The GAIA Project:
Evaluation of GPU-Based Programming Environments for Knowledge Discovery

Jeremy Meredith

David Bremer, Lawrence Flath, John Johnson, Holger Jones, Sheila Vaidya, Randall Frank*

Lawrence Livermore National Laboratory

UCRL-PRES-206819

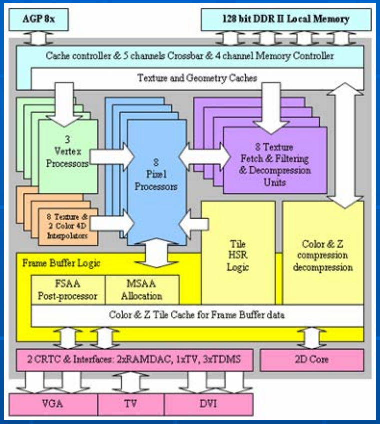
This work was performed under the auspices of the U.S. Department of Energy by the University of California, Lawrence Livermore National Laboratory under contract No. W-7405-Eng-48.

Motivation

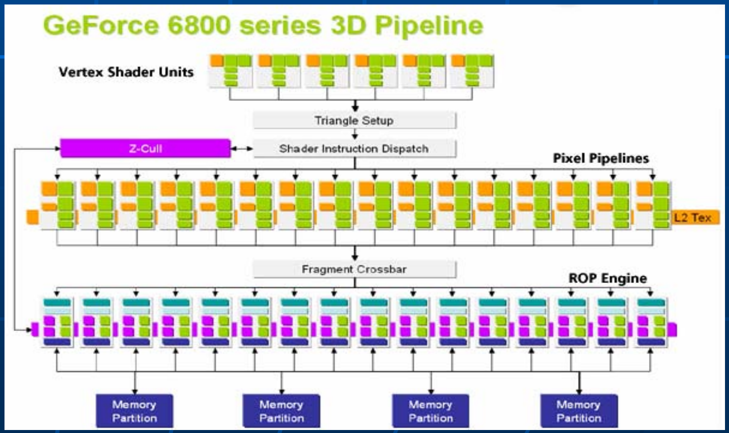
- Trends in the graphics marketplace
  - Inherent parallelism of graphics tasks
  - Performance increasing faster than for CPUs
  - Move to programmable hardware
  - Effects of mass markets
- Not expected to end anytime soon...
  - Today: 40GF, 2GB/s I/O, 30GB/s memory
  - 2006: 100GF, 8GB/s I/O, 60GB/s memory
  - 2007: 1TF...

The NV40 and the Sony Playstation 3

- Are graphics trends a glimpse of the future?
- The nVidia NV40 Architecture
  - 256MB+ RAM
  - 128 32bit IEEE FP units @ 400Mhz
  - 220M transistors, 110W of power
- The PlayStation3 (patent application)
  - Core component is a cell
    - 1 “PowerPC” CPU + 8 APUs (“vectorial” processors)
    - 4GHz, 128K RAM, 256GFLOP/cell
  - Multiple cells (Phone, PDA, PS3, …)
    - Four cell architecture (1TFLOP)
    - Central 64MB memory
- Keys
  - Streaming data models
  - Cache-driven/cache-oblivious computing

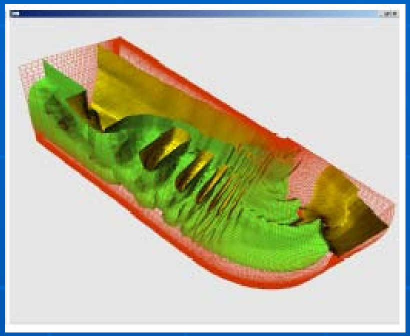


nVidia NV30

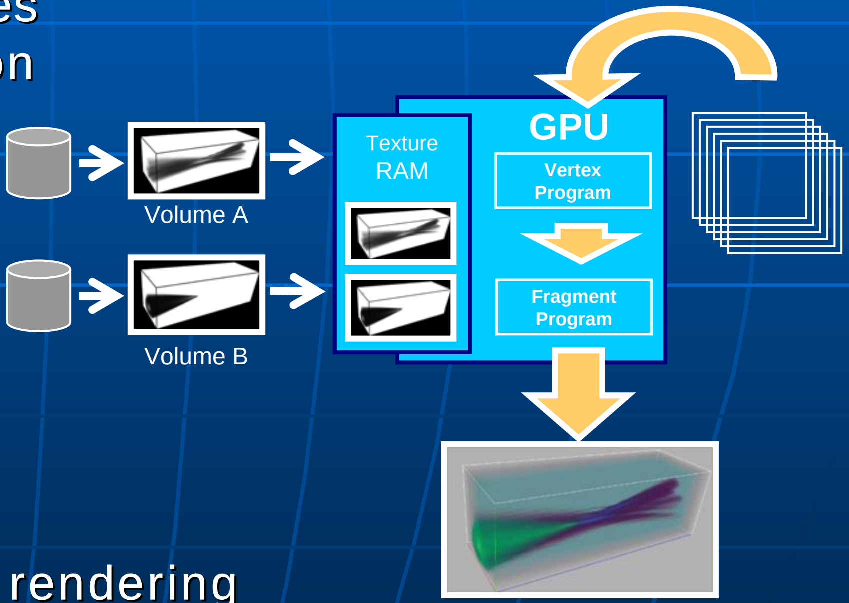


nVidia NV40

Data representations for GPUs



- Programmable FP SIMD engines, 40-100GF today, 1TF by ’06
- Where can they be exploited?
  - Many advantages for the data pipeline
  - Data/algorithmic design challenges
  - Possible applicability for simulation
  - Many current research projects on scientific computing, databases, audio processing
- Current projects
  - Programmable rendering pipeline
    - Multi-variate, interactive
    - Increased graphics precision
  - Image composition pipeline
  - Implementation of physics based rendering
    - Simulated radiography, diffraction computation
  - Large image geo-registration
    - 100x performance improvement over CPU



Specific Project Goals

- Investigate use of COTS technologies for computation
  - “Non-traditional” applications
    - Image and speech
    - String, statistical, graph…
  - Mechanisms necessary for exploitation
    - Data infrastructure (e.g. cache coherent streaming…)
    - Software abstractions
  - Delineate some boundary conditions on their use
    - Evaluation vs CPU based solutions
    - Parameter-space investigation

Data Infrastructure

- Forms the basis of a comparative framework
  - Support both GPU and CPU algorithmic implementations
  - Targets multiple platforms
  - Provides data abstraction
    - “Tile-based” streaming
    - Cache coherency control
    - CPU to GPU to CPU glue layer
  - Utilizes higher-level languages for algorithms
    - Cg, Brook, GLSL, etc

Image Processing Applications

- Common attributes
  - Large, streaming imagery on a single gfx card
  - Parallel 1D and 2D applications
  - Multi-spectral (four, possibly temporal channels)
- Discrete convolution
  - Arbitrary kernels
- Correlation
  - Separate threshold, search, and detection phase included

String Processing Applications

- Representation and bandwidth characteristics
- String comparison
  - "Bulk" comparison operations individual outputs
- String sorting
  - Based on string comparison
  - Batched sort based on radix algorithms
- String searching
  - "Wildcard" pattern matching
  - Sort-based element search

Other Application Targets

- Image transforms
  - FFT, Wavelet
  - Many application domains
- Statistical functions on images
  - Moments, regression (general linear model)
  - Hypothesis/model driven image processing, texture characterization, etc
  - Hidden Markov Models
- Graph search
  - Structured (fully connected) or unstructured graphs, detect and return lowest cost path
  - Many application domains

System Targets

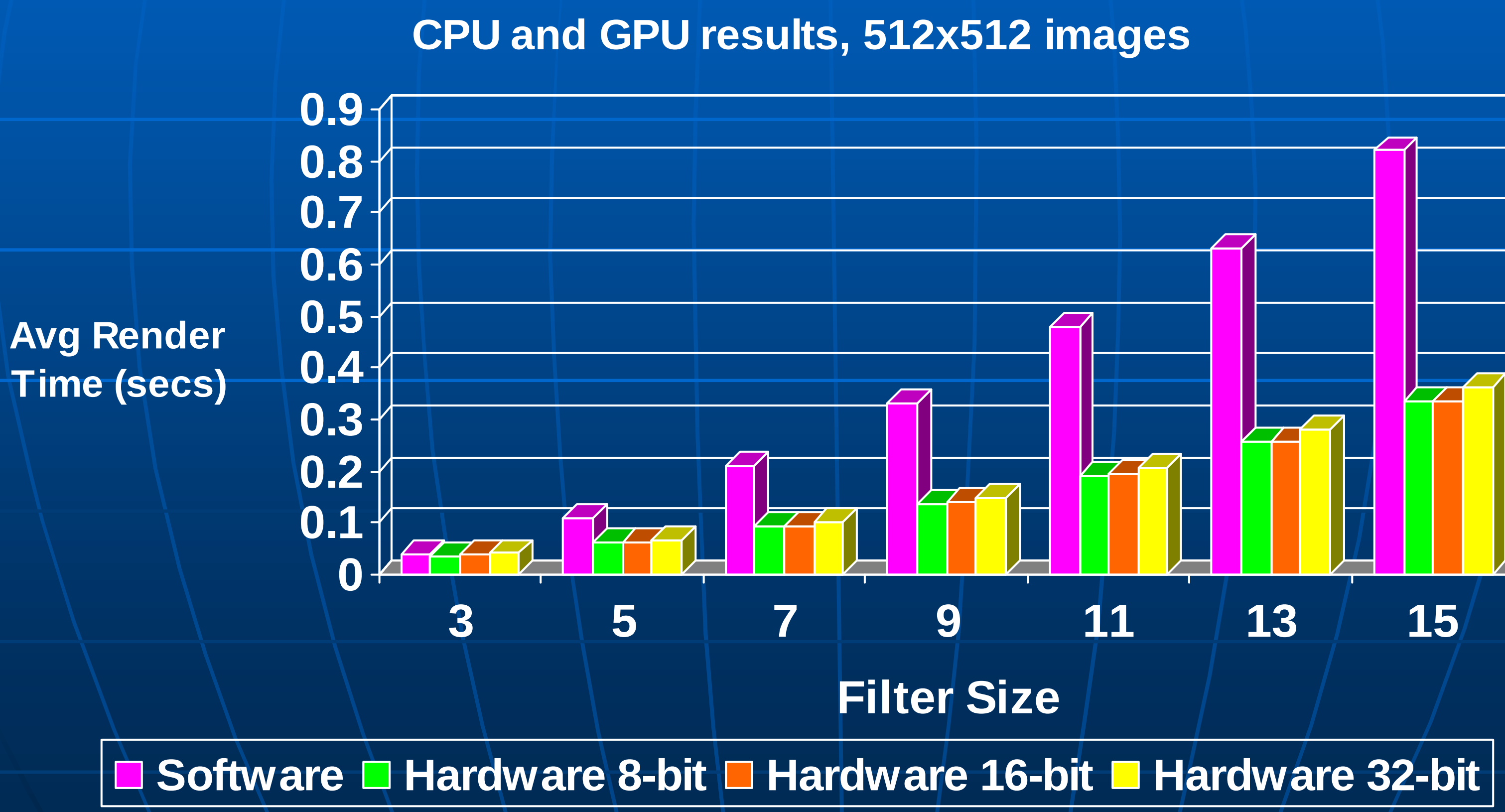
- Constrained system targets based on resource limits
- Hardware targets
  - nVidia: NV3x, NV4x, NV5x
    - Focus on NV4x due to new branching capabilities
    - Dual CPU IA32 platform
    - PCI-Express (PCIe) enhanced readback and async bandwidth
  - BG/L and Merrimac
- OS targets
  - Primarily Linux, some Windows due to driver issues
- Language targets
  - nVidia Cg, Brook

Convolution Timing Results

- All timings count download, render, and readback
- First render pass is excluded from the count
- Overhead to load shader can be substantial

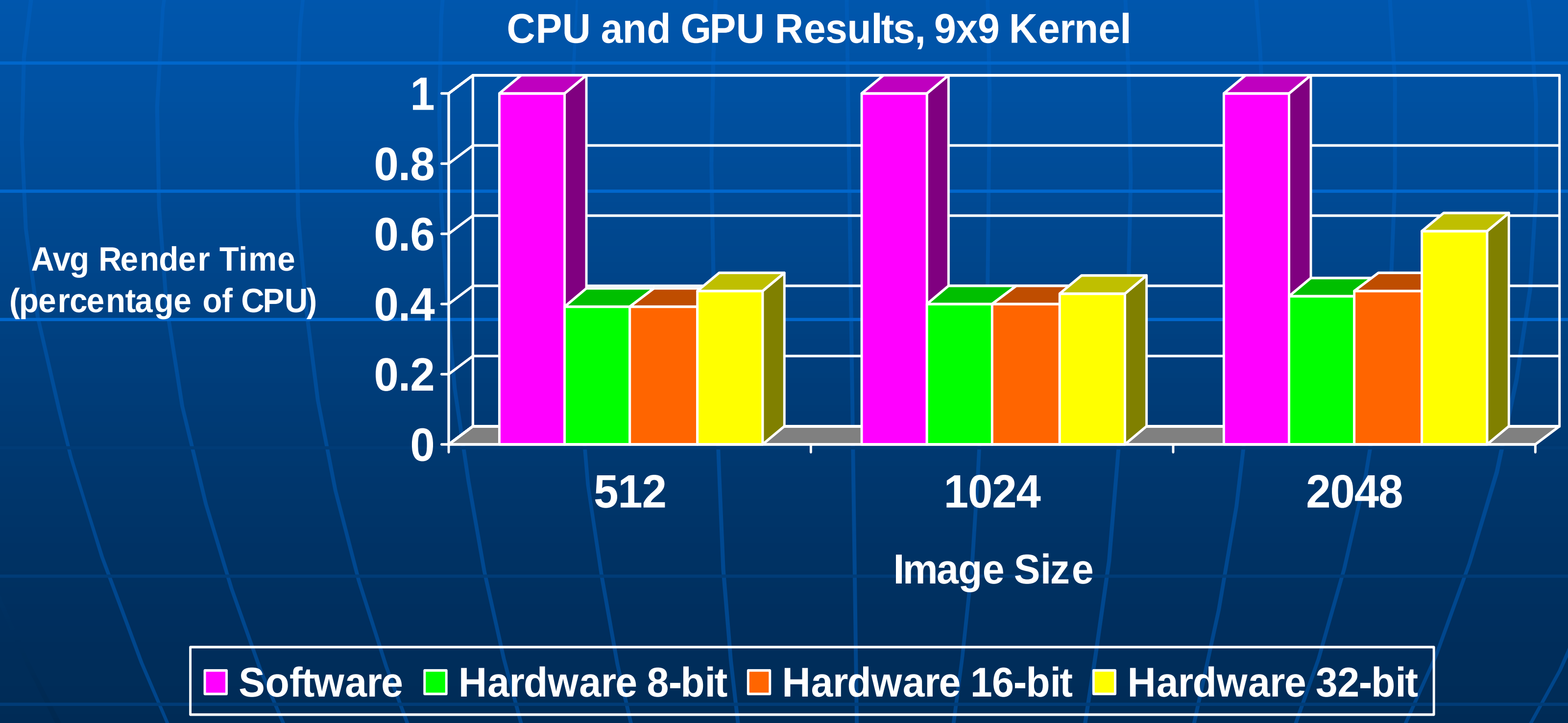
Convolution Timing Results

- Software vs. two-texture hardware implementation
- At all but the smallest kernel sizes, GPUs are much faster



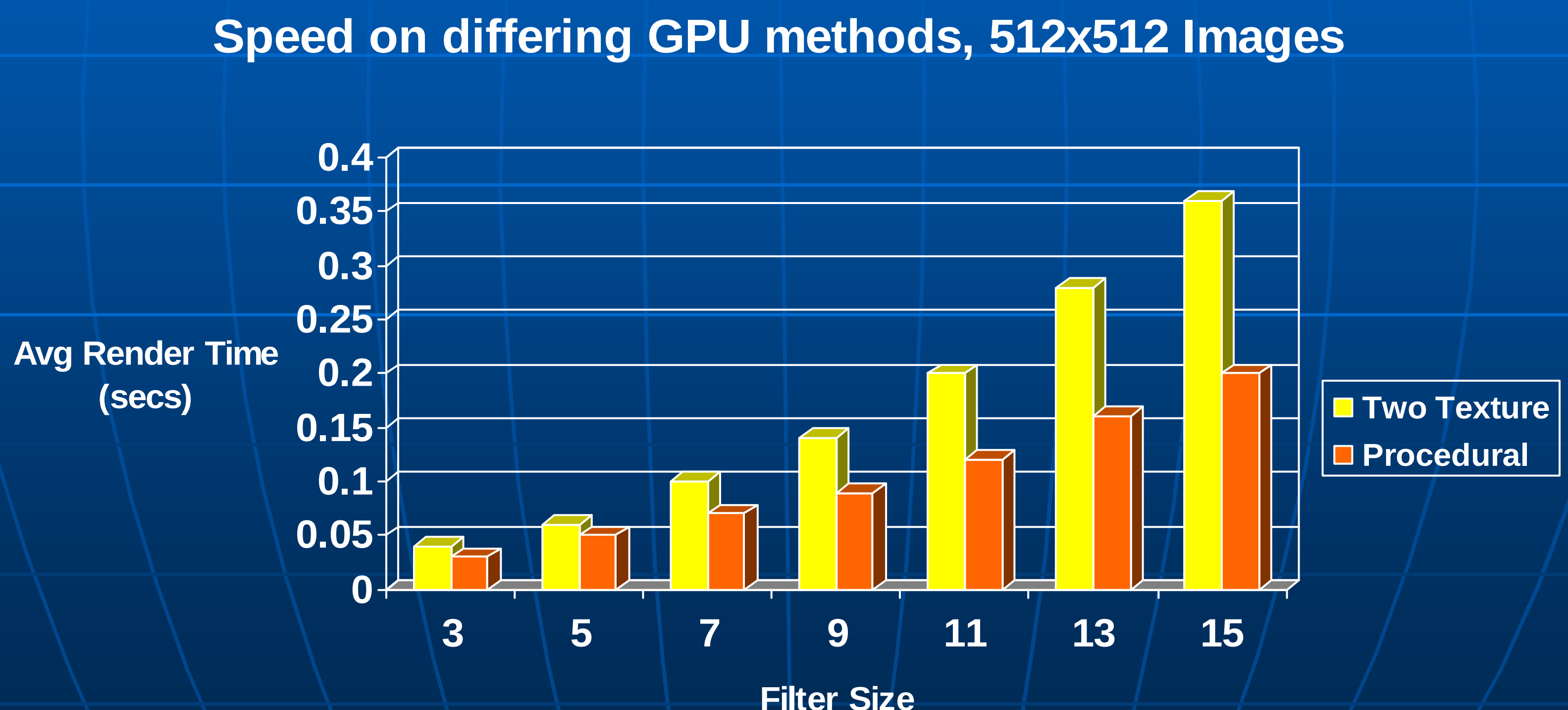
Convolution Timing Results

- Software vs. two-texture hardware implementation
- 32-bit textures use more memory bandwidth



Convolution Timing Results

- Two-texture vs. procedural hardware implementations
- Two-texture implementation requires more memory bandwidth



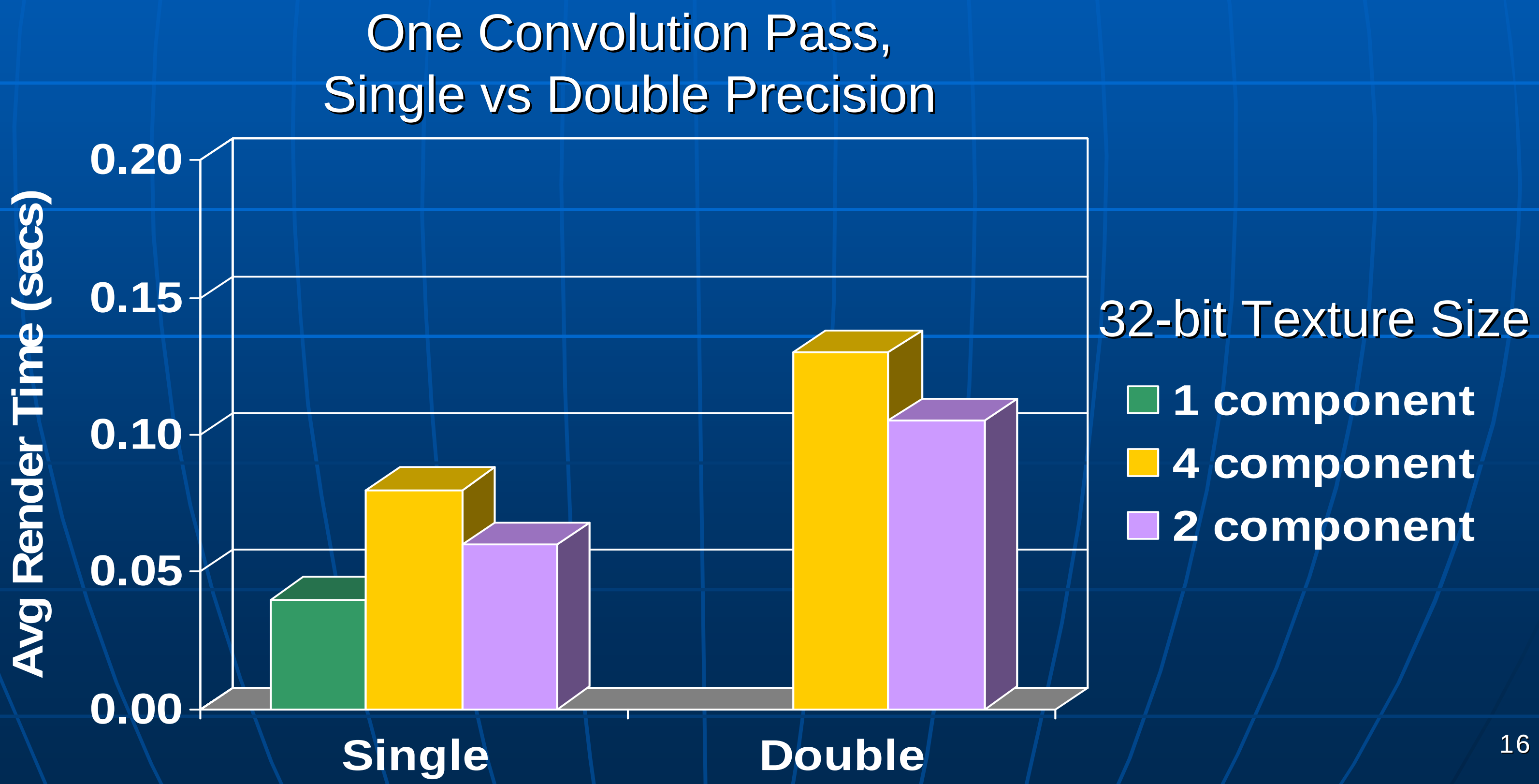
Double Precision

- Port of David Bailey's *single-double* Fortran library* to NVidia's Cg language
- Can emulate double precision
- Use two single-precision floats
- High order float is estimate to the *double*; Low order float is error of that estimate
- Resulting precision is almost *double*
- The exponent remains at *single* range

available at htpp://crd.lbl.gov/~dhbailey/mpdist

Double Precision Results

- Convolution with single and emulated-double arithmetic
- Double precision only 1.5x slower than single precision at the same texture depth



Future Plans

- Obtain results for a variety of algorithms including strings, HMMs, and FFTs
- Include performance and accuracy
- Extend to new architectures as available (e.g. Merrimac)
- Explore other high-level languages (e.g. brook implementations and other streaming languages)
- Launch a benchmarking web site: http://www.llnl.gov/gaia