

Das PARIS-Programm



Gliederung

- Namensgebung
- Problem und Forschungsstand
- Umsetzung
- Leistung
- Weiterführende Arbeiten

Namensgebung

PARIS

Portable and
Accelerated 3D
Reconstruction tool for radiation based
Imaging
Systems

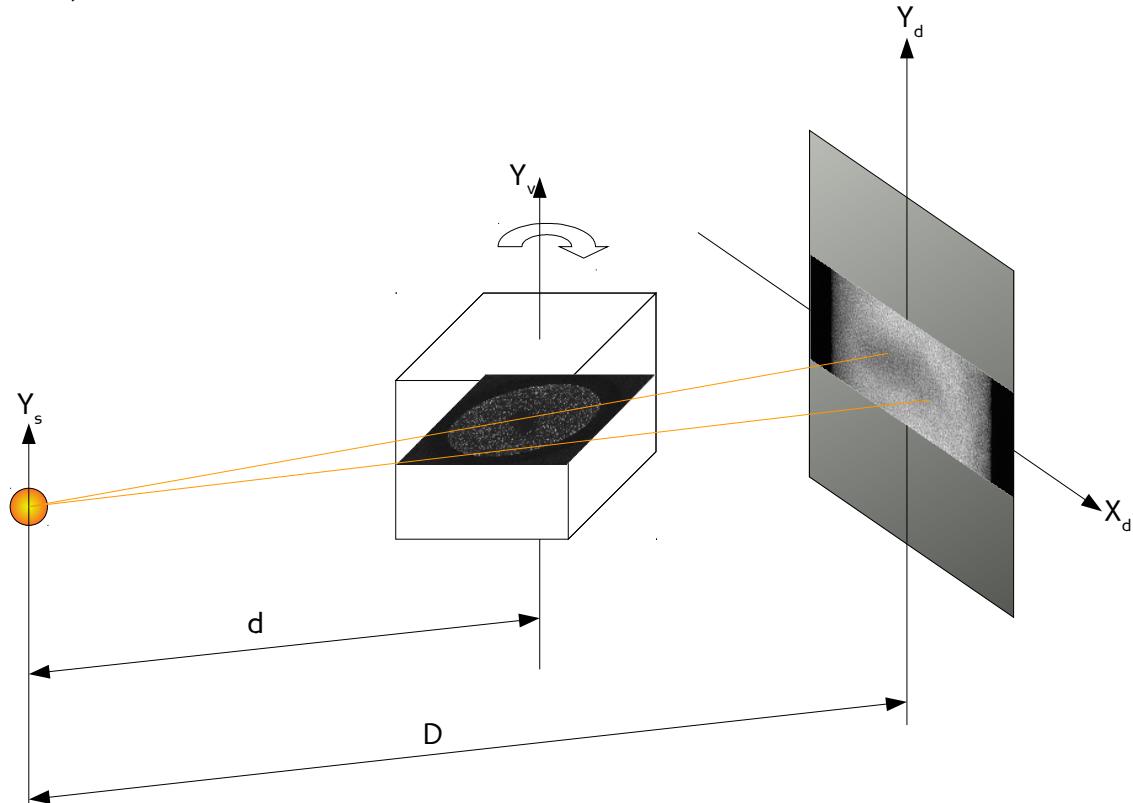


Quelle: http://voy.trekcore.com/gallery/albums/publicity/character/paris/paris_s7.jpg

Problem und Forschungsstand

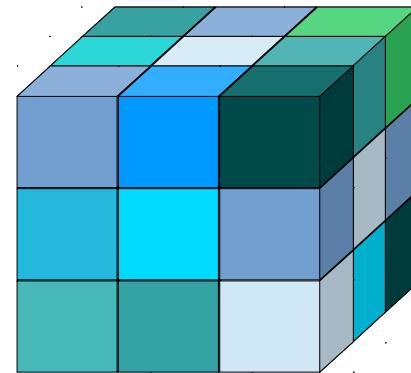
Der Feldkamp-Davis-Kress-Algorithmus (FDK)

L. A. Feldkamp, L. C. Davis & J. W. Kress: *Practical cone-beam algorithm*, Journal of the Optical Society of America A, Vol. 1, Issue 6, pp. 612 – 619, 1984



Problem und Forschungsstand

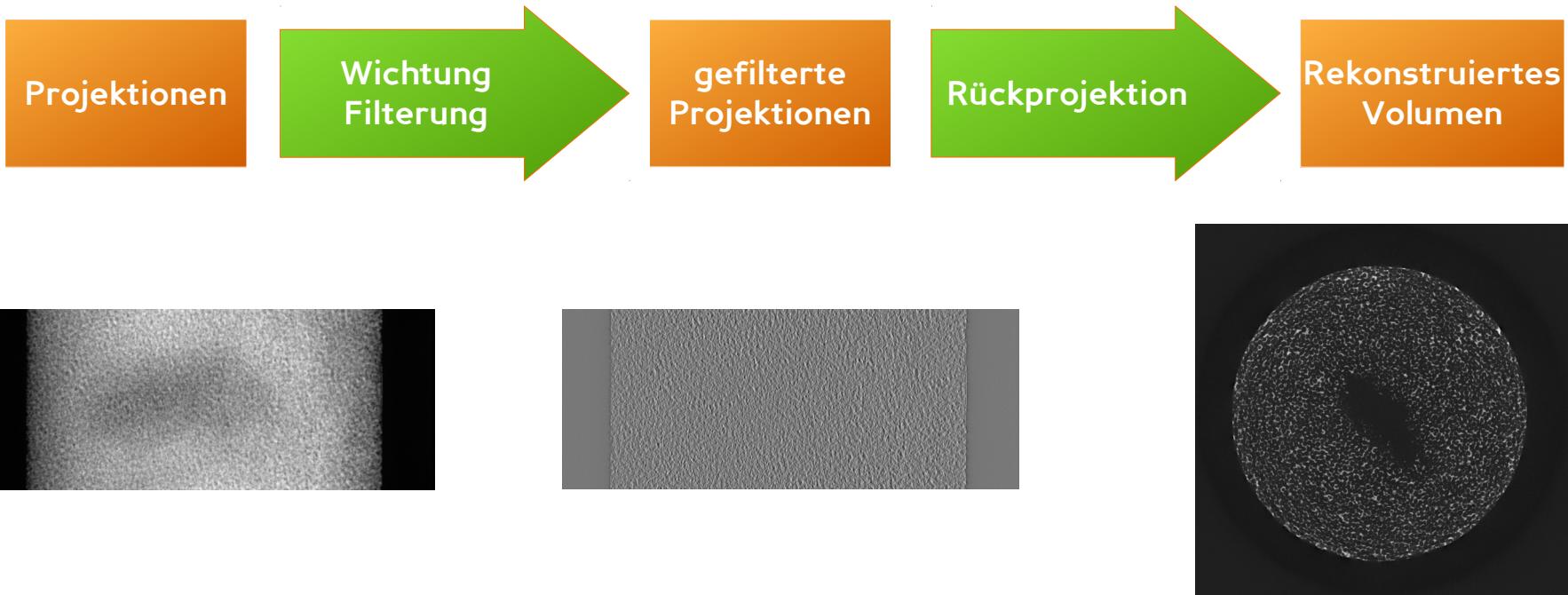
Parallelisierung



keine Abhangigkeiten zwischen Voxeln
→ stark parallelisierbar

Problem und Forschungsstand

Datenfluss



Problem und Forschungsstand

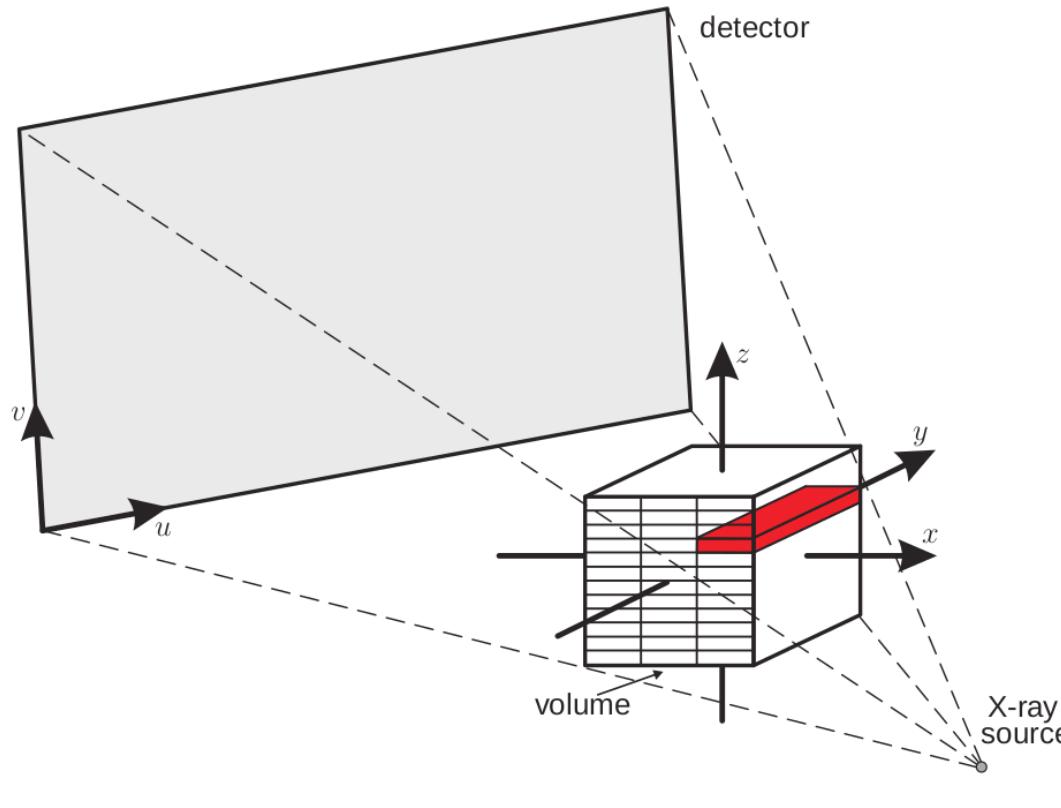
Bisherige Ansätze

- Xu et al.[1], 2004: vermutlich OpenGL
- Li et al.[2], 2005: FPGA
- Knaup et al.[3], 2006: PowerPC
- Scherl et al.[4], 2007: CUDA
- Domonkos et al.[5], 2009: OpenCL
- Zhao et al.[6], 2009: OpenGL und Symmetrien

Problem und Forschungsstand

Scherl et al.[4], 2007: CUDA

- einzelne Schritte in Pipeline
- Filterung mit cuFFT
- Anwendung von 2D-Kerneln auf 3D-Problem durch Iteration
- bilineare Interpolation durch Hardware



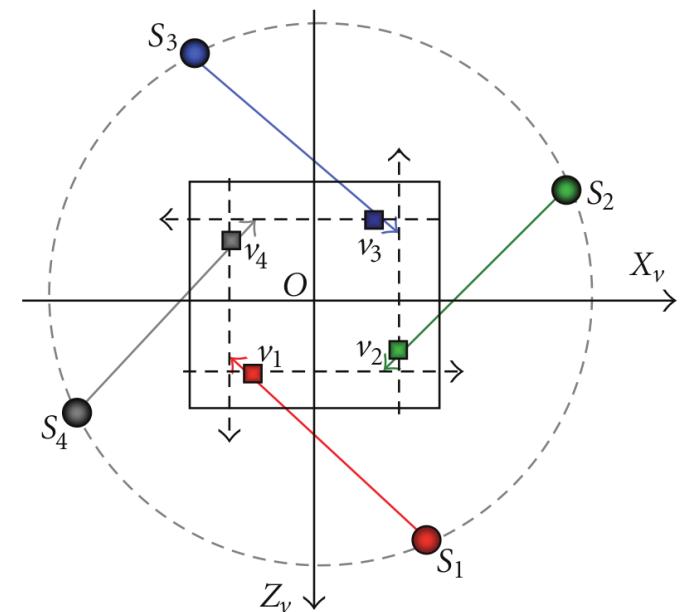
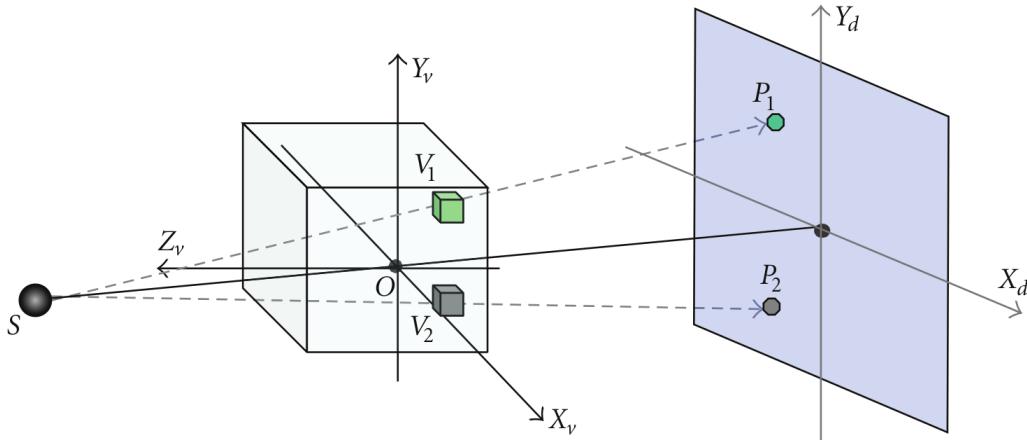
Problem und Forschungsstand

Zhao et al.[4], 2009: OpenGL und Symmetrien

- expliziter Verzicht auf CUDA:

[...] an implementation of the cone beam back-projection using the graphics pipeline is 3 times faster than the one made with CUDA interface [...]

- Nutzung von OpenGL-Shadern
- Ausnutzung von Symmetrien



Umsetzung

Aufgabe

- bestehendes Programm durch performantere Neuentwicklung ersetzen
- lauffähig auf unterschiedlich leistungsfähigen Systemen (Laptop / Workstation)
- Nutzung einer oder mehrerer GPUs
- einfache Wartbarkeit / Erweiterbarkeit
- Portabilität zu anderen Plattformen (OpenCL / OpenMP)

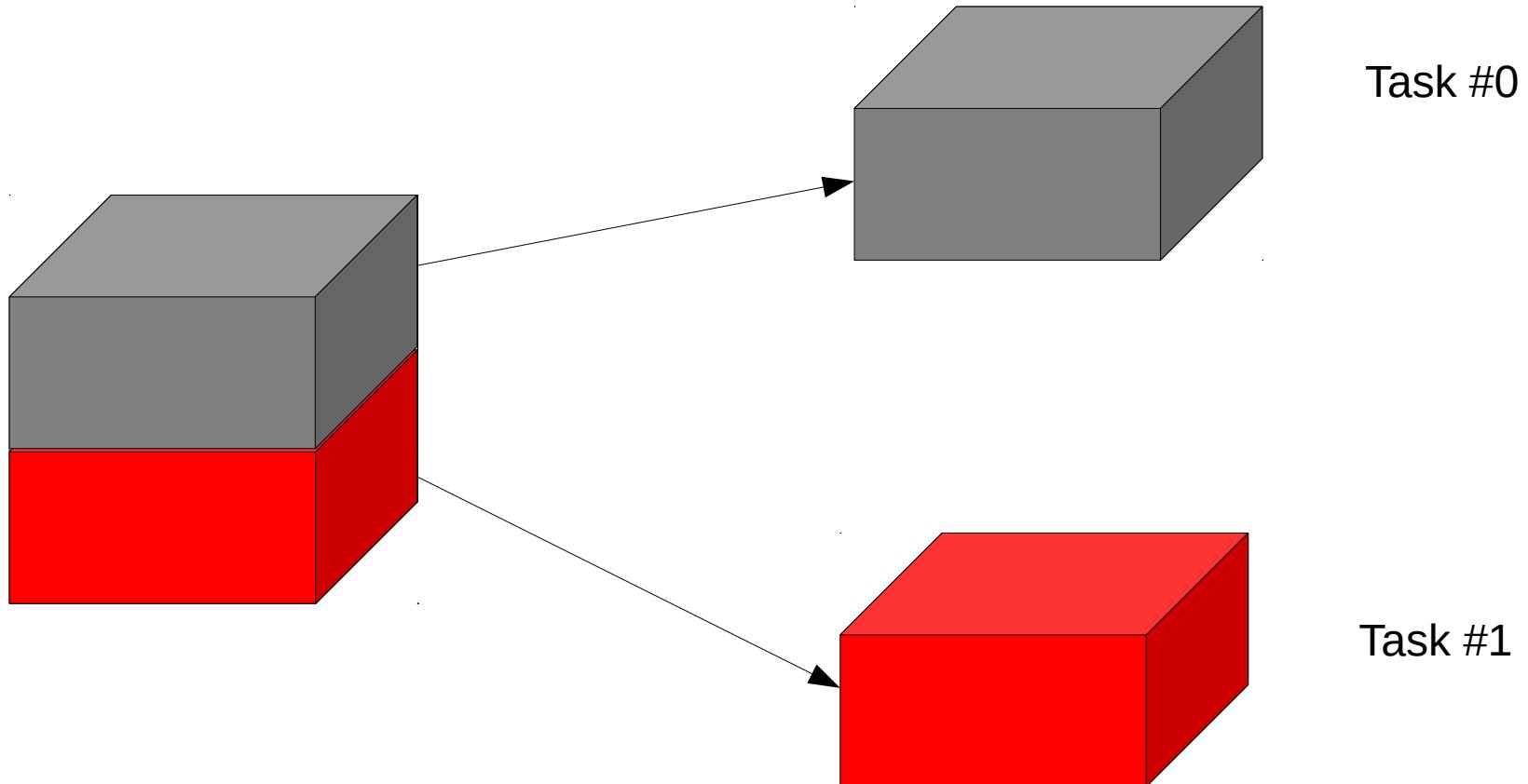
Umsetzung

Probleme

- Datenmenge oft zu groß für Verarbeitung „am Stück“
 - 1440 Projektionen (1024x1024; 32bit float)
 - Volumen (1070x1070x1033; 32bit float)
→ ca. 10 GiB Datenmenge
- unvollständige Eingangsdaten
- dynamische Nutzung unterschiedlicher GPUs

Umsetzung

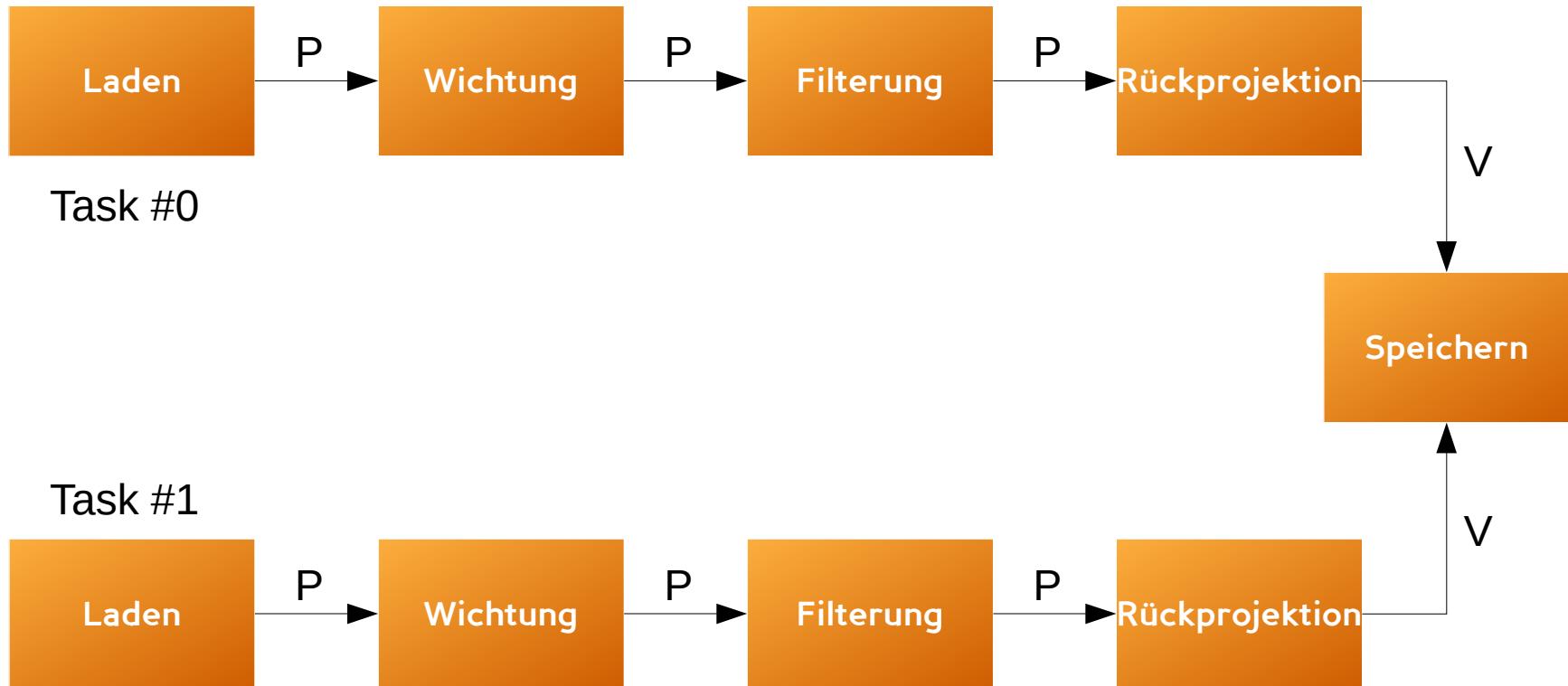
Teilvolumen



Umsetzung

Pipelining

Nutzung von GLADOS-Pipelines (<https://github.com/HZDR-FWDF/GLADOS>)



Umsetzung

Rückprojektion

- rechenintensiv
→ höchstes Optimierungspotential
- lineare Interpolation durch Hardware

```
auto proj_val = tex2D<float>(proj, h, v);
```

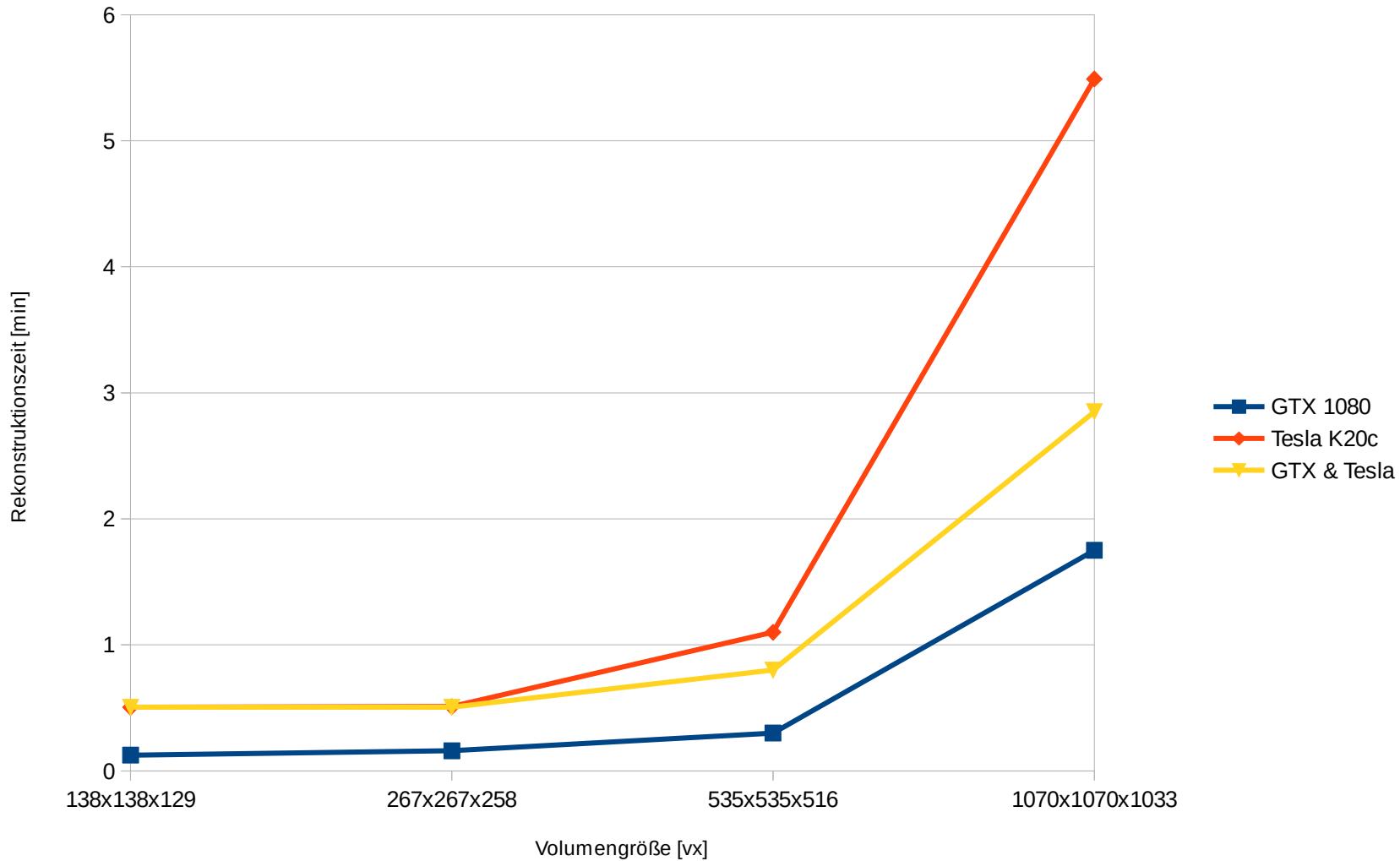
- Optimierung des globalen Speicherzugriffs

```
auto old_val = vol[coord];  
/* viele viele Berechnungen... */  
vol[coord] = old_val + 0.5f * proj_val * u * u;
```

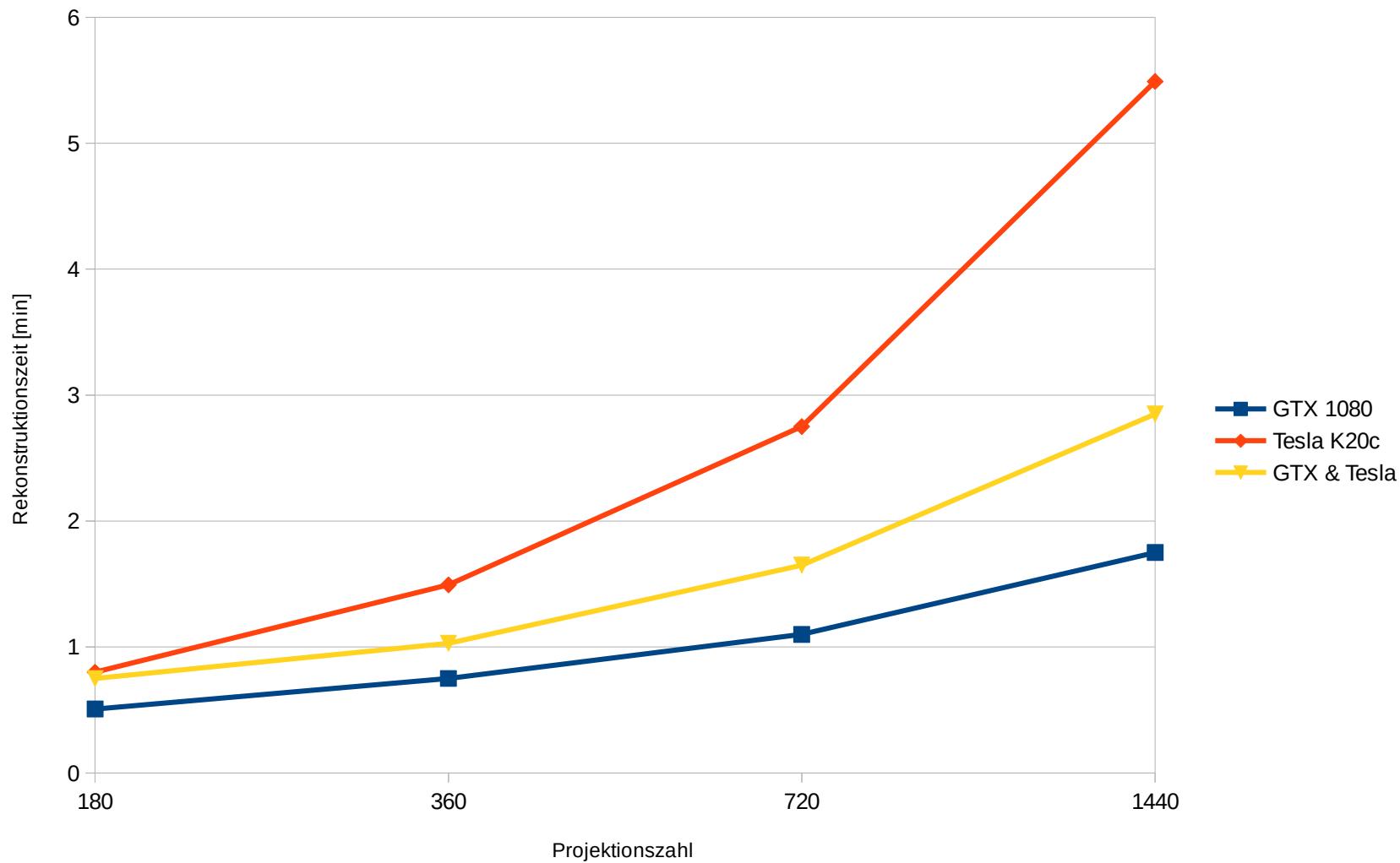
Leistungsmessung

	GTX 1080	Tesla K20c
Taktfrequenz	1607 MHz	706MHz
GPU-RAM	8 GiB	5GiB
CUDA-Kerne	2560	2496

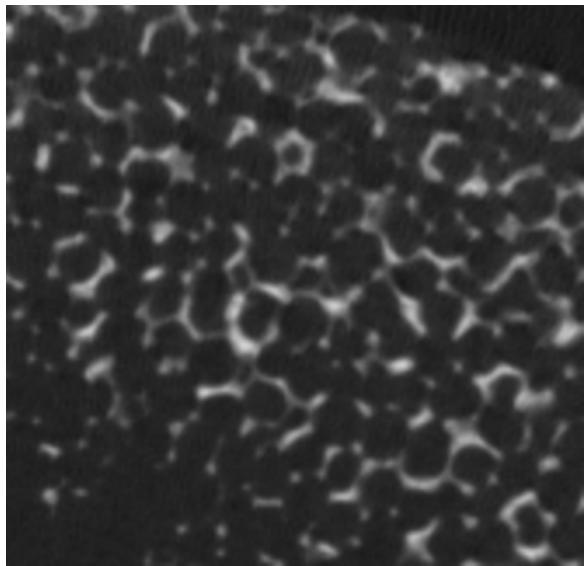
Leistungsmessung - Volumengröße



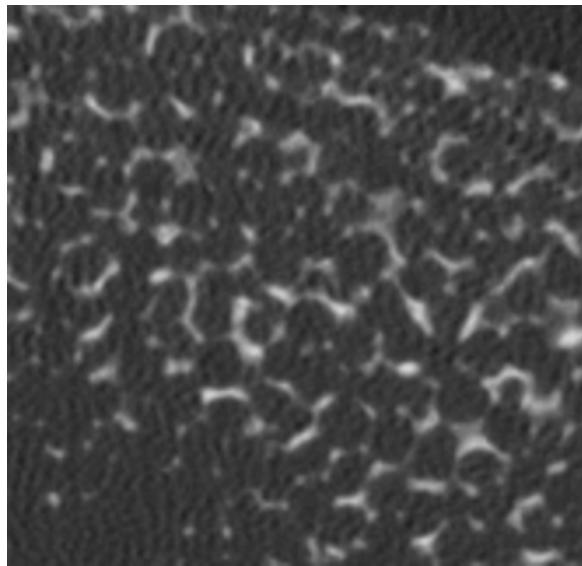
Leistungsmessung - Qualität



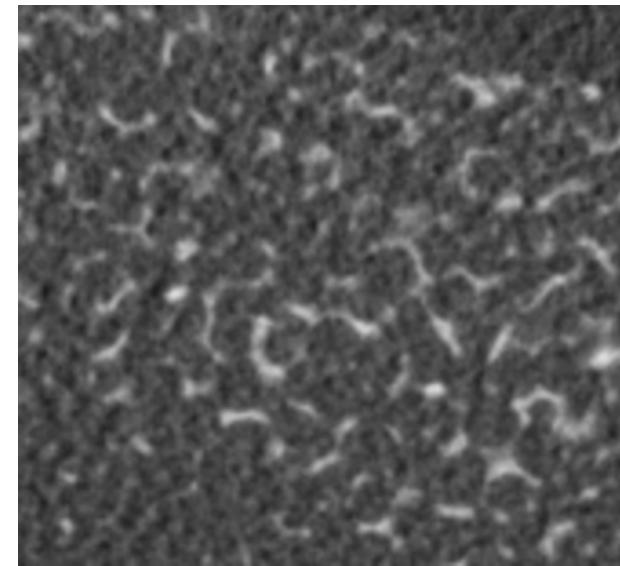
Qualitätsvergleich



1440 Projektionen



720 Projektionen



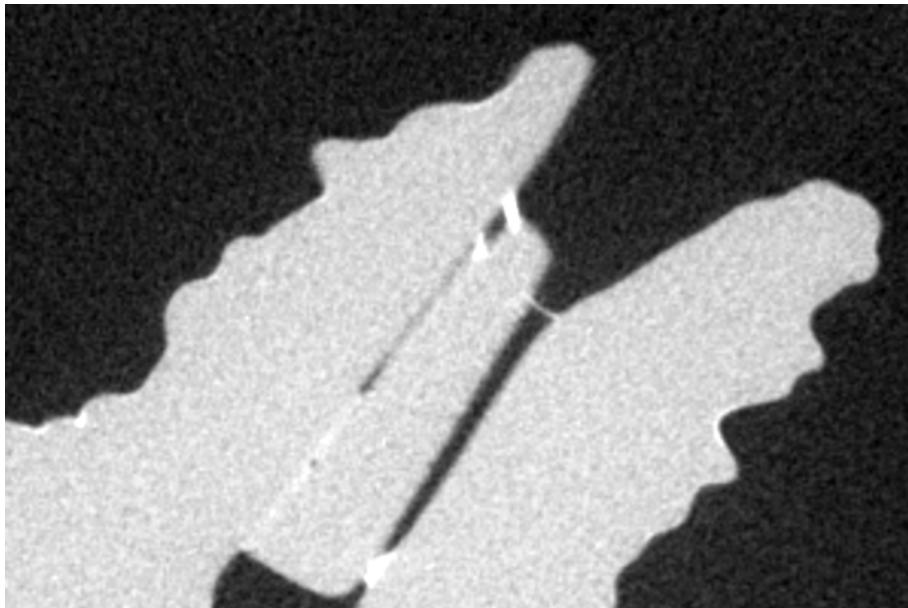
360 Projektionen

Vergrößerung 200%

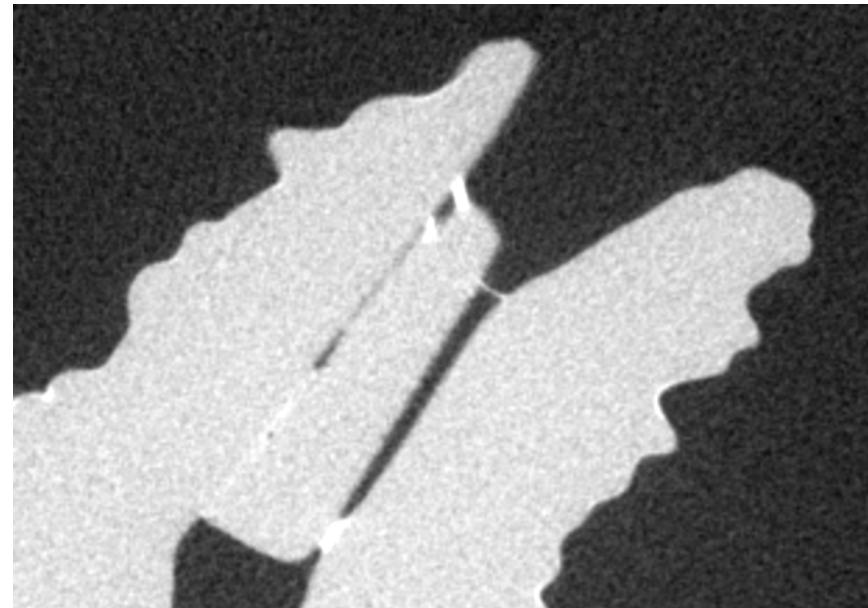
Leistungsmessung

	altes Programm	PARIS (GTX 1080)
Laufzeit (voller Datensatz)	mehrere Tage	wenige Minuten
Laufzeit (reduzierter Datensatz: 1024x1024x30)	3h11min	11s

Qualitätsvergleich



alt



neu

Vergrößerung 200%

Weiterführende Arbeiten

Bessere Ressourcenausnutzung

- Blockgröße zur Zeit statisch ($16 \times 16 \times 2$)
- Performance-Messung verschiedener Blockgrößen

Vorverarbeitungsschritte

- Korrektur von defekten Pixeln (Dunkelbild / Binärmaske)
- Berechnung der Schwächungswerte
- Winkelkorrektur

→ Gegenstand des CUDA-Praktikums an der TU Dresden

Weiterführende Arbeiten

Echtzeitrekonstruktion

- Rekonstruktion schneller als Aufnahme
→ bislang reine Theorie

Portierungen

- derzeit nur CUDA
- Portierung auf Xeon Phi Gegenstand meines Forschungsbeleges
- Portierung auf OpenCL erwünscht, aber nicht in Arbeit

Schluss.

Literatur

- [1] Fang Xu, Klaus Müller: *Ultra-Fast 3D Filtered Backprojection on Commodity Graphics Hardware*, IEEE International Symposium on Biomedical Imaging, April 2004
- [2] Jianchun Li, Christos Papachristou, Raj Shekhar: *An FPGA-Based Computing Platform for Real-Time 3D Medical Imaging and its Application to Cone-Beam CT Reconstruction*, Journal of Imaging Science and Technology, Vol. 49 Nr. 3, S. 237 – 245(9), Juli 2005
- [3] Michael Knaup, Sven Steckmann, Olivier Bockenbach, Marc Kachelrieß: *Tomographic image reconstruction using the cell broadband engine (CBE) general purpose hardware*, Proc. SPIE 6498, Computational Imaging V, 64980P, Februar 2007
- [4] Holger Scherl, Benjamin Keck, Markus Kowarschik, Joachim Hornegger: *Fast GPU-Based CT Reconstruction using the Common Unified Device Architecture (CUDA)*, 2007 IEEE Nuclear Science Symposium Conference Record, S. 4464 – 4466, Oktober 2007
- [5] Balázs Domonkos, Gábor Jakab: *A programming model for GPU-based parallel computing with scalability and abstraction*, SCCG '09 Proceedings of the 25th Spring Conference on Computer Graphics, S. 103 – 111, April 2009

Literatur

- [6] Xing Zhao, Jing-Jing Hu, Peng Zhang: *GPU-based 3D cone-beam CT image reconstruction for large data volume*, Journal of Biomedical Imaging, Vol. 2009, Art. 8, 2009