

## Animating Many Rigid Bodies

- Timewarp Rigid Body Simulation (Mirtich)
- Optimization Based Animation (Milenkovic & Schmidl)

Seminar „Physically-based methods for 3D games and medical applications“

Wintersemester 02/03  
Isha Geigenfeind

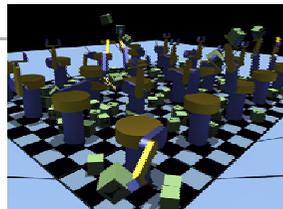
## Motivation

- Rigid Body Simulation (RBS) ist eine fortgeschrittene Technologie:
    - schnelle, robuste Collision Detection
    - viele Methoden zur Einhaltung von Motion- und Non-Penetration-Constraints
    - verfügbar in vielen Modellierungspaketen
    - wird in vielen Simulationen & Computerspielen benutzt
  - Immer noch Potential zur Verbesserung:
    - Schon gesehen: Methoden für Echtzeitsimulationen
- Heute: Methoden zur Simulation/Animation von **vielen** Rigid Bodies

2

## Motivation

- Anwendungen...
  - Physikalische Simulationen (z.B. in der Robotik)
- Animationen (z.B. in der Unterhaltungsindustrie)



3

## Inhalt

- Motivation
- Rigid Body Dynamics
- Zwei Paper zu diesem Thema
  - "Timewarp Rigid Body Simulation" 2000 - Brian Mirtich - MERL
  - "Optimization Based Animation" 2001 - Victor Milenkovic, Harald Schmidl - University of Miami
- Kurzer Vergleich

4

## Rigid Body - Wichtige Grössen

- konstant:
  - Masse  $M$
  - Trägheitstensor  $I_{\text{body}}$  bzw.  $I_{\text{body}}^{-1}$
- variabel:
  - $x_{\text{CM}}$
  - $v_{\text{CM}}$
  - $R$
  - $\omega$
  - $P = Mv$
  - $L = I\omega$ ;
  - $I = R^T I_{\text{body}} R$

5

## Dynamik

- Gegeben:
  - Kraft  $F$
  - Drehmoment (Torque)  $\tau$
- Differentialgleichung:

$$\frac{d}{dt} \begin{pmatrix} x(t) \\ R(t) \\ P(t) \\ L(t) \end{pmatrix} = \begin{pmatrix} v(t) \\ \omega(t) * R(t) \\ F \\ \tau \end{pmatrix}$$

- Sehr einfach!
- Schwieriger Teil: Collision Detection / Collision Resolution

6

## Kollision

- Collision Detection:
  - Gibt es 2 Körper die sich überlappen / berühren?
  - Wenn ja, wo?
- Collision Resolution:
  - Penalty Methoden: Berechne Penalty Force welche die Körper wieder auseinanderdrückt. Einfach aber inexakt und instabil.
  - analytische Methoden: Finde exakten Kollisionszeitpunkt.
    - falls Relativgeschw. negativ ("colliding contact"), berechne nötige (Dreh-)Impulsänderung beider Körper.
    - falls Relativgeschw. null ("resting contact"), berechne Kontaktkräfte  $f$ , typischerweise durch Lösen eines LCPs (Linear Complementarity Problem).

7

## Timewarp Rigid Body Simulation

- SIGGRAPH 2000 - Brian Mirtich – MERL
- Allgemein verwendbar für verschieden anspruchsvolle Rigid Body Simulationen.
- Basiert auf D. Jefferson's Timewarp Algorithmus.
- Zeigt den Weg auf für eine parallele Implementierung.
- Verbesserung auch für Einprozessorsysteme

8

## Probleme bei RBS

- RBS ist zwar ein kontinuierlicher Prozess, hat aber diskrete Eigenschaften.
- Kollisionen zweier Körper führen zu Unstetigkeiten
- Bei Unstetigkeiten muss Integration angehalten werden
- Kollisionen z.T. sehr schwer vorherzusagen.
- Zwei verbreitete Möglichkeiten:
  - Retroactive Detection
  - Conservative Advancement

9

## Probleme bei RBS (2)

- Retroactive Detection:
  - Nach jedem Zeitschritt wird überprüft ob es eine Diskontinuität gegeben hat, (z.B. Kollision).
  - Falls ja, wird der Zeitpunkt  $t$  der Diskontinuität gesucht (z.B. durch Bisektion des Zeitintervalls)
  - Alle RBs werden auf den Zeitpunkt  $t$  gebracht, obwohl die Diskontinuität nur wenige RBs beeinflussen könnte
  - Dann wird Integration wieder gestartet
  - Bei vielen RBs gibt es u.U. sehr viele Diskontinuitäten!

10

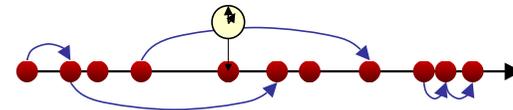
## Probleme bei RBS (3)

- Conservative Advancement:
  - Integriere immer nur soweit, dass garantiert keine Diskontinuität vorkommt.
  - Benötigt untere zeitliche Schranke für nächste Diskontinuität
  - Bei vielen RBs ist die Schranke ungenau, Integration muss sehr oft angehalten werden

11

## Timewarp Algorithmus

- Viele Rigid Bodies → Viel unnötige Synchronisation
- Vergleiche parallele diskrete Ereignissimulationen (DES)
- Einfache DES

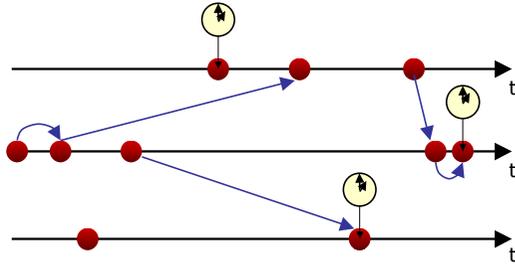


- Effizienter Algorithmus für parallele DES: "Timewarp" (Jefferson 1985)

12

## Verteilte DES

- verteilte DES:



- asynchron!

13

## Timewarp Algorithmus (für DES)

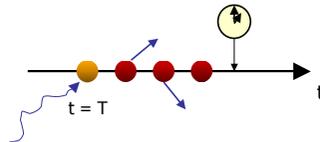
Jeder Prozess...

- ...unterhält einen Teilzustand.
  - ...hat eigene *local virtual time* (LVT).
  - ...verschickt Ereignisse
  - ...stempelt sie mit der Zeit der Ereignisse
  - ...ordnet Ereignisse in einer Input Queue
  - ...arbeitet Ereignisse ab.
  - ...erhöht seine LVT
- Optimistische Strategie!

14

## Timewarp Algorithmus (für DES)

Was passiert wenn ein Ereignis mit Zeitstempel  $T < LVT$  eintrifft?



- der Zustand bei T wird wiederhergestellt ("Rollback")
- jeder nach T versendeten Message wird eine *Antimessage* hinterhergeschickt
- Messages und Antimessages löschen sich in der Queue aus

15

## Timewarp Algorithmus

Wie wird Rollback implementiert?

- Frühere Zustände werden gespeichert

Verbraucht das nicht den Speicher im Laufe der Zeit?

- Def.: *global virtual time* (GVT) =  $\min(\text{LVTs, Zeitstempel in noch nicht verarbeiteten Messages})$ 
  - GVT ist untere Grenze für Rollbacks
  - GVT nimmt monoton zu
  - nur Zustände & Messages mit  $t > GVT$  müssen gespeichert werden

16

## Timewarp mit Rigid Bodies

- Rigid Body = Timewarp Prozess
- Kollisionen = Kommunikation
- 4 Event-Typen
  - *Collision Check Events* zwischen je **zwei** Rigid Bodies
  - *Group Check Events* zwischen kontaktierenden Rigid Bodies.
  - *Redraw Events* schreiben Position eines RB in einen Puffer
  - *Callback Events* führen benutzerdefinierte Aktionen aus
- Pro Collision & Group Check Event wird eine *safe time* mitgeführt

17

## Timewarp mit Rigid Bodies

- GVT = Minimum aller safe times und aller LVTs
- Nach Collision Resolution werden die Zustände zweier Prozesse verlinkt. Es entsteht ein *State Graph*.
- Antimessages werden auf dem Graph abgearbeitet:

18

## Timewarp mit Rigid Bodies

- Nachher:

19

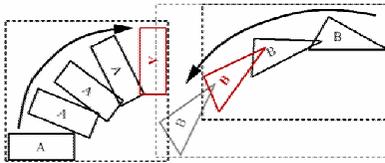
## Kontaktgruppen

- Kontaktierende RBs werden zu Kontaktgruppen zusammengefasst.
- Jede Kontaktgruppe hat eigenen Prozess.
- Kontaktgruppen können sehr dynamisch sein:

20

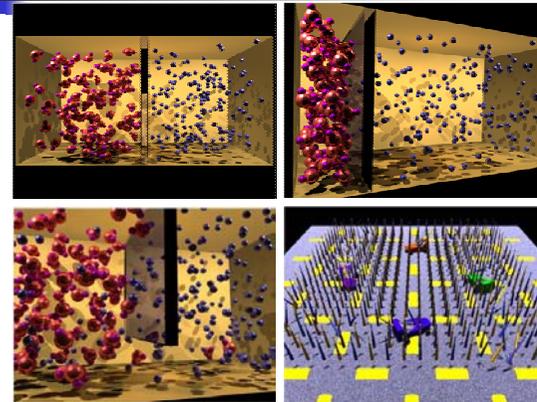
## Collision Checks

- Ständige Collision Checks zwischen allen möglichen Paaren ineffizient
- Nur Collision Checks zwischen *Active Pairs*
- Berechne Swept Volume (Axis-Aligned Bounding Boxes) über einzelnen Rigid Body
- Collision Check Events falls Überlappung



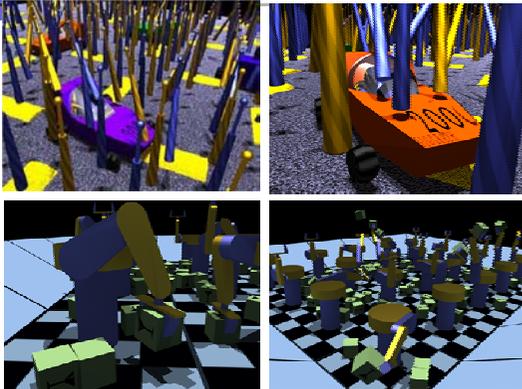
21

## Resultate



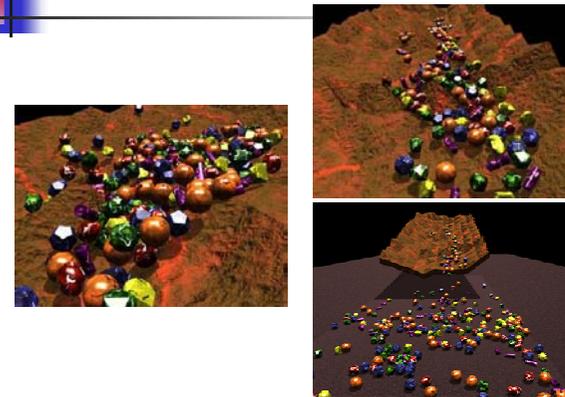
22

## Resultate



23

## Resultate



24

## Echt parallele Implementation?

- Master/Slave: einfacher, aber möglicher Flaschenhals
- Alle gleichberechtigt:
  - Skaliert besser
  - Aber wie verteilt man Rechenaufwand am besten?
  - Wie minimiert man nötige Kommunikation?
  - Rollback braucht jetzt echte Antimessages!
  - Wie verteilt man gemeinsame Datenstrukturen?
  - ...und Kontaktgruppen?
  - ...etc.

25

## Optimization Based Animation

- SIGGRAPH 2001 – Victor Milenkovic, Harald Schmidl – University of Miami
- Für viele Anwendungen reicht eine visuell plausible Bewegung der Körper aus
- Bei vielen Rigid Bodies auf einmal wird viel Rechenzeit aufgewendet für Bewegungen, die das menschliche Auge gar nicht sieht

26

## Optimization Based Animation

- OBA benutzt einen festen Zeitschritt  $\Delta t$
- *target position* = Zustand nach  $\Delta t$  ohne Kollisionbehandlung
- Idee: Bewege jeden Körper so nah wie möglich an seine target position ohne dass es Überlappung gibt
- führt auf ein Optimierungsproblem
- Kollisionen werden am Ende des Zeitschritts synchronisiert
- Bewegung der RB wird desynchronisiert

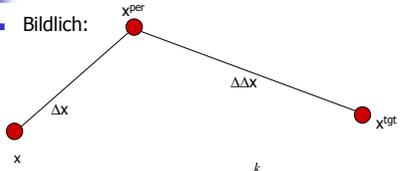
27

## Zielfunktion (Objective)

- Jeder Body hat...
- ...initiale Position  $x$  und Orientierung  $R$
  - ...Targetposition  $x^{tgt}$  und -orientierung  $R^{tgt}$
  - ...(zur Minimierung) eine perturbiertes  $x^{per}$  und  $R^{per}$
  - $\Delta x = x^{per} - x$  und  $\Delta R = R^{per}R^{-1}$
  - $\Delta x^{tgt} = x^{tgt} - x$  und  $\Delta R^{tgt} = R^{tgt}R^{-1}$
  - statt Matrix  $R$ , Vektorrepräsentation der Rotation durch  $r$
  - *target deficit*:  $\Delta \Delta x = \Delta x - \Delta x^{tgt}$  und  $\Delta \Delta r = \Delta r - \Delta r^{tgt}$
  - gesucht: benötigtes  $\Delta x$ ,  $\Delta r$

28

## Zielfunktion (Objective)

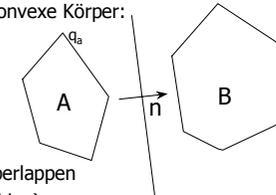
- Bildlich:
 
- Zielfunktion für k Bodies:  $\sum_{i=1}^k \Delta\Delta x_i \cdot \Delta\Delta x_i + D_i^2 \Delta\Delta r_i \cdot \Delta\Delta r_i$
- Alternativ:  $\sum_{i=1}^k m_i \Delta\Delta x_i \cdot \Delta\Delta x_i + \Delta\Delta r_i^T I_i \Delta\Delta r_i$
- $D_i$  ist ein Gewichtungsfaktor;  $I_i$  Trägheitstensor

29

## Constraints

- Separating Plane Constraint für konvexe Körper:

$$\exists n \exists d \quad \forall q_a \in A, \quad n \cdot q_a \leq d \\ \forall q_b \in B, \quad n \cdot q_b \geq d$$



- nur für RBs die sich am Target überlappen
- ist **nichtlinear** (n und q sind Variablen)
- Optimierung im schlimmsten Fall NP-hart
- Durch Linearisieren der Constraints entsteht ein QP, das in polynomieller Laufzeit gelöst werden kann
- Durch Lösen dieses QP nähert man sich einem lokalen Minimum
- Finden des Minimums durch iteriertes Lösen des QP.

30

## Linearisieren der Constraints

- Lineare Approx. der Normalen n:
  - $n^{\text{per}} \approx n + \delta_x n_x + \delta_y n_y$ ,  $n_x, n_y, n$  orthonormale Vektoren
- $q_a^{\text{per}}$  linearisiert durch  $\Delta x$  und  $\Delta r$  ausgedrückt:
  - $q_a^{\text{per}} \approx q_a + \Delta x + \Delta r \times (q_a - x_a)$
- Eingesetzt in Separating Plane Constraint
- $(n + \delta_x n_x + \delta_y n_y) \cdot (q_a + \Delta x + \Delta r \times (q_a - x_a)) < d$
- ausmultiplizieren, quadratische Terme vernachlässigen, umformen, alle Variablen nach links gibt:

$$n \cdot \Delta x_a - (n \times (q_a - x_a)) \cdot \Delta r_a \\ + (n_x \cdot q_a) \delta_x + (n_y \cdot q_a) \delta_y - d \leq -n \cdot q_a$$

31

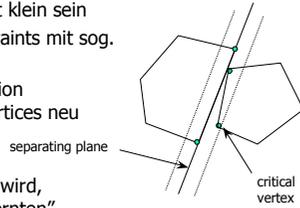
## Zusammenfassung des Problems

- Jeder RB hat 12 Variablen: gesuchte  $\Delta x$ ,  $\Delta r$ , target deficits  $\Delta\Delta x$ ,  $\Delta\Delta r$ .
- Jedes RB-Paar hat 3 gesuchte Variablen, welche Perturbation des Separating Planes ausdrücken:  $\delta_x$ ,  $\delta_y$ ,  $d$
- jeder RB braucht 6 equality constraints, die  $\Delta\Delta x$ ,  $\Delta\Delta r$  durch  $\Delta x$ ,  $\Delta r$  ausdrücken (trivial)
- Die Zielfunktion ist schön quadratisch positiv-definit durch  $\Delta\Delta x$ ,  $\Delta\Delta r$  ausgedrückt
- Die Constraints sind linear
- Dieses QP nun iteriert lösen, bis Perturbation 0 (lok. Min.)
- Am lokalen Minimum sind Linearisierungen korrekt, d.h. Non-Overlapping-Constraints werden eingehalten
- Fertig!

32

## Performance Verbesserung

- "Reckless Dynamic Update":
  - bis jetzt: jeder Vertex  $q_a$  zusammen mit einem Separating Plane liefern ein Constraint.
  - QPs sollten aber möglichst klein sein
  - Idee: Zunächst nur Constraints mit sog. *critical vertices*
  - Falls es dann zur Penetration kommt, die schuldigen Vertices neu hinzunehmen und das nächste QP lösen
  - falls QP dadurch unlösbar wird, Rollback und mit neu "erlernten" Vertices starten



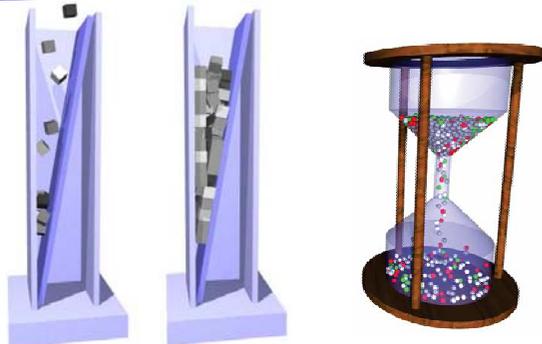
33

## Verschiedenes

- für die Collision Resolution (berechnen der Kontaktpulse / Kontaktkräfte) kann jeder geeignete Algorithmus verwendet werden. (Die Autoren benutzen eine Formulierung als QP).
- Bei schnellen Geschw. zwischen Körpern werden u.U. physikalisch sehr unrealistische Positionen berechnet.
- Mögliche Abhilfe: ein hybrider Algorithmus. Zum Berechnen der Target Positions wird zwischen je 2 Körpern eine beschränkte Anzahl Kollisionen ( $< c_{\max}$ ) berücksichtigt, falls die Geschw. gross genug ( $> v_{\min}$ ) ist.

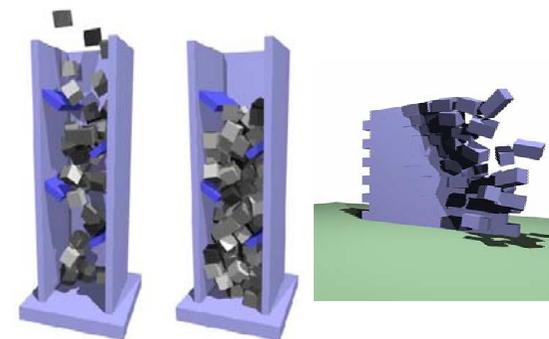
34

## Resultate



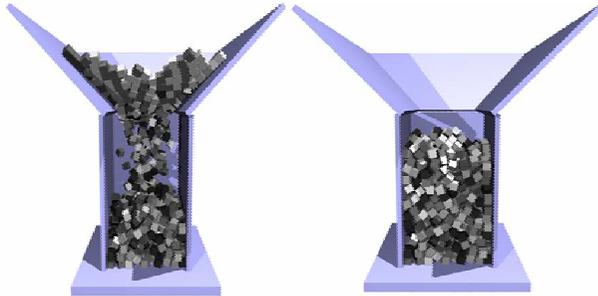
35

## Resultate



36

## Resultate



37

## Kurzer Vergleich

### Timewarp Rigid Body Simulation

- für allgemeine RBS nutzbar
- besonders gut auch für physikalisch exakte
- besonders hilfreich bei lose gekoppelten Anordnungen vieler Körper
- zeigt den Weg in Richtung parallele Implementierung

### Optimization Based Animation

- für Anwendungen wo visuell plausible Animationen ausreichen
- verwendet viele Heuristiken
- geeignet für "dicht gepackte" Anordnungen mit vielen Kontakten
- intensive Verwendung von Optimierungssoftware (QP-Solver)

38

## Quellen

- "Timewarp Rigid Body Simulation"  
Brian Mirtich - 2000  
<http://www.merl.com/papers/TR2000-17/>
- "Optimization Based Animation"  
V.Milenkovic H.Schmidl - 2001  
<http://www.cs.unc.edu/~schmidl/papers/sig2001.pdf>
- "Virtual Time"  
David R. Jefferson - 1985  
<http://portal.acm.org/citation.cfm?id=3988&dl=ACM&coll=portal>

39

## Quellen

Auch nützlich:

- Rigid Body Dynamics - Siggraph Course Notes  
David Baraff - 1997  
<http://www-2.cs.cmu.edu/~baraff/sigcourse/>
- Chris Hecker's Rigid Body Dynamics Information  
<http://www.d6.com/users/checker/dynamics.htm>
- Ende – Danke fürs zuhören!

40