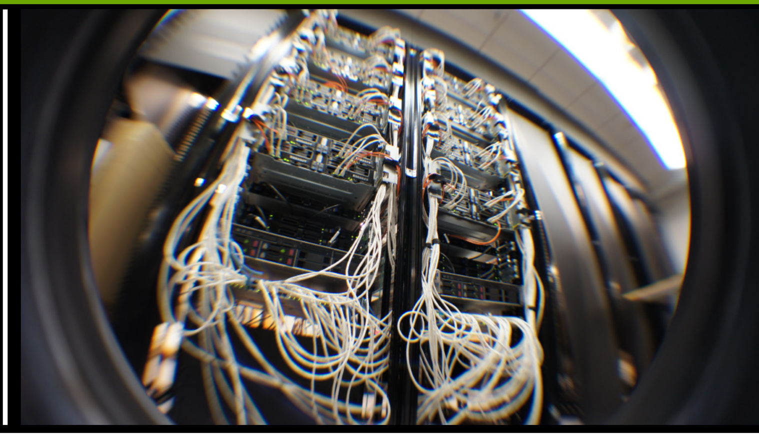
NVIDIA CUDA Center of Excellence
AT THE UNIVERSITY OF UTAH

www.sci.utah.edu

Tesla Cluster

32 Tesla S1070 GPU systems 4x GPUs per node
64 nodes (512 real cores, 1024 HT, 1.5TB RAM)
2x Intel Xeon CPU X5550 @ 2.67GHz, 4 cores
24GB RAM per node
DDR IB 16Gbit/s (64 links)
1x IB channel per node

Compute (CPUs Only)	4.9 TFLOPs
GPUs	9.8 TFLOPs
Total Performance	14.7 TFLOPS

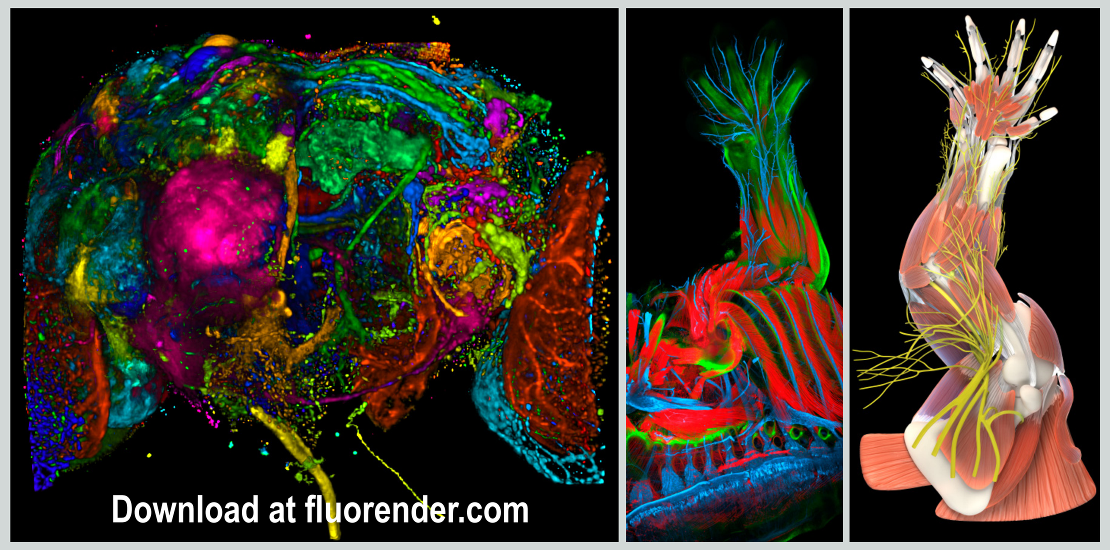


Kepler Cluster

64 internal Nvidia K20 GPUs
32 nodes (512 real cores, 1024 HT, 2TB RAM)
2x Intel Xeon E5-2660 @ 2.20 GHz, 8 cores
64GB RAM per node
FDR IB 56Gbit/s (128 links)
2x IB channels CPU / 4x IB links per node

Compute (CPUs Only)	9 TFLOPs
GPUs (Nvidia Keplar K20)	64 TFLOPs
Estimated Total Performance	73 TFLOPs

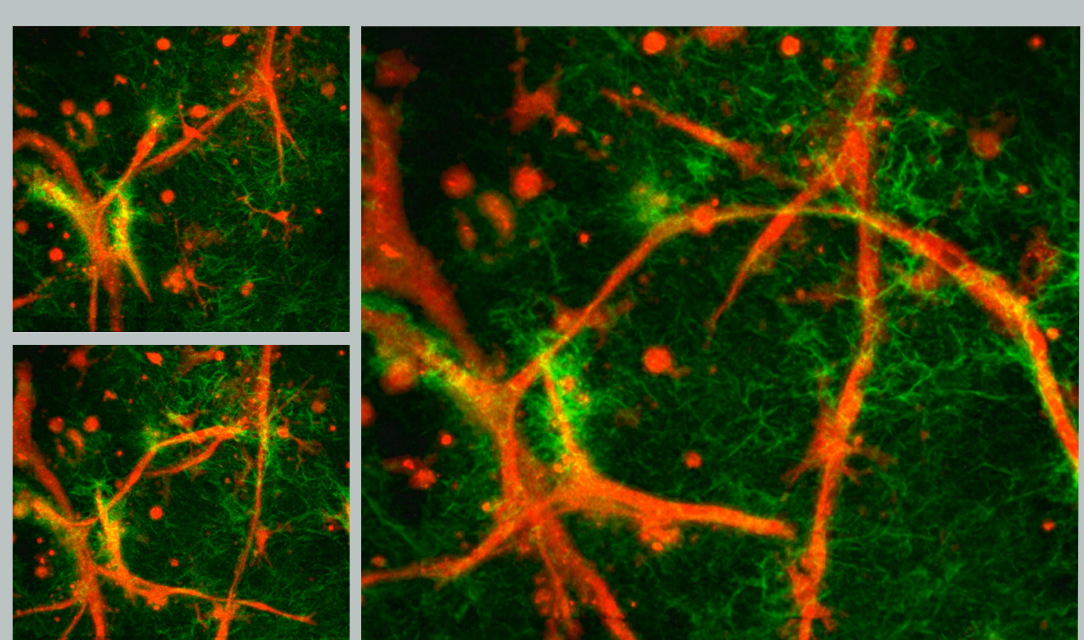
FluoRender
Yong Wan, Hideo Otsuna, Charles Hansen, Chi-Bin Chien



FluoRender is an interactive rendering tool for confocal microscopy data visualization. It combines the renderings of multi-channel volume data and polygon mesh data, where the properties of each dataset can be adjusted independently and quickly. The tool is designed especially for neurobiologists, helping them better visualize their fluorescent-stained confocal samples.

- Multi-Channel Inputs
- Transfer Functions
- Tone Mapping
- Filtering
- Multiple Rendering Modes
- 4D Data Inputs
- Overlays
- 4D Equalization

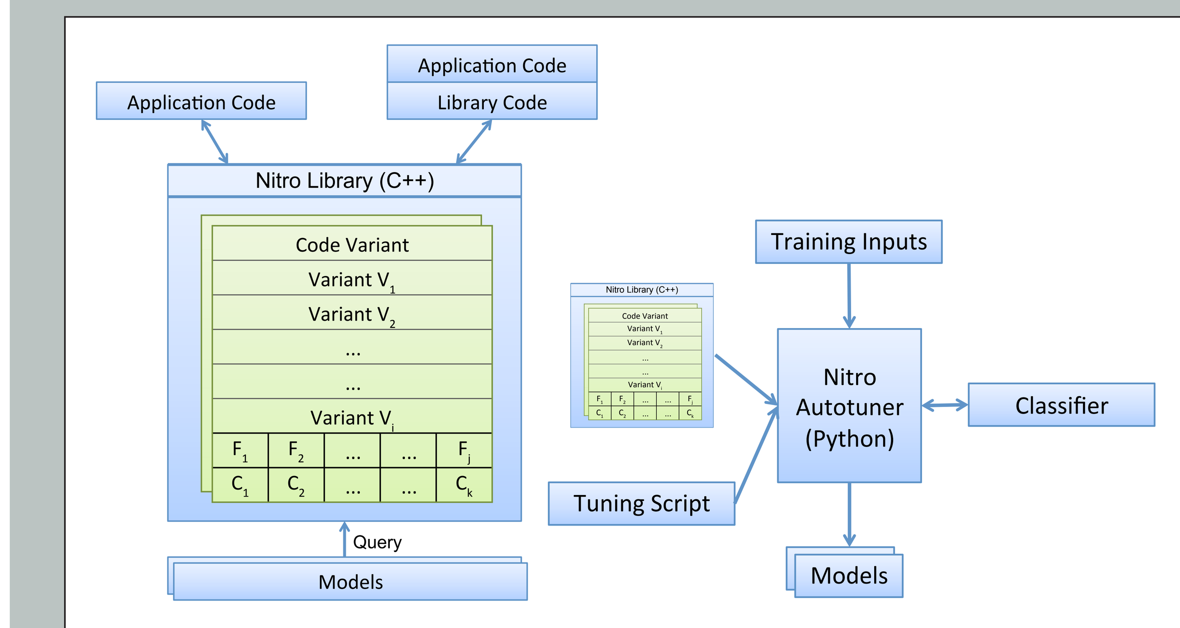
4D Two-photon Microscopy of Vascularized Construct
Urs Utzinger*, Brenda Baggett*, Lowell T. Edgar, **James B. Hoying, Jeffrey A. Weiss
University of Utah, *University of Arizona and ** University of Louisville



Live two-photon imaging was used to observe the dynamic features of angiogenic growth in an in vitro model of angiogenesis. A mosaic of four by five image stacks of a 3D vascularized construct was acquired every 2 hours over the course of two days. The resulting 4D dataset consisted of over 500 GB of data. This image shows a region of interest, rendered using FluoRender with an NVIDIA GeForce GTX TITAN graphics card. Neovessels sprout from the parent microvessel fragments and elongate into the extracellular space, forming a new vascular network. Live 2P imaging using GFP labeled cells (red channel) and second-harmonic generation (SHG) to visualize the collagen matrix structure allowed direct observation of neovessel sprouting, elongation and anastomosis and matrix reorganization in real time.

Nitro: An Adaptive Code Variant Tuning Framework
Saurav Muralidharan, Manu Shantharam, Mary Hall, Michael Garland*, Bryan Catanzaro* - University of Utah and *NVIDIA Research

- Selects variant to execute at run-time based on input data characteristics
- Builds a statistical model that maps from input characteristics to variants
- C++ and Python interfaces to specify variants, features, constraints etc. and to customize the tuning process.



Performance evaluated on five high-performance GPU libraries.

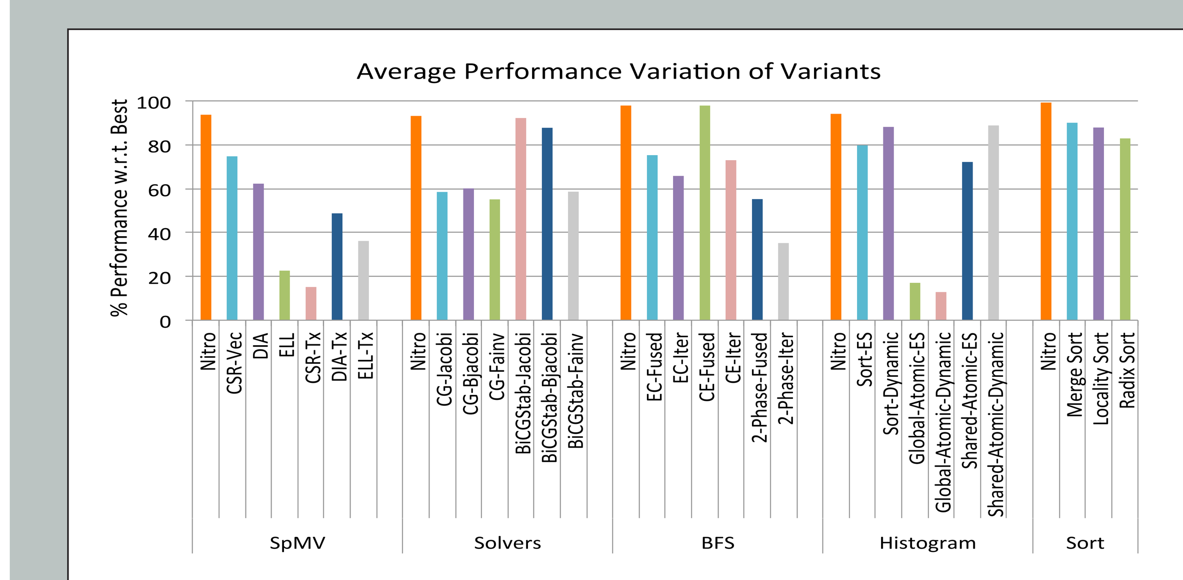
Sparse Matrix-Vector Multiplication
- 6 Variants from CUSP, 5 Features
- Training Set Size: 54, Testing Set Size: 100; Drawn from UFL matrix collection

Linear Solvers and Preconditioners
- 6 (Solver,Preconditioner) combinations from CULA , 8 Features
- Training Set Size: 26, Testing Set Size: 100; Drawn from UFL matrix collection

Breadth-First Search
- 6 Variants from Back40Computing, 5 Features
- Training Set Size: 20, Testing Set Size: 100; Drawn from DIMACS10

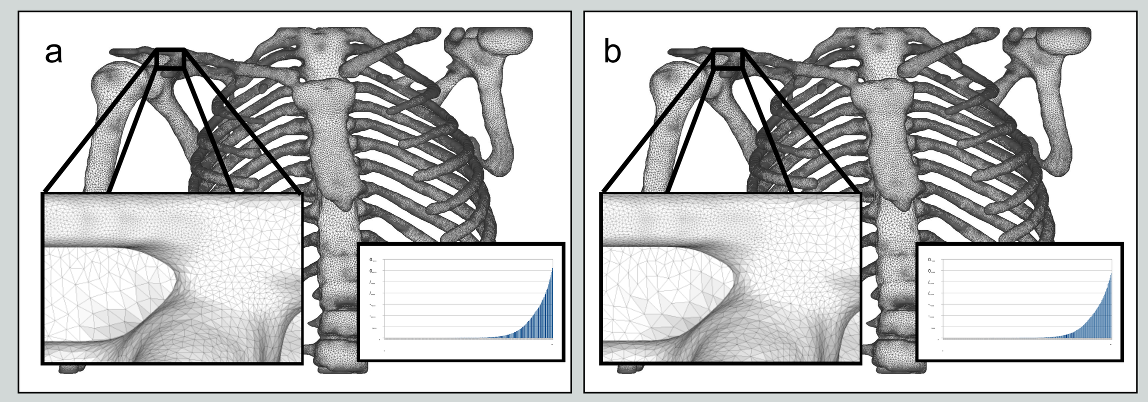
Histogram
- 6 Variants from CUB, 3 Features
- Training Set: 200, Testing Set: 1291; Images from INRIA Holidays dataset

Parallel Sort
- 3 Variants from ModernGPU and CUB, 3 Features
- Training Set Size: 120, Testing Set Size: 600; Generated

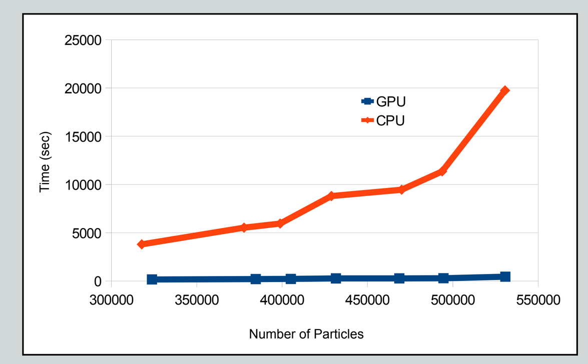


Dynamic Particle System on the GPU
Mark Kim, Guoning Chen and Charles Hansen

Extracting isosurfaces represented as high quality meshes from three-dimensional scalar fields is needed for many important applications, particularly visualization and numerical simulation. One recent advance for extracting high quality meshes for isosurface computation is based on a dynamic particle system. Unfortunately, this state-of-the-art particle placement technique requires a significant amount of time to produce a satisfactory mesh. To address this issue, we utilize the parallelism property in the particle placement and combine it with the CUDA implementation, a parallel programming technique on the GPU, to significantly improve the performance. We have applied our GPU based particle placement to a number of data from bioengineering where particle system is frequently used to generate isosurface meshes for simulations. Our results show comparable quality to the meshes generated using conventional CPU based particle system with at least ten fold speed up for most data.



Above: Images of ribcage data set, GPU (a) and CPU (b). Further, embedded is a zoomed in area for each image and the histogram for the data sets. The visual quality of the CPU implementation compared to the GPU implementation is very similar across the data sets. The histograms show that both the CPU and GPU systems are dominated by well-shaped triangles.



Timing results, in seconds, as the number of particles increase. The GPU times are in blue while the CPU times are in red.

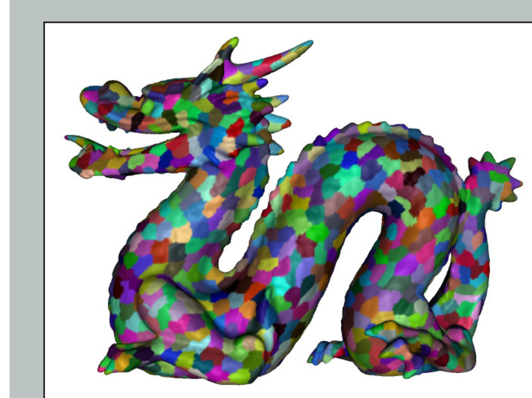
A Fast Iterative Method for Solving the Eikonal Equation on Triangulated Surfaces
Zhisong Fu, Won-Ki Jeong, Yongsheng Pan, Robert M. Kirby, Ross T. Whitaker

Mesh Fast Iterative Method (meshFIM)
- An iterative computational technique to solve the Eikonal equation efficiently on parallel architectures.
- This method relies on a modification of a label correcting method.
- The core elements for our FIM based method are:
  (1) Upwind scheme: calculate the value at a vertex with the values of the solved vertices.
  (2) Active list management: Active list contains the patches which has wave front vertices. If an active patch is convergent, it is removed from the Active list and its neighbor patches are added to this list.
  (3) Patch-based iteration: divide the whole mesh into patches to fit into GPU cores.
  (4) Triangle-based Jacobi update: update all the triangles inside a patch concurrently with parallel threads and each thread updates values of the three triangle vertices.

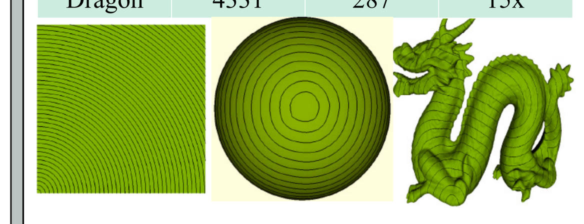
Suitability for GPU
- Each vertex updates independently
- According to the algorithm, update operation can be completed concurrently
- Computing only depends on the neighbors of same facet at every time step

Result
- CPU: Intel i7 920, 2.66GHz, 8M cache
- GPU: Nvidia GTX 275, 1.404GHz, 240 core

We test running time (ms) for a CPU version of meshFIM to compare with GPU version on three different meshes:



Mesh	CPU	GPU	Speedup
Square	6562	201	33x
Sphere	8591	415	21x
Dragon	4331	287	15x



The University of Utah Carbon Capture Multi-Disciplinary Simulation Center (CCMSC)
Phil J. Smith, Martin Berzins, Alan Humphrey

One of three large DOE NNSA PSAAP II Centers, CCMSC aims to use simulations at petascale and eventually exascale to facilitate the design of the next generation of clean coal boilers that will improve clean coal technologies for the generation of electric power.

Radiative Heat Transfer

The Uintah open source framework (www.uintah.utah.edu) has been one of the first computational frameworks to deploy methods such as the Discrete Ordinates method (developed at LANL) for radiative heat transfer in CFD applications. Uintah is now taking advantage of the petascale hardware and the advances in Monte Carlo ray tracing technology to develop an efficient and scalable solution to radiative heat transfer. Our approach, Reverse Monte Carlo Ray Tracing (RMCRT), lends itself to scalable parallelism because the intensities of each ray are mutually exclusive and amenable to domain decomposition. However, the all-to-all nature of radiation requires information about the entire computation to be available to each computational cell. To address this issue, we are currently developing scalable CPU and GPU multilevel RMCRT algorithms that take advantage of increased resolution of each ray in the near field while using coarse grid information from the far field. This is accomplished by using multilevel structured AMR (below). Our prototype GPU implementations have shown to be an order of magnitude faster than the CPU counter part, while retaining the required accuracy.

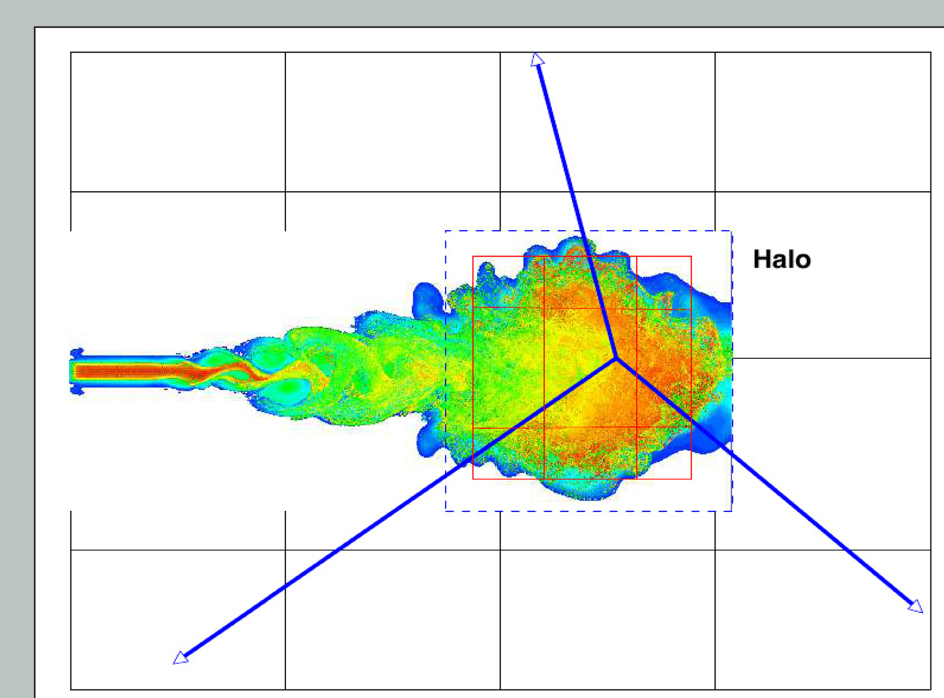


Illustration of multilevel RMCRT algorithm showing 3 levels of mesh refinement (red, blue, black), for the near field at the conjunction of three illustrative rays of radiation.