Inhalt

* Einführung in OpenCL
  + Anforderungen (Hardware, Treiber)
  + API (Platforms, Devices, CommandQueues, Buffer, Programs, Kernels)
  + Programmausführung (enqueue Kernels, async)
  + Memory model
    - Global, local, constant
    - Buffers: host, device, shared (heterogen)
  + Hardware Architekturen
    - Grafikkarten-/Prozessorarchitektur
    - Abstraktion durch OpenCL
    - Begriffsbestimmungen in OpenCL und des Hardwareherstellers (AMD, NVIDIA)
* State of the Art
* Implementierung
  + Matrix multiplication
    - Naiv
  + Parallel prefix sum (scan)
    - Apple
    - GPU Gems
  + Sorting
    - AMD (Radix sort)
    - Bealto (Bitonic sort)
    - Bekannte CPU Algorithmen (Quicksort, Radixsort)
  + Weitere Ideen:
    - Noise
    - Heuristics
    - Simulations (Fluids, Particles)
    - Image filter
    - Marching cubes
    - Heterogen?
* Optimierung
  + Using local memory
  + Bank conflicts
  + Coalesced memory access
  + Branching
* Benchmarks
  + Vergleiche zwischen den Implementierungen
    - Laufzeit
    - Speicherkomplexität
* Vergleich mit CUDA und DirectCompute
* Vor- und Nachteile
  + Von Grafikkarten
  + Von Prozessoren
  + Von OpenCL
* Anwendungsgebiete
  + Von OpenCL
  + Von GPGPU Beschleunigung
  + WebCL