

(Beachte: Wenn man Quaternionen zur Beschreibung von Drehungen verwendet, hat die Multiplikation eine ähnliche Bedeutung, wie die Summe bei skalaren und vektorwertigen Größen)

Kollisionen

Die eigentliche Kollisionserkennung hängt stark von den eigentlichen Objekten ab und ist meist rein geometrisch motiviert. Dagegen kann man die Berechnung der Kollisionsantwort größtenteils verallgemeinern. Dazu verwendet man ein simples Feder-Masse System, das nur dann Kräfte ausgibt, wenn die Objekte sich überschneiden. Dieses Modell verhindert die Durchdringung nicht sofort, jedoch sind Modelle, die dies tun ungleich komplexer und selten schnell genug für Echtzeitanwendungen. In Abbildung 2 ist exemplarisch dargestellt wie solch eine Kollisionsfeder wirkt.

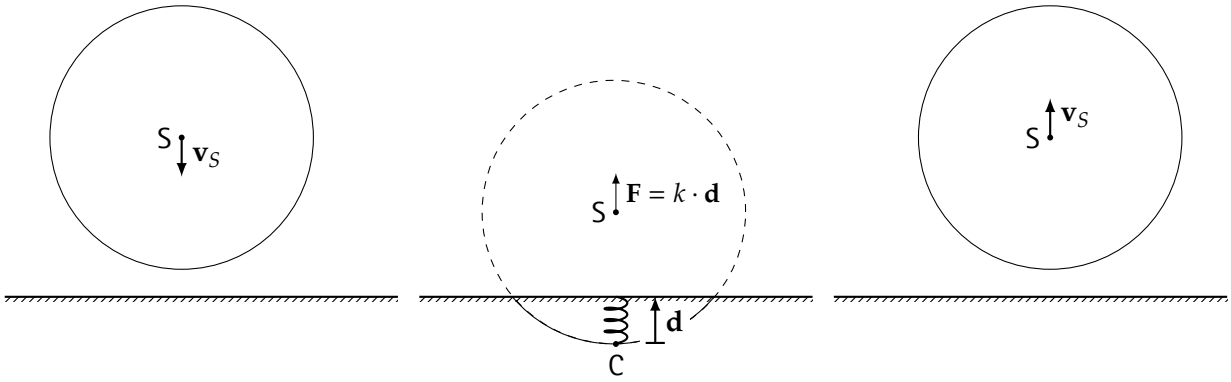

Abbildung 2: Kollisionsantwort realisiert mit Federkräften

Im einfachsten Fall benötigt man bei der Kollisionen zweier Objekte O_1, O_2 die Kontaktnormalen $\mathbf{n}_1, \mathbf{n}_2$ und die Eindringtiefe d. Mit diesen Größen lässt sich dann die Kraft auf die beiden Objekte berechnen als:

$$\begin{aligned} \mathbf{F}_1 &= kd\mathbf{n}_2 \\ \mathbf{F}_2 &= kd\mathbf{n}_1. \end{aligned}$$

Dieses Federmodell erhält im kontinuierlichen den Impuls und die Energie, man spricht von einem elasitschen Stoß. Durch die Diskretisierung der Zeit wird dem System fast immer Energie hinzugefügt, wodurch die Simulation instabil werden kann indem Objekte immer zu stark beschleunigt werden. Dem kann man entgegenwirken, indem man jede Kollision inelastisch macht, d.h. Energie aus dem System nimmt. Üblicherweise wird dies durch zwei geschwindigkeitsabhängige Federkonstanten $k_1 > k_2$ realisiert. Dazu verwendet man während die beiden Objekte ineinander eindringen eine größere Federkonstante, als wenn sie sich separieren, d.h:

$$\begin{aligned} \mathbf{F}_1 &= \begin{cases} k_1 d\mathbf{n}_2 & \mathbf{n}_2 \cdot (\mathbf{v}_1 - \mathbf{v}_2) < 0 \\ k_2 d\mathbf{n}_2 & \text{sonst} \end{cases} \\ \mathbf{F}_2 &= \begin{cases} k_1 d\mathbf{n}_1 & \mathbf{n}_1 \cdot (\mathbf{v}_2 - \mathbf{v}_1) < 0 \\ k_2 d\mathbf{n}_1 & \text{sonst} \end{cases}. \end{aligned}$$

Mit dieser Kraftberechnung können Objekte auch zur Ruhe kommen, jedoch exisitiert noch keine Form von

Reibung. Dazu bedienen wir uns des Coulombschen Reibungsgesetzes, das eine Unterscheidung zwischen der Haftreibung und der Gleitreibung macht.

$$\begin{aligned} |\mathbf{F}_t| &< \mu_H |\mathbf{F}_n| \Rightarrow v_t = 0 \\ \mathbf{F}_t^* &= -\mu_G |\mathbf{F}_n| \frac{\mathbf{v}_t}{|\mathbf{v}_t|}, \end{aligned}$$

wobei der index $\circ_t$ für den Kontakt-Tangential-Anteil steht und $\circ_n$ für den Kontakt-Normal-Anteil. Die Normalkraft $\mathbf{F}_n$ schließt alle herschenden Kräfte mit ein. Die Kraft $\mathbf{F}_t^*$ ist die resultierende Kraft durch die Reibung.

Aufgaben

Die folgenden Aufgaben sollen alle in 3D bearbeitet werden.

Aufgabe 1

Implementieren Sie eine einfache Kollisionsabfrage für Kollisionen zwischen Kugel-Kugel und Kugel-Ebene. Entwerfen Sie dazu eine einheitliche Datenstruktur zum Verarbeiten von Kollisionen, die alle nötigen Informationen enthalten soll.

Aufgabe 2

Fügen Sie den Kollisionsobjekten ein Attribut hinzu, welches diese fixieren kann (deren Bewegung verhindert). Fügen Sie dann folgende Kräfte hinzu:

- eine gravitative Kraft mit konstanter Beschleunigung, die auf alle nicht fixierten Objekte wirkt
- eine Kollisionskraft basierend auf dem Hook'schen Gesetz. Diese soll kollidierenden Objekte in Normalenrichtung mittels einer Kraft separieren.

Versuchen Sie sinnvolle Werte für die Konstante zu finden.
Implementieren sie auch die Integration der Rotation und testen Sie diese.
Hinweis: Die Trägheitsmatrix für eine homogene Kugel ist einfach $a \cdot E$, für ein $a > 0$.

Aufgabe 3

Erweitern Sie die bisherige Kollisionsbehandlung um eine Geschwindigkeitsabhängige Federkraft. Ziel soll es sein, ein Billardkugel-ähnliches Verhalten für Kugeln zu bekommen. Fügen Sie anschließend die Coulumbsche Reibung ein, beachten Sie dafür, dass Sie immer erst alle Kräfte durch Federn und Gravitation berechnen, um dann die Normalkraft für die eigentliche Reibung zu berechnen. Beachten Sie auch, dass bei der Berechnung der Relativgeschwindigkeit zweier Objekten die Rotationsgeschwindigkeit eine Rolle spielt:

$$\mathbf{v} = \mathbf{v}_S + \mathbf{w} \times \mathbf{r}$$

Die Reibung wirkt immer tangential zur Oberfläche, entgegen der relativen Bewegung.

Aufgabe 4

Als nächstes soll die Kollisionsbehandlung um Würfel erweitert werden. Dazu wird das Trägheitsmoment eines Würfels benötigt. Berechnen Sie dieses mittels:

$$I = \iiint_V \rho(x,y,z)\left((x^2+y^2+z^2)\mathbf{E}_3 - \begin{bmatrix} xx & xy & xz \\ yx & yy & yz \\ zx & zy & zz \end{bmatrix}\right) dx\,dy\,dz$$

Dabei kann das Integral für jede einzelne Matrixzelle separat ausgerechnet werden. Als Ergebnis erhält man dann natürlich wieder eine Matrix. Es ist üblicherweise hilfreich den Würfel achsenparallel auszurichten.

Aufgabe 5

Fügen Sie Ihrer Kollisionsbehandlung die Kollision von Box-Kugel und Box-Ebene hinzu. Überprüfen Sie dafür zunächst ob sich einer der acht Eckpunkte der Box innerhalb der Kugel / unter der Ebene befindet. Fügen Sie anschließend eine Kollisionsberechnung mit den Seiten des Würfels für die Box-Kugel Kollisionen hinzu.

Aufgabe 6

Um auch Box-Box Kollisionen zu ermöglichen kann man für die Seiten Ebenen einfügen. Welches Problem ergibt sich mit diesem Ansatz? Überlegen Sie sich, wie man die Probleme lösen könnte. Implementieren Sie die Kollision zwischen zwei Boxen.

Hinweis: Eine mögliche Lösung für das Box-Box Problem ist die Verwendung der Geschwindigkeiten. Des Weiteren werden gerne Ellipsoide an die Seiten des Würfels gesetzt um die durchdrungene Seite zu identifizieren.

Literatur

- Eberly, David H. Game physics. Morgan Kaufmann, 2003.
- Garstenauer, Helmut, and D. I. D. G. Kurka. Ä unified framework for rigid body dynamics."Degree Paper (2006).
- Nguyen, Hubert. Gpu gems 3. Addison-Wesley Professional, 2007.