

Projektvorstellung: Fourier-Transformation und Parallelisierung

Fabian Sponholz

July 15, 2024

Einführung

- ▶ Ziel des Projekts: Implementierung und Analyse verschiedener Ansätze zur Fourier-Transformation
- ▶ Vergleich von sequenzieller, multi-threaded und OpenCL-beschleunigter Implementierung

Lösung Aufgabe 3: Multi-Threading

- ▶ Parallelisierte Version des Cooley-Tukey FFT-Algorithmus mit C++ und Multi-Threading
- ▶ Nutzung der Bibliotheken `thread` und `mutex`
- ▶ Erheblicher Speed-Up festgestellt: $\sim 3.5\times$ auf 4-Kern-CPU

Lösung Aufgabe 4: OpenCL

- ▶ Implementierung der Diskreten Fourier-Transformation (DFT) mit OpenCL
- ▶ Herausforderungen und Lösungsansätze:
 - ▶ Rekursion in OpenCL nicht möglich, Umstellung auf iterative DFT
 - ▶ Initiale Probleme mit komplexem Kernel und instabilen Ergebnissen
 - ▶ Vereinfachter Kernel brachte deterministische und plausible Ergebnisse

Experimentelle Tests

- ▶ Testdateien: Stille, 420 Hz, Oktavensprung, Zufällige Frequenzen
- ▶ Ergebnisse bei allen Implementierungen konsistent:
 - ▶ Stille: Keine Frequenzen erkannt
 - ▶ 420 Hz: Genauigkeit der Frequenzerkennung
 - ▶ Oktavensprung: Erkennung beider Frequenzen
 - ▶ Zufällige Frequenzen: Erfolgreiche Erkennung mit angepassten Thresholds

Laufzeitanalyse und Vergleich

- ▶ Testsysteme:
 - ▶ System 1: Intel i5-2500K, NVIDIA GTX 1650, 4 GB VRAM
 - ▶ System 2: Intel i5-10400F, Intel Arc A770, 16 GB VRAM
- ▶ Laufzeitanalyse auf verschiedenen Parametern (Blockgröße, Schrittweite)

Laufzeiten Testsystem 1

Programm	512/2	Speedup	512/4	Speedup
Base	538 s	1	274 s	1
Threads	157 s	3.43	77 s	3.56
OpenCL	n/a	n/a	347 s	0.79

Programm	512/8	Speedup
Base	135 s	1
Threads	38 s	3.55
OpenCL	6.5 s	20.77

Laufzeiten Testsystem 2

Programm	512/2	Speedup	512/4	Speedup
Base	367 s	1	190 s	1
Threads	56.8 s	6.46	28.5 s	6.67
OpenCL	n/a	n/a	n/a	n/a

Programm	512/8	Speedup	256/1	Speedup
Base	92 s	1	332 s	1
Threads	14.3 s	6.43	53,0 s	6,26
OpenCL	2.29 s	40.17	5,98	55,52

Zusammenfassung

- ▶ Multi-Threading bietet stabilen Speed-Up, abhängig von der Anzahl der CPU-Kerne
- ▶ OpenCL zeigt großes Potenzial für Beschleunigung, limitiert durch VRAM und Implementierungsdetails
- ▶ Weitere Optimierungsmöglichkeiten bei der OpenCL-Implementierung vorhanden