

Die Flugbahn eines springenden Tennisballs

Timur Horlov

June 2025

Abstract

In dieser Arbeit wird die Flugbahn eines auf einem Tisch springenden Tennisballs untersucht. Zu diesem Zweck wurden zahlreiche unterstützende Softwaremodule entwickelt – zur Videoanalyse, zur Extraktion der Trajektorie, zur Umrechnung der Bildkoordinaten in reale Maße sowie zur Auswertung der Ergebnisse und zur Überprüfung des mathematischen Modells. Die Resultate zeigen eine ausreichende Übereinstimmung zwischen den experimentell bestimmten Trajektorien und dem theoretisch erwarteten Bewegungsverlauf.

1 Einleitung

Das Abprallen eines Balls vom Untergrund ist ein einfaches, aber interessantes physikalisches Phänomen, besonders bei einem Tennisball, der dutzende Male auf und ab springen kann. In dieser Arbeit wurde die Bewegung eines Tennisballs verfolgt und mit Hilfe eines mathematischen Modells simuliert.

2 Materialien und Methoden

Für das Experiment kamen folgende Hilfsmittel zum Einsatz:

- Smartphone-Kamera auf einem Stativ
- Tennisplatte und Tennisball
- Laptop für Datenerfassung und -verarbeitung

3 Softwareentwicklung

Zur Analyse und Modellprüfung wurden verschiedene Softwaremodule in C/C++ und GLSL entwickelt. Es wurden Bibliotheken wie FFmpeg, OpenGL und ImGui sowie meine eigene graphische Bibliothek eingesetzt.

3.1 Video-zu-Frame-Konverter

Dieses Modul wandelt komprimierte Videoformate (z.B. MP4, AVI) in rohe YUV420-Kadren um. YUV420 benötigt halb so viel Speicher wie RGB, da es unwesentliche Farbkonfigurationen reduziert. Das Modul ermöglicht zudem nahezu verzögerungsfreies Auslesen einzelner Frames.

3.2 Frame-Stepper

Der Frame-Stepper ermöglicht die Einzelbilddarstellung für detaillierte Analyse jedes Frames.

3.3 Ball-Markierungsmodul

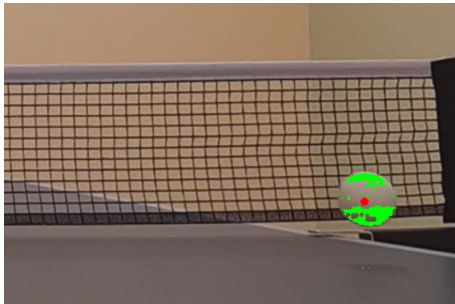


Figure 1: wie der Algorithmus den Ball 'sieht'

Mit diesem Modul kann der Ball in jedem Frame manuell oder halbautomatisch markiert werden. Ein Untermodul zur Bild-differenzierung erstellt einen Referenzframe ohne Ball, subtrahiert ihn von jedem Frame und findet so nach Filterung und Mittelung die Ballposition in wenigen hundert Millisekunden.

3.4 Koordinaten-Transformation

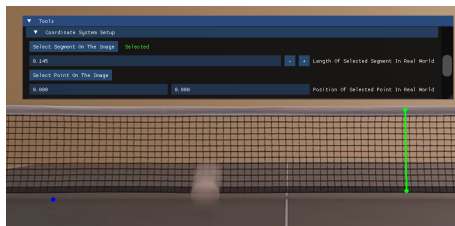


Figure 2: die Einrichtung des Koordinatensystems

Das Modul rechnet Bildkoordinaten in Weltkoordinaten um. Zunächst wird eine bekannte Distanz im Bild (z. B. Netzbreite) zugewiesen, um den Skalierungsfaktor zu bestimmen. Anschließend werden beliebige Nullpunkte festgelegt.

3.5 Graphen-Plotter

Ein Plotter visualisiert ausgewählte Messgrößen mit einem SI-Einheiten-Raster (Abstand 1 Einheit: Meter, m/s, Sekunde).

4 Datenauswertung

4.1 Trajektorie im xy -Diagramm

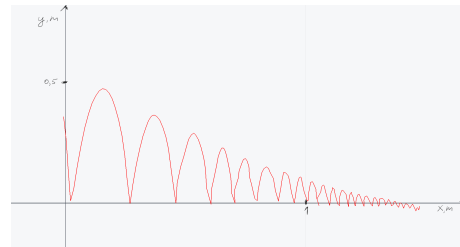


Figure 3: $y(x)$ Diagramm

Der Plot der Trajektorie $y(x)$ zeigt aufeinanderfolgende Parabelbögen mit abnehmender Scheitelhöhe. Die Abnahme der Maxima erfolgt zuerst annähernd linear, verlangsamt sich jedoch ab dem 6.-7. Abprall und nähert sich exponentiellem Verhalten.

4.2 Bewegung entlang der Achsen

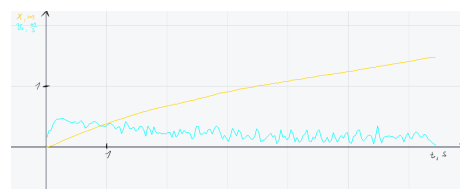


Figure 4: $x(t)$ (gelb) und $v_x(t)$ (blau)

Die Funktion $x(t)$ ist nahezu linear, da auf den Ball kaum horizontale Kräfte wirken. Ein Geschwindigkeitsverlust tritt nur bei jedem Abprall auf. Das Rauschen in $v_x(t)$ resultiert aus der 30-fps-Aufnahme und begrenzter Markierungspunkte, zeigt jedoch deutlich fast konstante horizontale Geschwindigkeit.

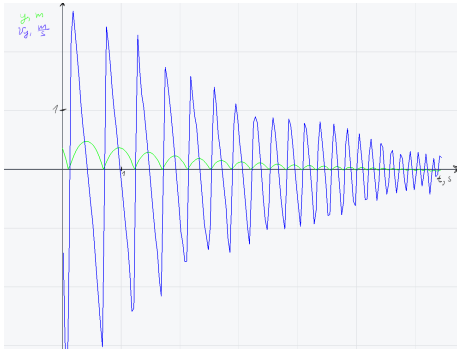


Figure 5: $y(t)$ (grün) und $v_y(t)$ (blau)

Die Vertikalposition $y(t)$ bildet eine Serie von Parabeln, da $v_y(t)$ zwischen Abprallereignissen linear abfällt. Jeder Abprall erzeugt einen Sprung in $v_y(t)$ mit verringerter Amplitude. Bis zum 6.-7. Abprall folgt die Abnahme der Scheitelhöhen einer linearen Annäherung.

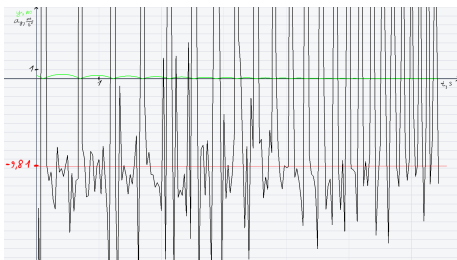


Figure 6: $y(t)$ (grün) und $a_y(t)$ (schwarz)

Der Plot von $a_x(t)$ wird hier nicht dargestellt, da er lediglich rauschbedingte Schwankungen um null zeigt und somit für die Analyse wenig aussagekräftig ist. Im Gegensatz dazu oszilliert $a_y(t)$ um Werte zwischen 8 und 12 m/s^2 , was nahe am freien Fall liegt ($g = 9.81 \text{ m/s}^2$). Scharfe Spitzen entsprechen Momenten, in denen die Normalkraft den Ball beschleunigt.

5 Modellbildung und Simulation

5.1 Beobachtete Bewegungscharakteristika

Aus den experimentellen Auswertungen ergeben sich folgende Schlüsse:

- Die Bewegung entlang der x -Achse ist nahezu gleichförmig.
- Die Bewegung entlang der y -Achse ist gleichmäßig beschleunigt.
- Beim Aufprall verliert der Ball einen Teil der Geschwindigkeit in beiden Richtungen.

5.2 Rebound-Koeffizienten

Zur Modellierung des Geschwindigkeitsverlusts werden die Rebound-Koeffizienten e_x und e_y eingeführt, welche den prozentualen Anteil der Geschwindigkeit nach dem Aufprall beschreiben. Der Energieverlust beim Aufprall entsteht dabei hauptsächlich durch Reibung sowie durch die irreversible Deformation des Balls und der Aufprallfläche.

$$\begin{aligned} v_{x,\text{neu}} &= e_x v_{x,\text{alt}}, \\ v_{y,\text{neu}} &= -e_y v_{y,\text{alt}}. \end{aligned}$$

5.3 Iteratives Simulationsmodell

Die iterativen Gleichungen für Position und Geschwindigkeit lauten:

$$\begin{aligned} a_x &= 0, \\ a_y &= -g. \end{aligned}$$

x_0 = Startpunkt auf der x -Achse,

y_0 = Startpunkt auf der y -Achse,

$v_{x,0}$ = konstante Anfangsgeschwindigkeit in x -Richtung,

$v_{y,0}$ = konstante Anfangsgeschwindigkeit in y -Richtung.

Für jeden Zeitschritt $n = 1, 2, \dots$ mit Schrittweite Δt :

$$\begin{aligned}v_{x,n} &= v_{x,n-1} + a_x \Delta t = v_{x,n-1}, \\v_{y,n} &= v_{y,n-1} + a_y \Delta t, \\x_n &= x_{n-1} + v_{x,n-1} \Delta t, \\y_n &= y_{n-1} + v_{y,n-1} \Delta t.\end{aligned}$$

Sobald $y_n \leq 0$:

$$\begin{aligned}y_n &\mapsto 0, \\v_{x,n} &\mapsto e_x v_{x,n}, \\v_{y,n} &\mapsto -e_y v_{y,n}.\end{aligned}$$

Listing 1: Kompakter Pseudocode des iterativen Modells

```
1 # Anfangsbedingungen
2 x, y, vx, vy = x0, y0, vx0, vy0
3 dt, g = 0.01, 9.81
4 e_x, e_y = 0.95, 0.80
5 traj = []
6
7 for t in range(0, T_max, dt):
8     # Bewegung
9     x += vx*dt
10    y += vy*dt
11    vy -= g*dt
12
13    # Aufprall
14    if y <= 0:
15        y = 0
16        vy = -e_y*vy
17        vx *= e_x
18
19    traj.append((t, x, y, vx, vy))
```

5.4 Modellbewertung und Vergleich mit Experiment

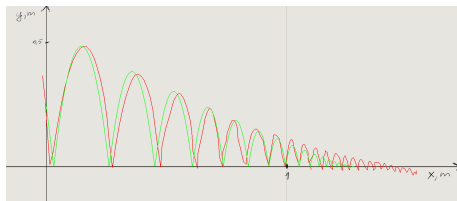


Figure 7: $y(x)$ Experiment (rot) und $y(x)$ Modell (grün)

Beim Vergleich der simulierten und experimentell gemessenen Trajektorien zeigt sich eine gute Übereinstimmung, insbesondere im mittleren Bereich der Bewegung. Für die Simulation wurden die experimentell bestimmten Parameter $e_x = 0.925$ und $e_y = 0.893$ verwendet.

Die modellierte Bewegung reproduziert das Verhalten des Balls nach mehreren Sprüngen mit hoher Genauigkeit. Kleinere Abweichungen zu Beginn und am Ende der Bewegung lassen sich möglicherweise dadurch erklären, dass die Rebound-Koeffizienten in Wirklichkeit nicht konstant sind, sondern von der momentanen Geschwindigkeit oder weiteren Faktoren wie Rotation abhängen könnten.

Trotz dieser Einschränkungen ist das Modell im Rahmen dieses Experiments als vollständig zufriedenstellend zu bewerten.

6 Mögliche Verbesserungen

Im Rahmen dieses Projekts wurden mit begrenzten technischen Mitteln bereits erstaunlich präzise Ergebnisse erzielt. Dennoch ergeben sich mehrere Ansatzpunkte zur Verbesserung:

- **Höhere Bildrate:** Die Verwendung einer Kamera mit höherer Bildrate (z.B. 120 fps oder mehr) würde die Genauigkeit der Geschwindigkeits- und Beschleunigungsberechnungen erheblich verbessern.
- **Rotation und 3D-Modellierung:** Die Berücksichtigung von Effekten wie Ballrotation oder einer vollständigen 3D-Trajektorie könnte das Modell realistischer und umfassender gestalten.
- **Variable Rebound-Koeffizienten:** Eine Geschwindigkeits- oder winkelabhängige Modellierung der Rebound-Koeffizienten e_x und e_y könnte die Genauigkeit am Anfang und Ende der Bewegung weiter erhöhen.

7 Fazit

Diese Arbeit zeigt, dass es mit einfachen Mitteln möglich ist, ein physikalisches Phänomen wie die Sprungbewegung eines Tennisballs zu analysieren, mathematisch zu modellieren und numerisch zu simulieren. Die entwickelten Softwaremodule ermöglichen eine präzise Analyse und Visualisierung der Bewegung. Die Resultate der Simulation stimmen gut mit der Realität überein, insbesondere im mittleren Bereich der Trajektorie.

Insgesamt belegt das Projekt, wie effektiv mathematische Modelle in Kombination mit moderner Softwareentwicklung eingesetzt werden können, um reale physikalische Prozesse zu untersuchen und verständlich darzustellen.

Erklärung: Da ich kein deutscher Muttersprachler bin, wurde ein Teil des Textes mit Google Translate erstellt und anschließend mit KI überarbeitet. Ansonsten kam keine KI zum Einsatz.