

Architektur und Programmierung von Grafik- und Koprozessoren

Motivation und Übersicht

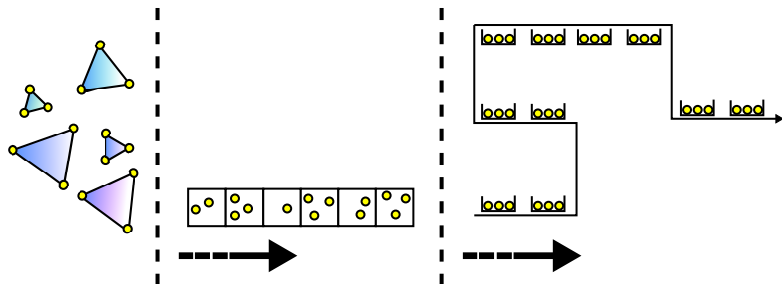
Stefan Zellmann

Lehrstuhl für Informatik, Universität zu Köln

SS2019

Motivation: High-Level Grafik Pipeline moderner GPUs

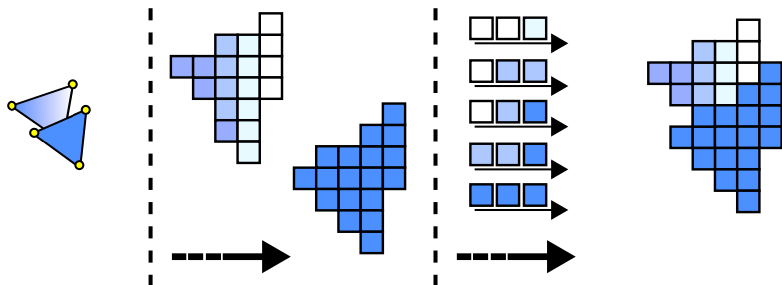
Vertex Phase



Hochparallele Verarbeitung von Millionen von Vertices.

Motivation: High-Level Grafik Pipeline moderner GPUs

Rasterisierung und Fragment Phase



Hochparallele Verarbeitung von Fragmenten und Pixeln.

Motivation: General Purpose GPUs

- ▶ Anfangs separate Cores für Vertex und Fragment Phase.
Problem: Lastimbilanzen.
- ▶ Daher haben die Hersteller schnell sog. “Unified Shader Cores” entwickelt, die beide Aufgaben durchführen können.
- ▶ Problem: Generalität - Vertex und Fragment Operationen haben unterschiedliche Charakteristiken \Rightarrow Unified Shader Cores daher genereller als “Special Purpose Cores”.
- ▶ Resultat: Chip mit hunderten, massiv parallelen, generellen, aber sehr leichtgewichtigen Cores. Andere Anwendungsdomänen haben das schnell erkannt und für ihre Zwecke ausgenutzt \Rightarrow “General Purpose GPU Programming” (GPGPU).

Motivation: Koprozessoren

- ▶ Orthogonale Entwicklungen: Intels “Larrabee” als Konkurrenz zu Nvidia und ATI GPUs.
- ▶ Many-Core im High-Performance Computing, Intel Xeon PHI Koprozessor Karten.
- ▶ Intel: Koprozessoren auf dem Rückzug, aber Charakteristiken halten in die Hauptprozessorlinien Einzug.
- ▶ FPGAs als PCIe Einschubkarten.

Übersicht

1. Performanz von Computerprogrammen
2. Rendering Algorithmen
3. Die Grafik Pipeline
4. Programmieren mit dem Vulkan API
5. General Purpose Programmierung für Grafikprozessoren
6. Anwendungen

Eventuell:

7. Multi-GPU Systeme
8. Intel Koprozessoren
9. Field Programmable Gate Arrays
10. Ray Tracing Hardware

Termine

1. Do 04.04. & Fr 05.04.
 2. Do 11.04. & Fr 12.04.
 3. nur Do 18.04. (Ostern)
 4. Woche 25.-26.04.:
Keine Vorlesung, nur Übung!
 5. Do 02.05. & Fr 03.05.
 6. Do 09.05. & Fr 10.05.
 7. Do 16.05. & Fr 17.05.
 8. Do 23.05. & Fr 24.05.
 9. nur Fr. 31.05.
(Christi Himmelfahrt)
 10. Do 06.06. & Fr 07.06.
 11. Pfingstferien
 12. nur Fr. 21.06.
(Fronleichnam)
 13. Do 27.06. & Fr 28.06.
 14. Do 04.07. & Fr 05.07.
 15. Do 11.07. & Fr 12.07.
 16. Klausur: Mo 24.07.
 17. Nachklausur: Fr 25.09.
- Probeklausur:
- ▶ Entweder in Woche 04.-05.07. oder 11.07.-12.07.

Organisatorisches

- ▶ Unterlagen online unter <https://vis.uni-koeln.de/apgk2019.html> sowie in Ilias.
- ▶ **Ilias Passwort: apgk2019**
- ▶ Sprechstunde: nach Vereinbarung
- ▶ Bitte nutzen Sie die Mailingliste: apgk-vorlesung@uni-koeln.de
 - ▶ Anmeldung erforderlich.
 - ▶ Technische Fragen können an Mailingliste gerichtet werden.

Organisatorisches (Übungen)

- ▶ Erste Übung (04.04., heute) entfällt.
- ▶ Zweite Übung (11.04.) Sonderübung zu C++
- ▶ Besprechung des ersten Übungsblatts am 18.04.
- ▶ Keine Zulassungsbeschränkung auf Grund von Übungspunkten
- ▶ Übungsblätter ca. eine Woche vor Besprechung online
 - ▶ Ilias (Achtung: separate Kurse für Übung und Vorlesung, Passwort für Kurs zu Übungen ebenfalls apgk2019)
 - ▶ <https://vis.uni-koeln.de/apgk2019.html>