

10 Exkurs III: Modellierung

Inhalt

10.1 Flächenbasierte Modelle	10-2
10.2 Volumenbasierte Modelle	10-6
10.2.1 Operationen mit bestehenden Volumina	10-8
10.2.2 Definition von Volumina	10-9
10.2.3 Speicherung von Volumina	10-11
10.3 Prozedurale Modelle	10-15
10.3.1 Fraktale Modelle	10-16
10.3.2 Objekterzeugung mit Grammatiken	10-19
10.3.3 Partikelsysteme	10-24

Ziel: der jeweiligen Anwendung angepasste Repräsentation von Objekten

10.1 Flächenbasierte Modelle

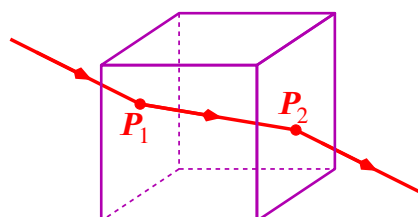
Motivation: Wenn die Objekte nur dargestellt werden sollen, ist die Beschreibung durch ihre Oberfläche ausreichend.

Information über die Zusammengehörigkeit mehrerer Flächenstücke zum selben **Körper** werden nicht benötigt.

Beispiel 10.1: Für rekursives Raytracing wird der Würfel durch sechs Quadrate (jeweils mit Materialangabe für die **Außen- und Innenseite**) beschrieben.

Das vom Würfel belegte **Volumen** ist nur **implizit definiert**:

- Durchdringt der rückverfolgte Strahl bei P_1 eines der sechs Quadrate **von außen nach innen**, so wird angenommen, dass der Strahl bis zum nächsten Schnittpunkt P_2 innerhalb des Würfels verläuft.



- ⊕ Objekte effizient speicher- und darstellbar
- ⊖ Die Oberfläche der Objekte muss (i. Allg. vom Anwender!) in „einfache“ Teile (**Patches**) zerlegt werden:

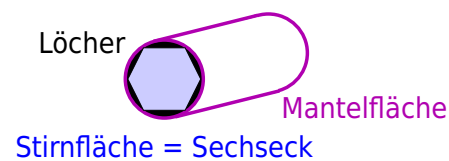
- Polygone
- Quadriken (Kugeloberflächen, ...)
- parametrisierte Flächen $\mathbf{P} = \mathbf{P}(s, t)$

$$\text{- Splineflächen } \mathbf{P} = \mathbf{P} \left(\underbrace{\mathbf{P}_i}_{\text{Steuerpunkte}}, \underbrace{\omega_i}_{\text{Gewichte}} \right)$$

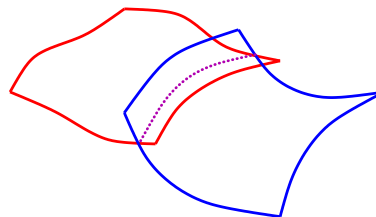
Aufwand
Flexibilität

- ⊖ Konsistenz muss explizit gesichert werden.

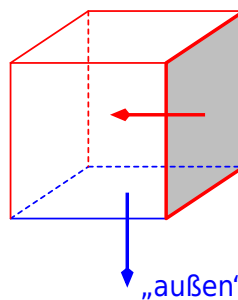
- Der Rand des Körpers muss vollständig überdeckt sein.



- Die Patches müssen an den „Nahtstellen“ zusammenpassen.

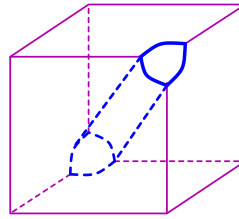


- Die „innen“/„außen“-Information muss konsistent sein.



⊖ Operationen mit Körpern sind nur eingeschränkt möglich:

„Bohre ein zylinderförmiges Loch durch den Würfel.“



⇒ Oberfläche muss (explizit) wieder zerlegt werden

- Volumenabhängige Information (z. B. Gewicht eines Werkstücks) ist nur mit großem zusätzlichem Aufwand zu beschaffen.

10.2 Volumenbasierte Modelle

Motivation: Beim Entwurf von Maschinen usw. sind die Objekte die Körper selbst, nicht Teile ihrer Oberflächen.

⇒ Der vom Objekt ausgefüllte Raum sollte aus der Darstellung des Objekts im Raum hervorgehen.

Forderungen an das Modell:

Mächtigkeit: Die für die jeweilige Anwendung benötigten Objekte sollten (leicht) erzeugbar sein.

Eindeutigkeit: Jedes Objekt sollte eine eindeutige Darstellung besitzen.

(⇒ einfacher Test, ob zwei Objekte gleich sind)

Exaktheit: Die „üblichen“ Objekte sollten ohne Approximation darstellbar sein.

(z. B. Kugel nicht durch Polyeder approximieren)

Intuitivität: Es sollte möglich sein,

- aus bereits konstruierten Objekten andere zusammenzusetzen,
- Stücke aus Objekten „herauszuschneiden“,
- usw.

Konsistenz: Es sollte unmöglich (oder zumindest sehr schwierig) sein, unsinnige Objekte (z. B. nicht geschlossene Körper) zu konstruieren.

Effizienz: Die Darstellung sollte möglichst einfache

- Bilderzeugung,
- Gewinnung relevanter Information (z. B. ein CNC-Programm zum Fräsen des Werkstücks) erlauben.

Anwendungen:

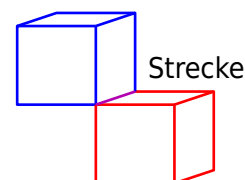
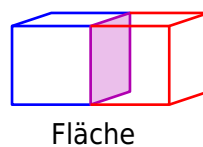
- Entwurf von Werkstücken
- Medizin: Volumenmodell von Organen (gewonnen aus Computer-Tomografie-Daten)
 - Volumen eines Tumors,
 - günstiger Zugang für Operation, ...

10.2.1 Operationen mit bestehenden Volumina

gewünschte Operationen:

- Zusammensetzen von Objekten
(\triangleq Vereinigung)
- „Abschneiden“ von Teilen eines Objekts
(\triangleq Durchschnitt, Mengendifferenz)

Problem: Diese Operationen liefern nicht immer ein Volumen:



Abhilfe: **regularisierte Mengenoperationen:**

$$A \text{ op}^* B := \overline{(A \text{ op } B)}^\circ$$

op: \cup , \cap , \setminus

$(\dots)^\circ$: Inneres

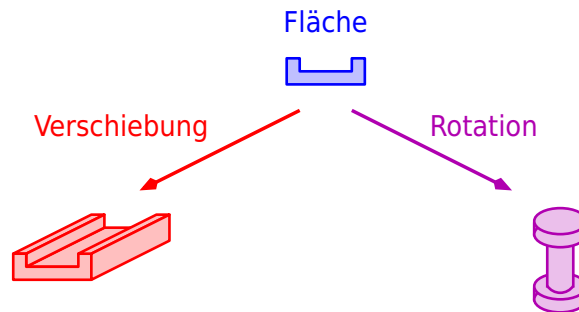
$\overline{(\dots)}$: Abschluss

10.2.2 Definition von Volumina

1. Transformiere gegebene Volumina (z. B. Würfel, Kugel) auf die gewünschte Form.

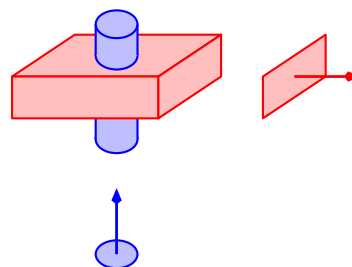
Die vorgegebenen Bausteine sind oft noch parametrisiert, z. B. „Rad mit n Speichen“ (n kann bei jedem Aufruf unterschiedlich festgelegt werden).

2. **Sweep**: Bewege eine Fläche oder ein Volumen im Raum und erzeuge dadurch ein neues Volumen.



Bemerkungen 10.2:

1. Oft werden auch Bewegungen entlang von Raumkurven zugelassen sowie die Möglichkeit, die Fläche während der Bewegung zu ändern.
2. Mengenoperationen mit Sweeps ergeben i. Allg. nicht wieder Sweeps:



⇒ Sweeps können nicht immer in der Form

erzeugendes Element + Bewegung

gehandhabt werden.

3. Mit Sweeps können beispielsweise die von einem Fräskopf aus einem Werkstück geschnittenen „Bahnen“ modelliert werden.
4. Bewegung einer Fläche liefert nicht immer ein Volumen!

10.2.3 Speicherung von Volumina

Randspeicherung

(Boundary representations, **b-reps**)

Das **Modellierungssystem** speichert Volumina in Form ihrer Oberflächen.

Nach Mengenoperationen o. ä. wird die Oberfläche des resultierenden Volumens **automatisch** in einfache Teile zerlegt.

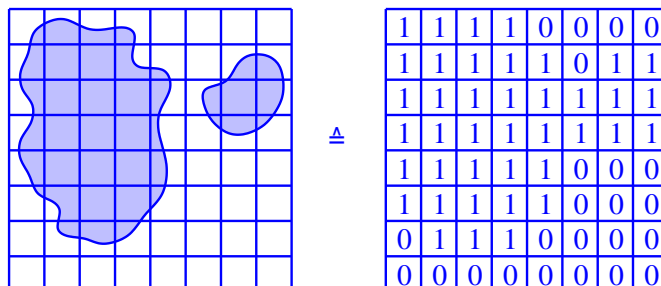
- ⊕ Die Oberflächen können direkt zur Bilderzeugung verwendet werden.
 - ⊖ relativ aufwändige Algorithmen zur Verwaltung der Oberflächen notwendig (Bestimmung von Schnittkurven, ...)
- ⇒ Systeme oft auf Volumina mit polygonalen Oberflächen beschränkt

Volumenaufzählung

Zerlege den Raum in kleine Zellen und zähle die vom Objekt (teilweise) überdeckten Zellen auf:

1. Zellraster-Unterteilung:

- Alle Zellen sind gleich groß.
 - mögliche Zustände jeder Zelle: „belegt“/„nicht belegt“
- $\underbrace{\qquad}_{\triangleq 1} \qquad \underbrace{\qquad}_{\triangleq 0}$

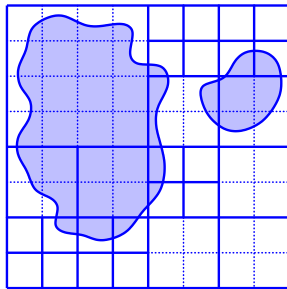


- Speicherung:
 - durch Aufzählen der belegten Zellen oder
 - als Bitfeld

⇒ Mengenoperationen werden sehr einfach und effizient.

2. Octrees:

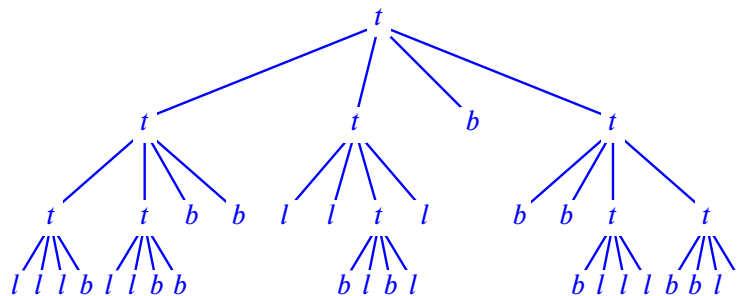
- Hierarchie von Zellen.
- mögliche Zustände: „belegt“ / „teilweise belegt“ / „leer“
 $\underbrace{\hspace{1cm}}_b \quad \underbrace{\hspace{1cm}}_t \quad \underbrace{\hspace{1cm}}_l$



ergibt bei Quadrantennummerierung

3	4
1	2

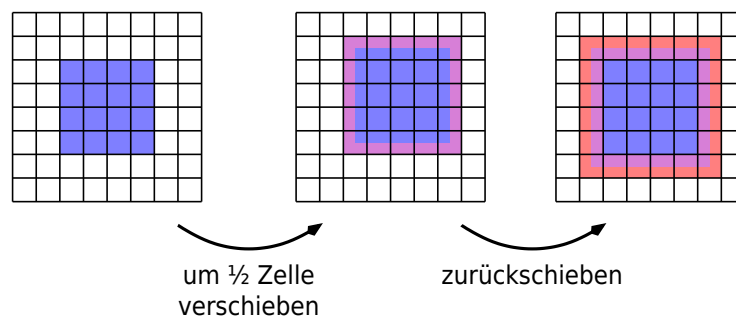
 den Quadtree



- Mengenoperationen einfach durchführbar (Durchlaufen der beiden Bäume)

Bemerkungen 10.3:

1. Volumenaufzählung benötigt i. Allg. viel mehr Speicherplatz als die Randspeicherung.
2. Die Objekte werden i. Allg. nur approximiert.
3. Ohne Vorsichtsmaßnahmen sind die Volumina **nicht bewegungsinvariant**:



(ähnlich bei Rotationen)

4. Vor der Bilderzeugung muss die **Oberfläche** des Körpers **approximiert** werden (die Wände der „Randzellen“ ergäben ein „Lego“-Bild).

10.3 Prozedurale Modelle

Motivation: Enthält eine Szene sehr viele Details, z. B.

- „realistisches“ (nicht-glattes) Terrain,
- Bäume mit Blättern,
- dreidimensionale Textur,
- ...,

so ist es i. Allg. nicht mehr möglich oder sinnvoll, diese Information durch explizite Vorgabe aller entsprechenden Objekte einzubringen.

Ein Terrain kann beispielsweise durch ein „Dreiecksnetz“ approximiert werden.

typische Größenordnung:

$\gg 10^6$ Dreiecke,

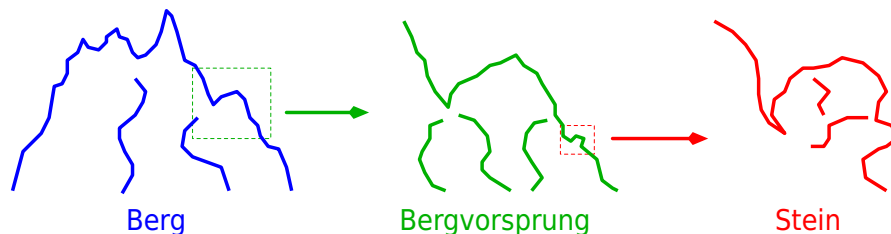
falls auch Bildvergrößerung möglich sein soll

Ansatz: Gib statt der Objekte eine Vorschrift an, mit der die Objekte „bei Bedarf“ erzeugt werden können.

10.3.1 Fraktale Modelle

Beobachtung: Viele in der Natur vorkommenden Formen sind zu einem gewissen Grad **selbstähnlich**, d. h. Teile der Objekte sehen aus wie verkleinerte Kopien der Objekte selbst.

Beispiel: Gebirge



Je näher man kommt, desto kleinere (im Wesentlichen der ursprünglichen Form ähnelnde) Details werden sichtbar.

Ansatz: Starte mit einer groben Form und füge nach und nach immer kleinere Details (vom selben „Typ“) hinzu.

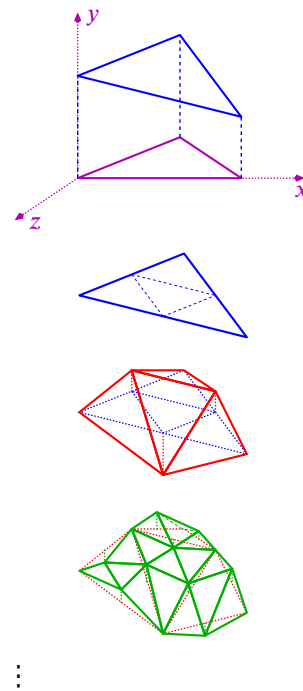
Beispiel 10.4: Erzeugung eines Bergs:

Starte mit einem „beliebigen“ Dreieck.

Zerlege das Dreieck an den Seitenmittelpunkten in vier kongruente Teile.

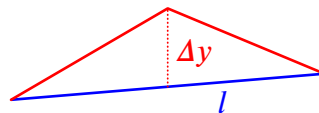
Verschiebe jeden der drei neuen Punkte um einen zufälligen Betrag in y -Richtung.

Wiederhole diese beiden Schritte rekursiv für jedes der entstandenen vier Dreiecke.

**Bemerkungen 10.5:**

1. Die Verschiebung Δy hängt von der Länge der gerade unterteilten Dreiecksseite ab:

$$\Delta y = \underbrace{\text{rnd}}_{\substack{\text{Zufallszahl} \\ \in [-1; 1]}} \cdot \underbrace{f(l)}_{\substack{\text{streng monoton steigend,} \\ \text{z. B. } f(l) = l^\alpha, \alpha \in (0; 1]}}$$



2. Die Rekursionstiefe hängt vom Abstand zum Betrachter ab: je größer die Entfernung, desto weniger Details sind notwendig
(Abbruch z. B., wenn Dreiecke etwa Pixelgröße besitzen)
3. Die Dreiecke werden i. Allg. nicht explizit gespeichert, sondern bei Bedarf neu erzeugt (dabei wird jeweils nur ein Dreieck weiter unterteilt).

⇒ Die Verschiebungen (d. h. die Zufallszahlen) müssen **reproduzierbar** sein!

10.3.2 Objekterzeugung mit Grammatiken

Beobachtung: Ein Zweig **z** mit einer Knospe **K** an der Spitze kann sich auf verschiedene Arten entwickeln:

1. Die Knospe kann absterben.
2. Die Knospe kann eine Blüte **B** erzeugen und dann absterben.
3. Die Knospe kann ein Blatt **I** erzeugen.
4. Die Knospe kann austreiben:
 - Neben der Knospe entstehen evtl. eine oder zwei weitere Knospen.
 - Die ursprüngliche Knospe wächst weiter zu einem Zweig.
 - An der Spitze dieses Zweigs ist wieder eine Knospe.

Abstraktion: Beschreibe diese Zustandsveränderungen durch eine (kontextfreie) „**Grammatik**“.

Nichtterminale: **K, B**

Terminale: **z, I, (,), [,]**

Produktionen: **$K \rightarrow K \mid \varepsilon$** ($\varepsilon \triangleq$ „nichts“)

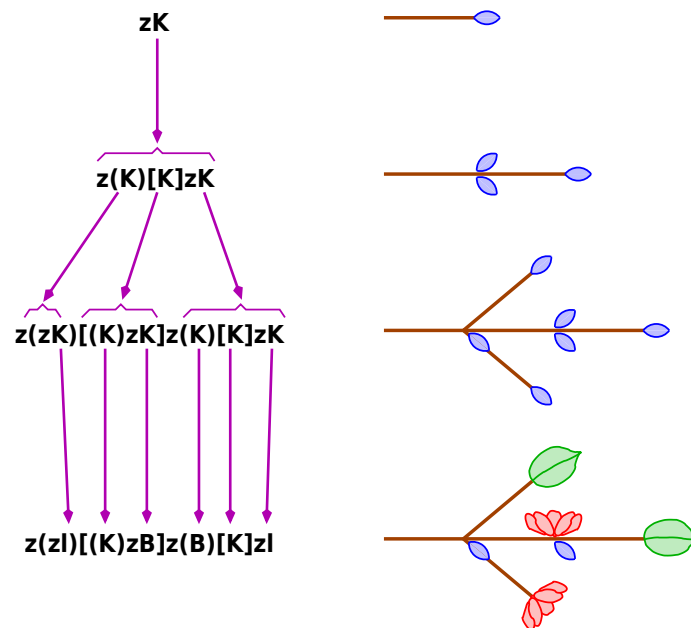
$K \rightarrow B$

$B \rightarrow B \mid \varepsilon$

$K \rightarrow I$

$K \rightarrow zK \mid (K)zK \mid [K]zK \mid (K)[K]zK$

Startwort: **zK**

Beispiel 10.6: Eine mögliche Ableitung:**Bemerkungen 10.7:**

1. Beim Übergang von einer Generation zur nächsten wird auf jedes Nichtterminal im Wort irgendeine der passenden Produktionen angewandt.
2. Produktionen der Form $\mathbf{B} \rightarrow \mathbf{B}$ können entfallen, wenn nur das Endergebnis interessiert, seine zeitliche Entwicklung aber ohne Belang ist.
3. Man kann eine beliebige (i. Allg. kontextfreie) Grammatik und eine **Interpretation** vorgeben, z. B.:

$(\dots) \triangleq$ Verzweigung nach links mit Inhalt \dots

4. Die Interpretation eines Symbols muss nicht positionsunabhängig sein, z. B.:

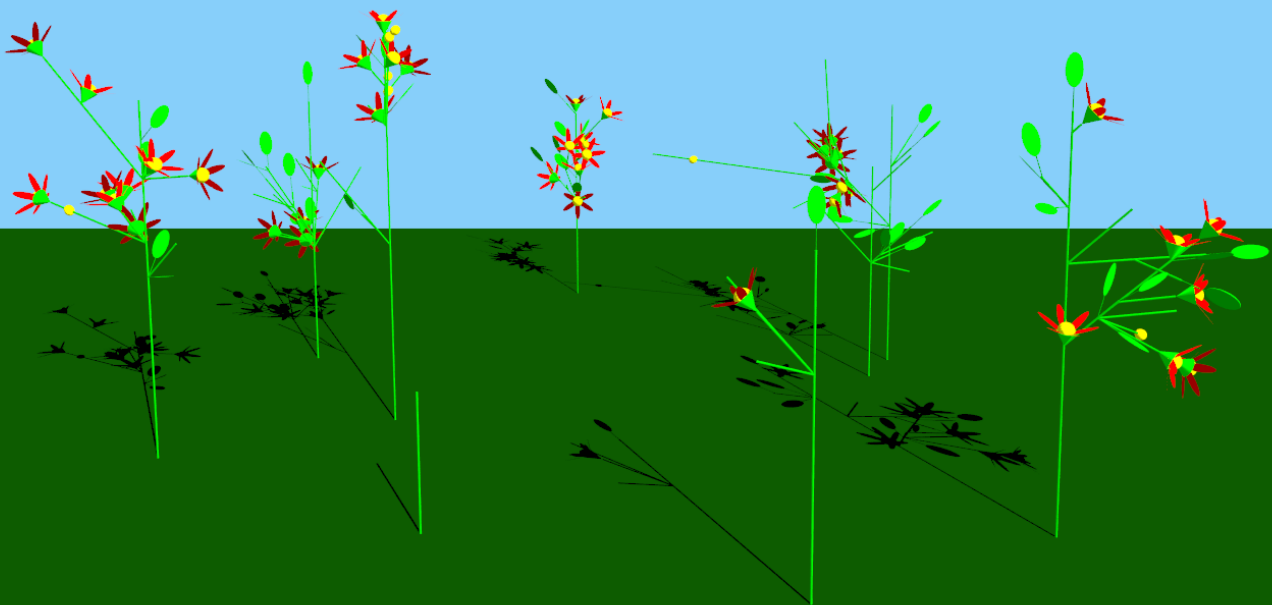
Zweige, die in späteren Generationen entstehen, kürzer machen

5. Auswahl der Produktionen mit vorgegebenen Wahrscheinlichkeiten (und ggf. nach zusätzlichen Regeln)
6. Die Objekte können vorab oder bei Bedarf erzeugt werden.

Für das folgende Beispielbild wurde einige Erweiterungen vorgenommen:

- Übergang auf 3D
- damit Verzweigungen in verschiedene Raumrichtungen
- Auswahl zwischen mehreren Produktionsregeln durch Zufallszahlen, dabei Anpassung der Zufallssteuerung an Fortschritt des Wachstums
- Erweiterung der Grammatik: Speicherung von Winkeln und Zweiglängen innerhalb der Ableitungsstrings

Bilder/blumen/blumen03
Quelle: Programm: Paula Denk,



10.3.3 Partikelsysteme

Ziel: Modellierung von „Objekten“, die nicht adäquat durch eine Oberfläche oder ein Volumen beschrieben werden können, etwa weil sich ihre Topologie im Laufe der Zeit stark ändert

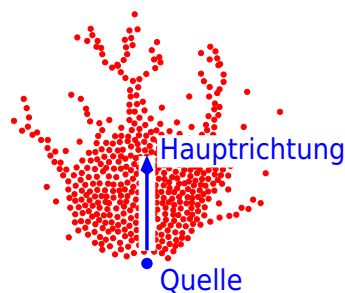
Ein **Partikelsystem** ist eine Menge von Teilchen, das sich (durch Anwenden physikalischer Regeln auf die Teilchen) im Laufe der Zeit verändert. Die Teilchen können

- ihre Eigenschaften (z. B. Geschwindigkeit, Farbe) ändern,
- neue Teilchen erzeugen,
- selbst verschwinden.

Anwendung z. B. bei der Modellierung von

- Funkenflug bei Feuer, Explosionen, Feuerwerk,
- Nebel,
- Laub dichter Bäume.

Beispiel 10.8: Ioderndes Feuer



$\text{Richtung}(\text{Teilchen } i, \text{Zeitpunkt } t + 1) = \text{Richtung}(i, t) + \text{Einfluss der Schwerkraft} + \text{Zufallsstörung}$

Farbe der Teilchen geht mit der Zeit von Weiß(glühend) (entstehen) über Rot nach Schwarz (verschwinden)

Problem: Partikelsysteme umfassen oft mehrere Millionen von Teilchen.

⇒ Partikelsysteme können nicht problemlos in die „Standardverfahren“ zur Bilderzeugung (z. B. rekursives Raytracing) eingebunden werden.

Ansatz: Bestimme direkt den Beitrag jedes Teilchens zum Bild.

hier: Verfolge jedes Teilchen auf einem kurzen Stück seiner Flugbahn und erhöhe die Intensität aller von dieser Strecke betroffenen Pixel.

Auswirkungen:

- Die Teilchen gehen nicht in die Verdeckungsanalyse der übrigen Objekte ein (d. h. durch andere Objekte verdeckte Teilchen müssen nach anderen Kriterien entfernt werden).
- Der Einfluss der Teilchen auf die übrigen Objekte (hier: Beleuchtung durch das Feuer) bleibt unberücksichtigt.
(Abhilfe hier: Bewegliche Lichtquelle variabler Intensität im Feuer positionieren)

Bemerkungen 10.9:

1. Für jede neue Anwendung der Partikelsysteme (Feuer, Nebel, ...) muss i. Allg. ein neuer Darstellungsalgorithmus (mit vielen heuristischen Ansätzen) entwickelt werden.
 2. Trotz der vielen Heuristik oft überzeugende Effekte.
-