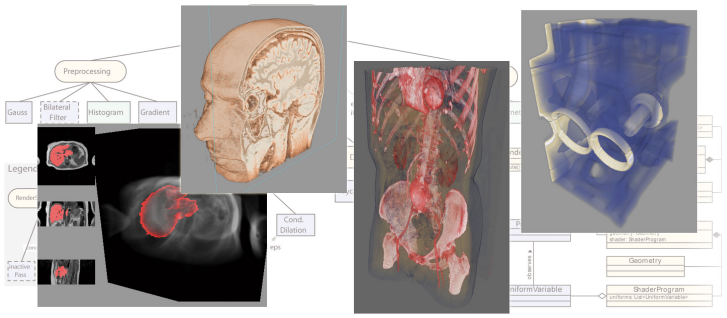
UNIVERSITÄT KOBLENZ · LANDAU

COMPUTERVISUALISTIK

Projekt: Cascada



cascada

Cascada...

...bedeutet in verschiedenen Sprachen "Wasserfall" — und genau das beschreibt die Arbeitsweise des Systems: Daten fließen als Strom durch verschiedene Verarbeitungsstufen und können dabei bei Bedarf direkt visualisiert werden. Ob nun ein kleiner einstufiger Wasserfall zur Glättung oder Kantendetektion oder eine komplexe, mehrstufige Kaskade von Einzelschritten zur Segmentierung und Analyse von großen Volumendaten — Cascada bietet als plattformübergreifendes und objektorientiertes GPU-Framework ein weites Einsatzgebiet.

Warum Grafikhardware?

Die Weiterentwicklung von Grafikkarten hat in den vergangenen Jahren derart zugenommen, dass Grafikprozessoren (GPU) die Performance mehrerer hundert CPUs erreichen können. Um diese Leistung auch für mehr als nur Visualisierung nutzbar zu machen, ist in mehreren Projekten und Arbeiten das System "Cascada" entstanden. Hiermit können ganze Abläufe zur Verarbeitung von Volumendaten abgebildet und auf der Grafikhardware ausgeführt werden. Neben dem Performancegewinn, der bereits in mehreren Arbeiten veröffentlicht wurde, können die Prozesse direkt visualisiert werden: die Daten liegen bereits im Grafikspeicher, aufwendiger Datentransfer entfällt. Das typische Anwendungsgebiet von Cascada ist die Verarbeitung und Visualisierung medizinischer Volumendaten wie CT- oder MR-Aufnahmen. Darüber hinaus können aber prinzipiell beliebige Bilddaten verarbeitet werden, z.B. auch Videosequenzen.

Cascada vs. CUDA

Seit der Einführung von Grafikkarten der neuesten Generation gibt es spezielle APIs, die von den Computergrafik-spezifischen Details abstrahieren und die GPUs als eine Art Coprozessor in einer C-ähnlichen Sprache steuern. Dieser Trend steht aber nur auf den ersten Blick in direkter Konkurrenz zu Cascada. Während Systeme wie CUDA oder CTM klassisch, aber sehr nah an der jeweiligen Hardware zu implementieren sind, repräsentieren die hierarchisch aufgebauten Komponenten in Cascada komplette Algorithmen und Arbeitsschritte. Außerdem können die bisherigen GLSL-Shaderprogramme um eine CUDA-basierte Implementation ergänzt werden. Dies gehört neben flexibleren Datenformaten und Automatisierungen zu den aktuellen Weiterentwicklungen.

Team

Matthias Raspe, Guido Lorenz, Nils Hering und Studierende

Kontakt

Dipl.-Inform. Matthias Raspe
Arbeitsgruppe Computergrafik
Institut für Computervisualistik
Universitätsstr. 1
56070 Koblenz
Tel.: 0261-287-2794
Fax.: 0261-287-2735
mraspe@uni-koblenz.de