TECHNISCHE UNIVERSITÄT DRESDEN

FAKULTÄT INFORMATIK
INSTITUT FÜR TECHNISCHE INFORMATIK
PROFESSUR FÜR RECHNERARCHITEKTUR
PROF. DR. WOLFGANG E. NAGEL

Großer Beleg

Untersuchung der Parallelisierung des Feldkamp-Davis-Kress-Algorithmus mittels CUDA

Jan Stephan
(Mat.-Nr.: 3755136)

Hochschullehrer: Prof. Dr. Wolfgang E. Nagel
Betreuer: Dr.-Ing. André Bieberle
Dr.-Ing. Guido Juckeland
Matthias Werner

Dresden, 21. März 2017

Inhaltsverzeichnis

1 Einleitung

- Computertomographie ist gesellschaftsrelevant etc.
- FDK-Algorithmus ist wichtiger Bestandteil von CT
- Ausgangssituation am HZDR: uraltes FDK-Programm braucht mehrere Tage für eine Rekonstruktion (schlecht)
- Gesamtziel: Dieses Programm soll durch ein schnelleres und möglichst gut optimiertes abgelöst werden
- Aufgabenvorstellung Großer Beleg

2 Grundlagen

2.1 Der Feldkamp-Davis-Kress-Algorithmus

- Welches Problem wird gelöst und wo tritt dieses Problem auf? -> evtl. kurzer geschichtlicher Überblick zur CT
- Wie funktioniert der FDK-Algorithmus? -> Detaillierungsgrad?
- Wie sieht der Datenfluss aus? -> Detektor, Vorverarbeitung, Wichtung, Filterung, Rückprojektion
- Warum ist dieser Algorithmus besonders gut für Parallelisierung geeignet? -> viele viele Voxel, keine Abhängigkeiten
- Welche Algorithmen gäbe es noch?

2.2 Parallelisierungsansätze

- Welche Ansätze sind seit 1984 (Erscheinungsjahr des FDK-Papers) entstanden? -> Vorstellung der Interessanten
- Welche Vor- und Nachteile haben diese Ansätze? -> Warum brauchen wir etwas neues / warum können wir sie nicht anwenden?

2.3 Die CUDA-Plattform

- Warum GPUs für wissenschaftliche Berechnungen? -> Vergleich zu CPUs / anderen Beschleunigern
- Warum CUDA? Was gäbe es noch? (OpenCL, OpenMP, OpenACC, ...) / Nachteile?
- Das CUDA-Programmiermodell

3 Umsetzung

3.1 Variantenvergleich

- Welche Parallelisierungsstrategien sind denkbar?
- Warum CUDA? Was gäbe es neben CUDA noch?
- Welche Limitierungen (CUDA, Hardware, HZDRismen...) gibt es?
- Warum ist die gewählte Variante die beste?

3.2 Implementierung

- Implementierungsziele: Wartbarkeit, Portabilität, sinnvolle Geschwindigkeit, nutzbar auf Laptop / Workstation / Cluster
- kurze Vorstellung/Erwähnung von GLADOS -> Pipeline-Struktur
- Aufteilung des Volumens in Teilvolumen erklären
- Implementierung der einzelnen Stufen
- besonderer Fokus auf Rekonstruktionsstufe: welche Konstanten und Variablen gibt es, was kann optimiert werden, welche Schwierigkeiten gibt es

4 Leistungsanalyse

- Optimierungsziele vorstellen (Auslastung, Time-to-solution)
- Messungen des Gesamtprogramms (Datentransfers, alle Stufen)
- grobe Messungen der Teilstufen (Wichtung, Filterung, Rückprojektion)
- Ergebnis: Rückprojektion braucht am längsten
- detaillierte Messungen der Rückprojektion (Registerverbrauch, GPU-Auslastung, ...)

5 Fazit

- Faktencheck: Wurden die in der Einleitung genannten Ziele erreicht?

Literaturverzeichnis

Danksagung

Vielleicht die HZDR-Betreuer lieber hier nennen?

Erklärungen zum Urheberrecht

Ich habe wirklich nichts geklaut