

Praxis der Multikernprogrammierung

Zwischenergebnisse 15.12.2016 – Gruppe Nostrum

Manuel Karl, Dominik Kleiser, Marc Leinweber, Nico Mürdter

Institut für Programmstrukturen und Datenorganisation – Programmiersysteme



Ansätze zur Beschleunigung

- Feststellung: hohe Datenparallelität
 - OpenMP
 - CUDA
 - AVX
- C-Threads und Futures erscheinen aufgrund der Datenparallelität nicht sinnvoll
- Beschleunigung der Matrixmultiplikation (Transformation) mit Quaternionen
- (massive) Reduktion der Schnittests durch räumliche Datenstrukturen
 - BVHs
 - kD-Bäume

Aktuelle Implementierung

- Fehlerquellen: FOV, Normalisierung, Rundung, Initialisierung von Member-Variablen, Schnittdistanzen
- Dreiecksschnitt mit Möller-Trumbore-Algorithmus
- Datenstruktur und Schnittberechnung
 - zunächst: vollständige Suche auf allen Primitiven
 - aktuell: KD-Baum mit Objektmedian (Bau in $O(n^2 \log(n))$) und AABBs
- „embarrassingly parallel“: Parallelisierung über Y-Dimension mit OpenMP
- Benchmark-Ergebnisse: Wie viele Masterstudenten braucht man, um einen KD-Baum zu implementieren?

Fazit und weiteres Vorgehen

- Lernkurve aktuell: räumliche Datenstrukturen, 3D-Geometrie
 - KD-Bäume sind „toll“

- Nächste Optionen:
 - Parallelisierung des Datenstrukturaufbaus
 - Ausnutzung der Datenparallelität: CUDA oder AVX?

- Frage: Einsatz von AVX sinnvoll (Compileroptimierung O3)?
- Frage: Raytracing auf GPUs sinnvoll?
 - wenn ja: Datenpartitionierung?