
4 Partikelsysteme (particle systems)

Computer-Animation & Kollisions-Detektion

Wintersemester 2011/12

4 Partikelsysteme

Beispiele von Partikelsysteme in Filmen :

- „Herr der Ringe“, Balrog Animation
- „Die Mumie“, Sandanimation
- „Hidalgo“, Sandanimation
- „Star Trek II“, beschriebener Genesis-Effekt



„Hidalgo“,
Sandsturm



„Herr der Ringe“,
Balrog

4 Historie - Wegbereiter

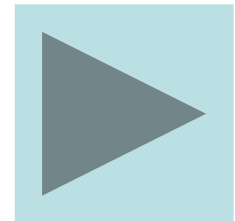
William Reeves

- Urvater der Partikel Systeme, veröffentlicht 1983 das Paper „*Particle Systems: A Technique for Modeling a Class of Fuzzy Objects*“
 - 1980 -1986: Arbeit bei Lucasfilm / ILM
 - U.a. mitgearbeitet an
Star Trek (The Wrath of Khan), Star Wars (Return of the Jedi)
 - Seit 1986: bei Pixar
 - Mitgearbeitet an: Luxo Jr., ..., A bug's life, ..., Ratatouille

Karl Sims

- 1990 Veröffentlichung des Papers „*Particle Animation and Rendering Using Data Parallel Computation*“

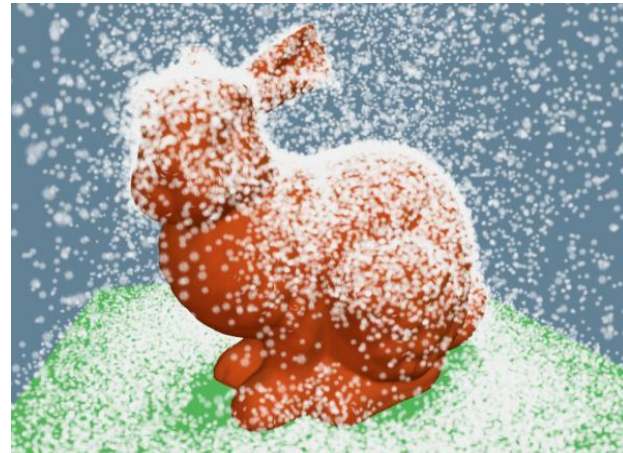
<http://www.karlsims.com/particle-dreams.html>



4 Partikelsysteme - Allgemeines

Was ist ein Partikel?

- Der Grundgedanke Reeves stellt sich ein Partikel als ein minimales Teilchen vor...
 - Sandkorn
 - Funke
 - Wassertropfen
 - Schneeflocke



4 Partikelsysteme - Allgemeines

Die Partikel in einem Partikelsystem haben gewisse Eigenschaften (Attribute), abhängig von ihrer Funktionalität.

Haupteigenschaften:

- Haupteigenschaften:
 - *Position*
 - *Geschwindigkeit / Bewegungsrichtung*
 - *Lebenszeit / Energie / Masse*
- Optionale Eigenschaften:
 - *Größe/Form*
 - *Normale*
 - *Farbe & Transparenz*
 - *vorherige Position*
 - ...

4 Partikelsysteme - Allgemeines

- Partikel interagieren mit ihrer Umgebung
 - Kollision
 - Einfluß von Kräften
- Partikel interagieren **nicht** miteinander
 - Keine Kollisionen untereinander
 - Keine gegenseitigen Kräfte
- Partikelsysteme eignen sich hervorragend zur Parallelisierung

4 Partikelsysteme - Allgemeines

Der *Systemkern*, des Partikelsystems „verwaltet“ die Partikel.

- Generiert und vernichtet Partikel
- Definiert Koordinatensystem
- Verwaltet Bewegung der Partikel
- Steuert äußere Einflüsse
wie z.B. Gravitation, Wind

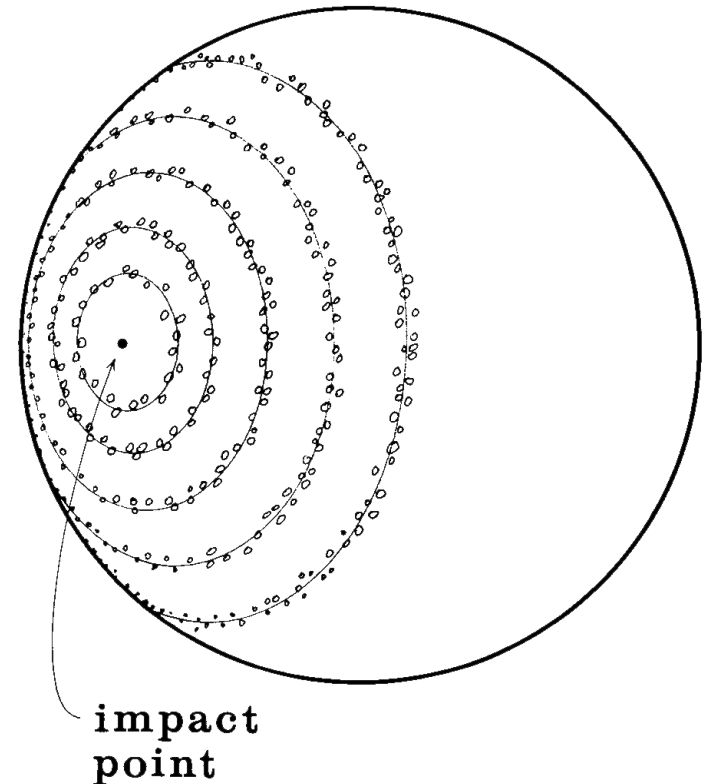
Es kann mehrere Systemkerne geben;
Systemkerne können hierarchisch
gegliedert sein.



4 Partikelsysteme - Allgemeines

„Genesis-Effekt“: Top-Level System

- Erzeugt auf den Ringen Second-Level Systeme
- Anzahl der Second-Level Systeme auf dem Ring ist von dem Umfang und einem Dichte Wert abhängig
- Dadurch wird eine „Feuerwand“ erzeugt



4.1 Phasen eines Partikel Systems

Erstellung eines neuen frames erfolgt in **5 Schritten**.

- (1) Generierung neuer Partikel
- (2) Initialisierung der Attributen von neuen Partikel
- (3) Entfernen von Partikeln
- (4) Update der Attribute (insbes. Transformation)
- (5) Rendern des neuen Frames

4.1 Phasen eines Partikel Systems

(1) Generierung neuer System Partikel:

- **Startposition** der neuen Partikel
(Punktquelle - Flächenquelle – Volumenquelle).
$$InitialPosition = MeanPosition + Rand() * VarPosition$$
- Nach Reeves ist die **Anzahl** der erzeugten Partikel sowohl von festen Werten als auch von stochastischen Anteilen abhängig.

$$NParts_f = MeanParts_f + Rand() * VarParts_f$$

$NParts_f$:	Gesamtanzahl an emittierten Partikeln (per frame)
$MeanParts_f$:	Die Durchschnittliche Anzahl an emittierten Partikeln
$VarParts_f$:	Varianzwert
$Rand()$:	Zufallsfunktion liefert Wert zwischen -1 und +1

4.1 Phasen eines Partikel Systems

(1) Generierung neuer System Partikel:

- Aus Effizienzgründen ist es sinnvoll, Zahl der Partikel von der Größe des Bildabschnittes abhängig zu machen
- Erweiterung der Formel:

$$NParts_f = (MeanParts_{SA(f)} + Rand() * VarParts_{SA(f)}) * ScreenArea$$

4.1 Phasen eines Partikel Systems

(2) Zuordnung von Attributen an neue Partikel:

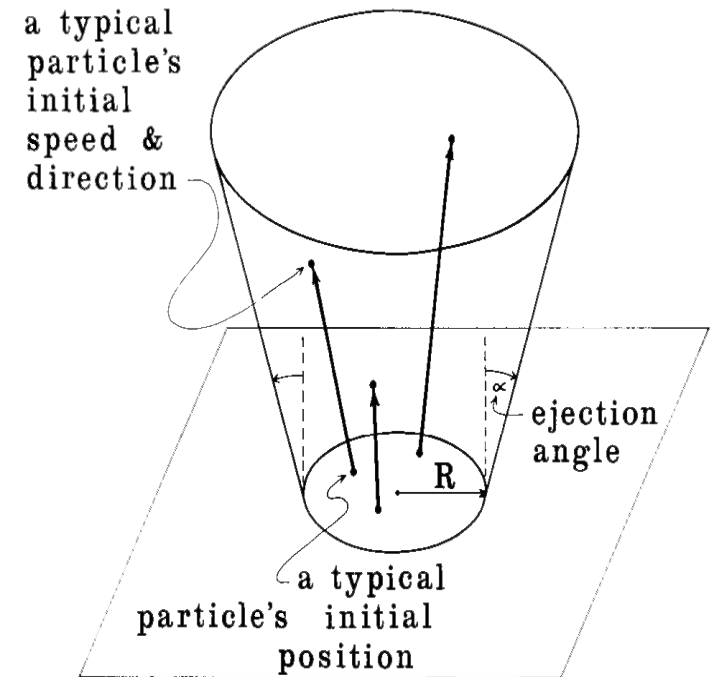
- Jedem neuen Partikel werden die im System definierten Eigenschaften zugewiesen.
- Die möglichen Initialwerte werden i.a. vom Systemkern spezifiziert
 - Farbbereich
 - Ausstoßrichtung
 - min/max Geschwindigkeit
 - min/max Lebenszeit

4.1 Phasen eines Partikel Systems

(2) Zuordnung von Attributen an neue Partikel:

- In der Regel gibt es bei der Zuweisung der Attribute einen stochastischen Anteil.
z.B. Geschwindigkeit und Farbe

$$\begin{aligned} \text{InitialSpeed} &= \\ &\text{MeanSpeed} + \text{Rand}() * \text{VarSpeed} \\ \text{InitialColor} &= \\ &\text{MeanColor} + \text{Rand}() * \text{VarColor} \end{aligned}$$



4.1 Phasen eines Partikel Systems

(3) Entfernen von Partikeln

- Entfernung, weil Lebenszeit abgelaufen
 - z.B. bei Feuerpartikeln, wenn sie abgekühlt sind
 - Initialisierung besitzt einen stochastischen Anteil
- Entfernung, wenn Partikel den Systembereich verlassen

4.1 Phasen eines Partikel Systems

(4) Update der Attribute

- Die Attribute der Partikel, die sich noch im System befinden werden verändert.
- Insbesondere:
 - Geschwindigkeit
 - Position
 - Lebenszeit
- Aber auch optionale:
 - Farbe
 - Form
 - Transparenz
 - ...

4.1 Phasen eines Partikel Systems

(5) Rendern des neuen Frames:

- Nachdem Position und Eigenschaften der Partikel für den neuen Frame festgelegt wurden, wird das Bild gerendert.
- Jedem Partikel ist eine darstellbare Geometrie zu gewiesen
 - Punkt, Linie, ...
 - Kugel, Würfel, ...
 - texturiertes Primitivbekommt „Farbe“, „Transparenz“, ..., Normale

4.2 Transformationen

Neben den einfachen Transformationen (lineare Bewegung ggf. stochastisch verändert) gibt es zahlreiche spezielle Bewegungsmuster

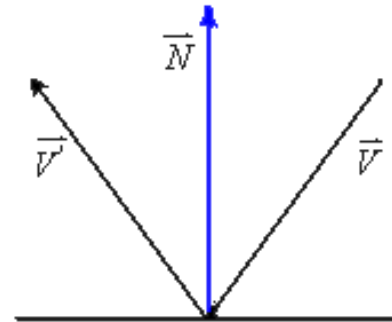
1. Dynamik (aufgrund physikalischer Kräfte, z.B. Gravitation)
2. Abprall-Effekte
3. Wirbel
4. Spiralbewegung
5. Dämpfung

4.2 Transformationen

2. Abprall-Effekte

- Formel ohne Energieverlust:

$$\vec{V}' = \vec{V} - 2\langle \vec{V}, \vec{N} \rangle \cdot \vec{N}$$



- Formel mit Energieverlust

$$\vec{V}' = (1 - \mu)\vec{V}_t - \epsilon\vec{V}_n \quad \epsilon: \text{Dämpfungsfaktor}(0..1)$$

$$\vec{V}_n = \langle \vec{V}, \vec{N} \rangle \cdot \vec{N} \quad \mu: \text{Reibungsfaktor}(0..1)$$

$$\vec{V}_t = \vec{V} - \vec{V}_n$$

V_n : Geschwindigkeit in Normalenrichtung

V_t : Geschwindigkeit in Tangentenrichtung

4.2 Transformationen

3. Wirbel (Verwendung um z.,B. Wind zu simulieren)

Ein Wirbel ist definiert durch...

- Rotationsachse
- Zentrum
- Stärke (magnitude μ)
- Enge (tightness τ)

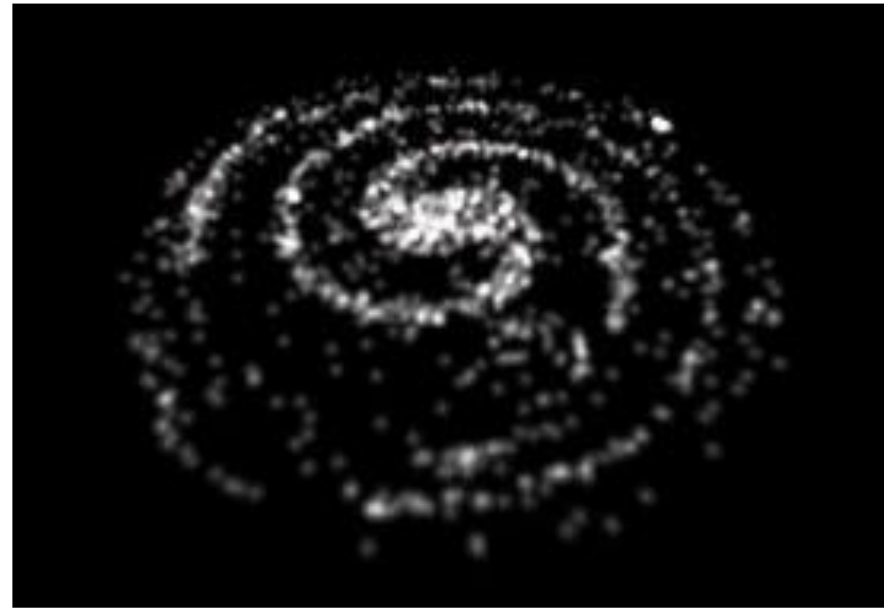
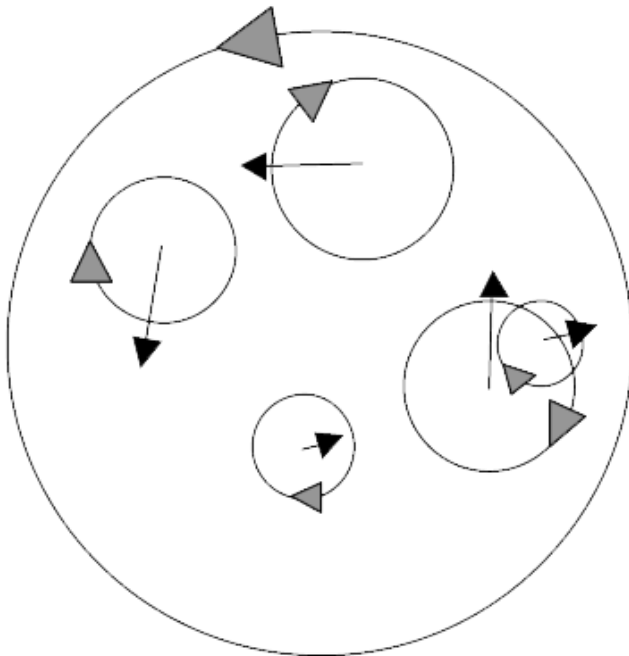
Position wird abhängig von Entfernung
R zum Wirbelzentrum um den Winkel
um die Rotationsachse gedreht..

$$\varphi = \frac{\mu}{R^\tau} \cdot \Delta t$$

4.2 Transformationen

3. Wirbel (Verwendung um z.B. Wind zu simulieren)

Mehrere Wirbel können überlagert werden
(lineare Superposition)



4.2 Transformationen

4. Spiralen (z.B. Schneeflocken)

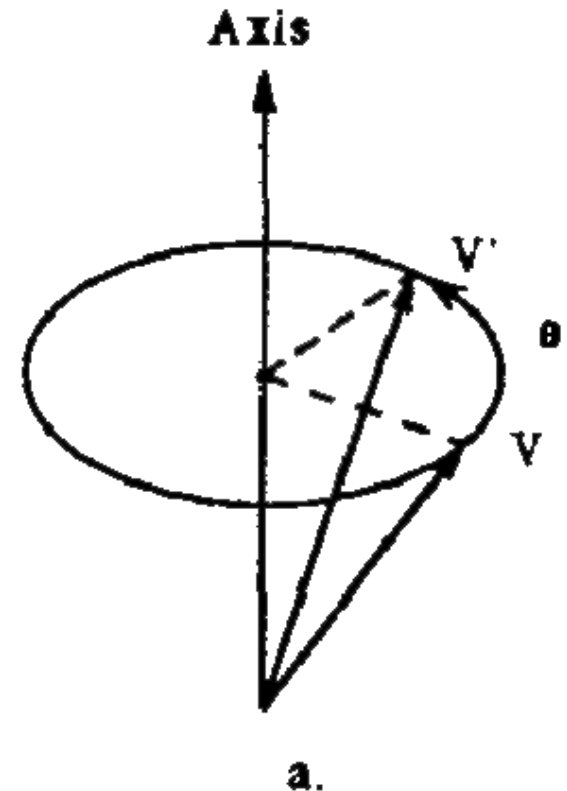
Es werden

- Spiralachse (Axis)
 - Spiral-Geschwindigkeit s (Winkel/Zeit)
- spezifiziert

Der Geschwindigkeitsvektor V eines Partikels wird um den Winkel θ um die Spiral-Achse gedreht

$$\theta = s \cdot \Delta t$$

- Je nach Wahl der Achse lassen sich verschiedene Bewegungsmuster erzeugen.



4.2 Transformationen

5. Dämpfung (z.B. zur Simulation von Luftwiderstand)

- Ziel: Geschwindigkeit passt sich (langsam) der Umgebung an (! threshold (Schwellwert))
- Abschwächungs Parameter d:
d>0: Das Partikel wird bis auf die threshold Geschwindigkeit abgebremst.
(d<0: Das Partikel wird bis auf die threshold Geschwindigkeit beschleunigt.)

$$\vec{V}' = \vec{V} \cdot \max\{1.0 - d\Delta t, \min\{1.0, \frac{threshold}{|\vec{V}|}\}\}$$

4.3 Motion Blur

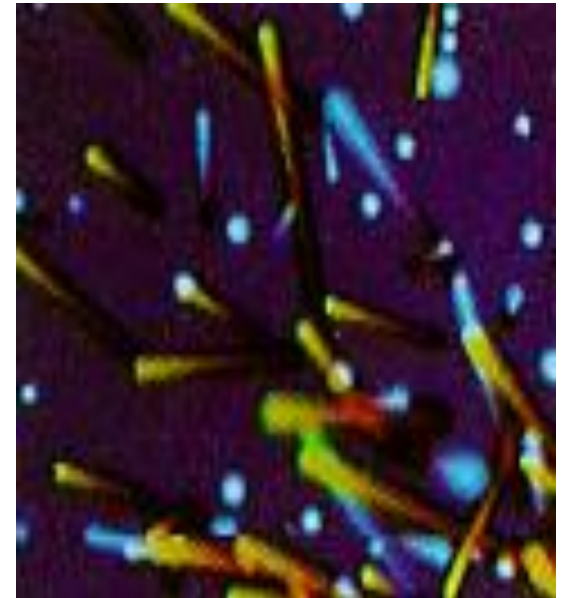
- Entsteht beim Fotografieren, bei langer Belichtung bewegter Szenen.
- Spezielle Technik der Computergraphik um Bewegung zu suggerieren und bewegte Bilder flüssiger erscheinen zu lassen.



4.3 Motion Blur

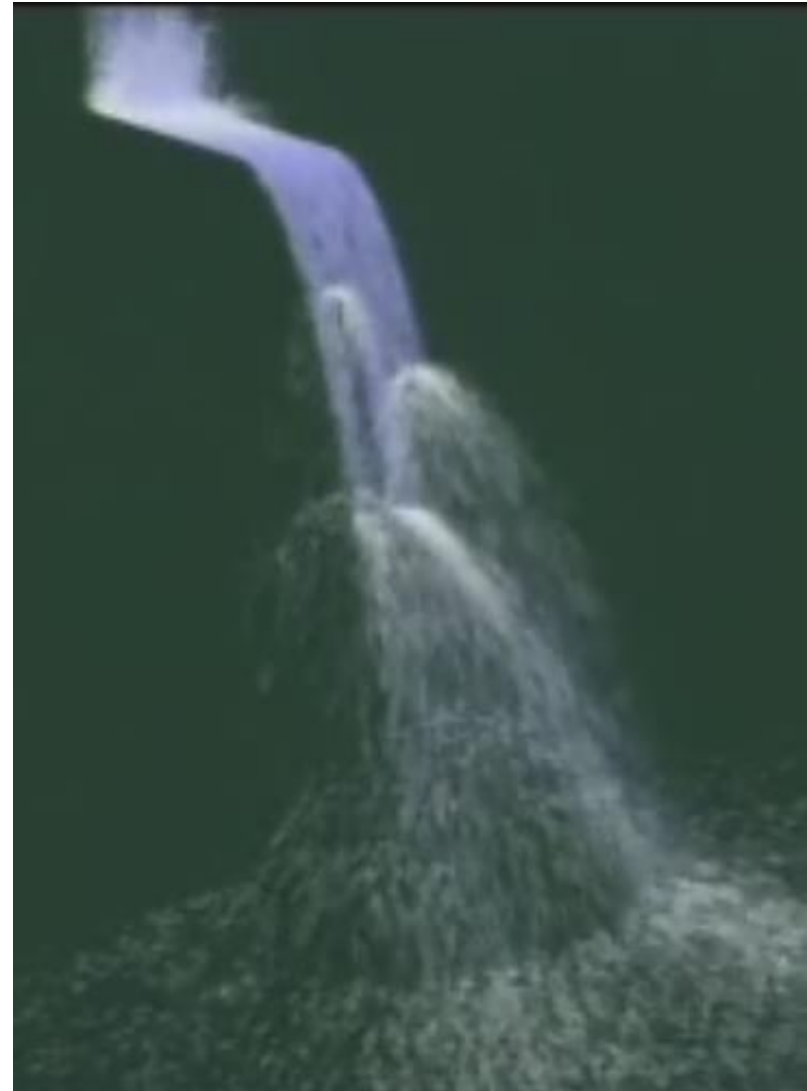
Motion Blurring in Partikel Systemen:

- Jedes Partikel hat aktuelle und letzte Position gespeichert.
- Zwischen diesen Positionen werden Partikelfarbe, Form und Transparenz interpoliert
- Die Eigenschaften zwischen aktueller und letzter Position können gezielt geändert werden um spezielle Effekte zu erzeugen



4.3 Motion Blur

- Bei der Simulation des Wasserfalls wird motion blur verwendet um das „Fließen“ des Wassers hervor zu heben. Es wird zwischen aktueller und letzter Position interpoliert.



Aus Sims "Particle Dreams"

Partikelsysteme – Sonstiges

- Die meisten Animationssysteme verfügen über Partikelsysteme (Maya, Blender, 3D Studio Max, ...)
- Partikelsysteme kann man gut parallelisieren, insbesondere kann man die Graphik-Hardware nutzen für effiziente Implementierungen

siehe dazu:

- A. Kolb, L. Latta, C. Rezk-Salama: *Hardware-Based Simulation and Collision Detection for Large Particle Systems*. Graphics Hardware 2004
<http://www.cg.informatik.uni-siegen.de/data/Publications/2004/GH2004.pdf>