LA-UR- 04-7045

Approved for public release; distribution is unlimited.

Title: Scout: A Hardware-Accelerated System for Quantitatively Driven Visualization and Analysis (Talk)

Author(s): Patrick S. McCormick
Jeff Inman
James P. Ahrens
Charles Hansen
Greg Roth

Submitted to: Slides for presentation at IEEE Visualization 2004

Los Alamos
NATIONAL LABORATORY

Abstract

Quantitative techniques for visualization are critical to the successful analysis of both acquired and simulated scientific data. Many visualization techniques rely on indirect mappings, such as transfer functions, to produce the final imagery. In many situations, it is preferable and more powerful to express these mappings as mathematical expressions, or queries, that can then be directly applied to the data. In this paper, we present a hardware-accelerated system that provides such capabilities and exploits current graphics hardware for portions of the computational tasks that would otherwise be executed on the CPU. In our approach, the direct programming of the graphics processor using a concise data parallel language, gives scientists the capability to efficiently explore and visualize data sets.

VIS 2004 austin, texas

Scout: A Hardware-Accelerated System for Quantitatively Driven Visualization and Analysis

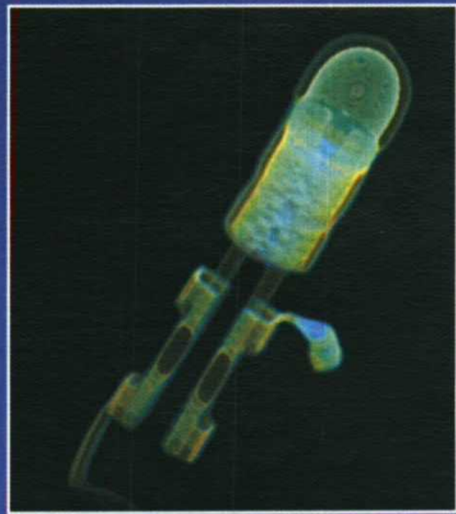
Patrick S. McCormick, Jeff Inman, James P. Ahrens
Los Alamos National Laboratory

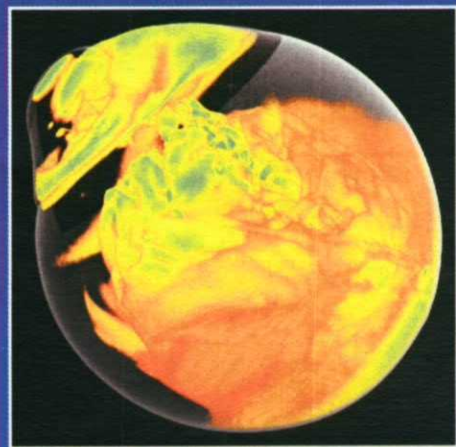
Charles Hansen and Greg Roth
University of Utah

LA-UR-????

Overview

- Motivations
- Scout
  - Language, GPU, and runtime
- Examples
- Future work and conclusions



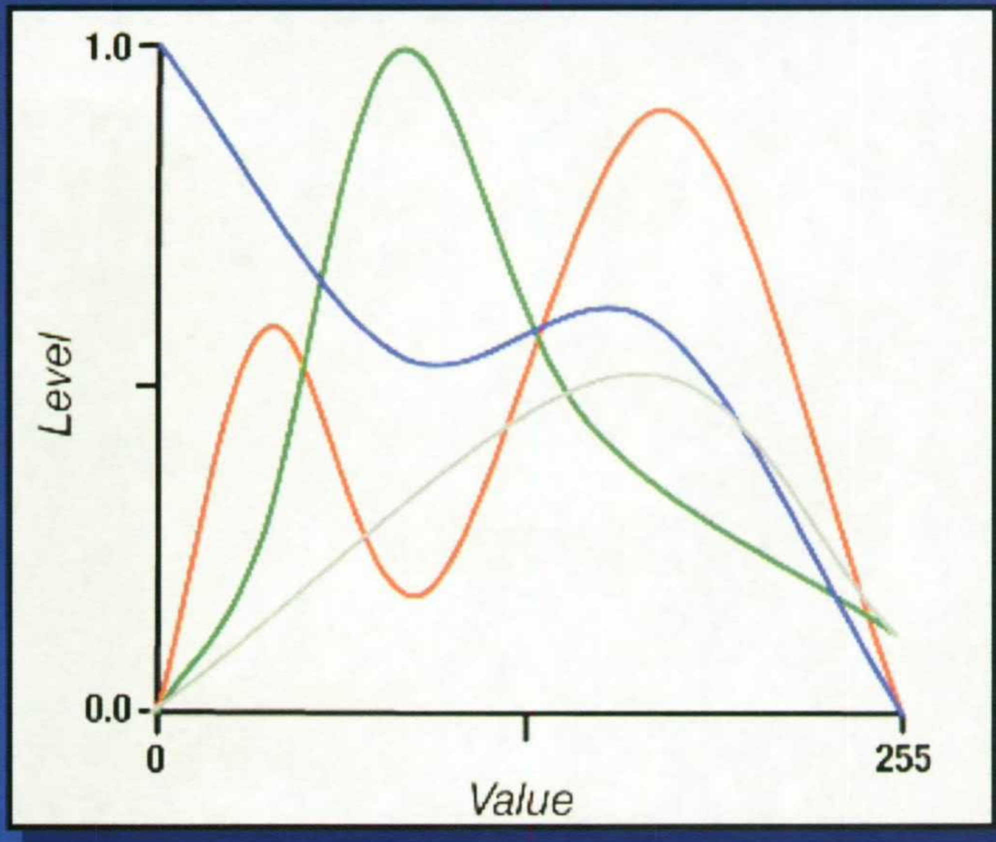


Motivations

- Scientists need quantitative and qualitative tools
  - Take steps towards merging visualization and analysis
  - Interactivity
  - Many approaches force scientists to work in the wrong domain
  - Volume rendering is a perfect example…

"Rise and Fall" of Volume Rendering

- Hardware-accelerated direct volume rendering fast!
  - Users thrilled!
- Users meet the transfer function editor…
  - Users appalled!
- Why?
  - Wrong/awkward domain
  - Too qualitative
  - Difficult user interface



What If?

- We allowed scientists to work in the data domain?
- Provided better quantitative controls?
- Allow this to be done programmatically?
- Leverage the GPU as a coprocessor?
  - High GFLOPS rate
  - High peak memory bandwidth
  - E.g. Pat Hanrahan's slides from ACM GP^2 Workshop:
    - http://www.graphics.stanford.edu/~hanrahan/talks/gp2/

Scout Language

- Goals
  - Simple and concise
  - Reveal the parallel nature of graphics hardware
  - Hide any nuances introduced by graphics API and hardware
  - Allow for both general computations and visualization results

- Recent GPU Languages
  - Brook: General purpose computation
    - *"Brook for GPUs: Streaming Computing on Graphics Hardware", Buck et al., SIGGRAPH 2004.*
  - Sh: Graphics focused
    - *"Shader Algebra", McCool et al., SIGGRAPH 2004.*
  - Others: Cg, OpenGL Shading Language

Scout Language

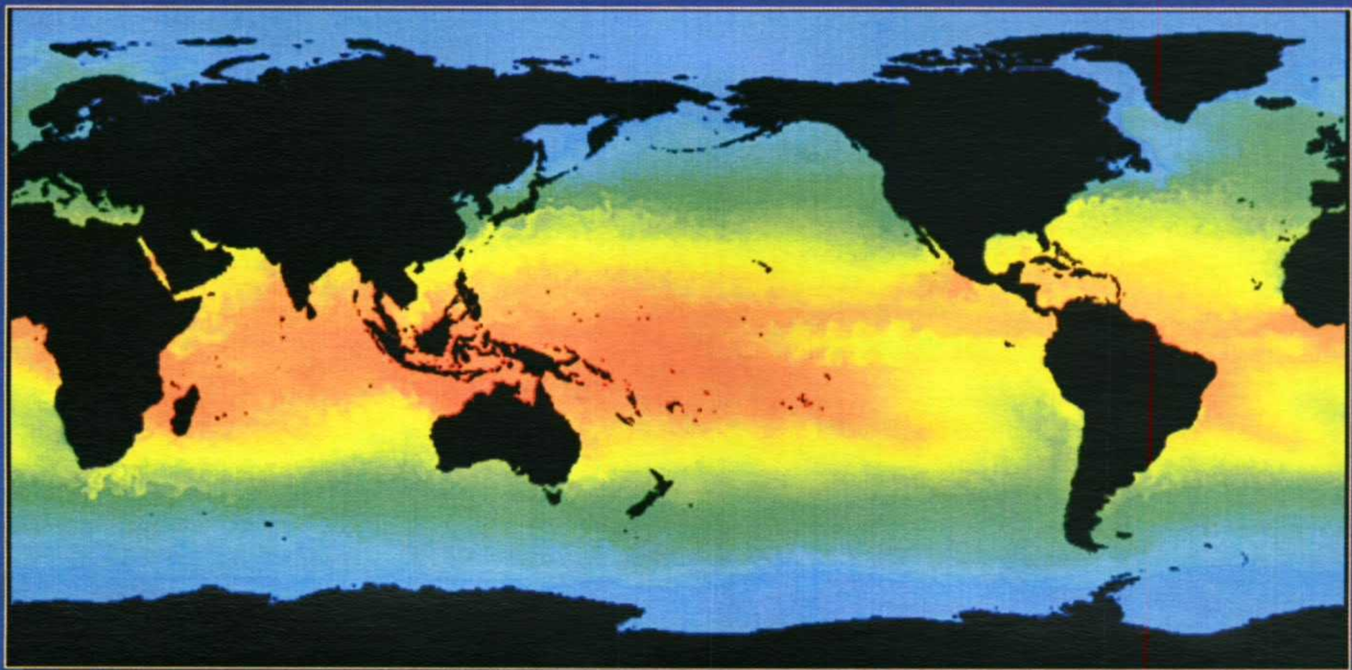
- GPU Limitations
  - Must use graphics API to program (OpenGL or DirectX)
  - Currently limited to regular grid structures (texture memory)
  - Precision concerns
    - Latest hardware 32-bit floats but only 16-bit blending
  - Programming model limitations -- no virtualization
    - Hard limits on instruction count, registers, etc.
  - Bus bandwidth

User Interface

- Data sets automatically in scope
- "On the fly" compilation
- Time series support

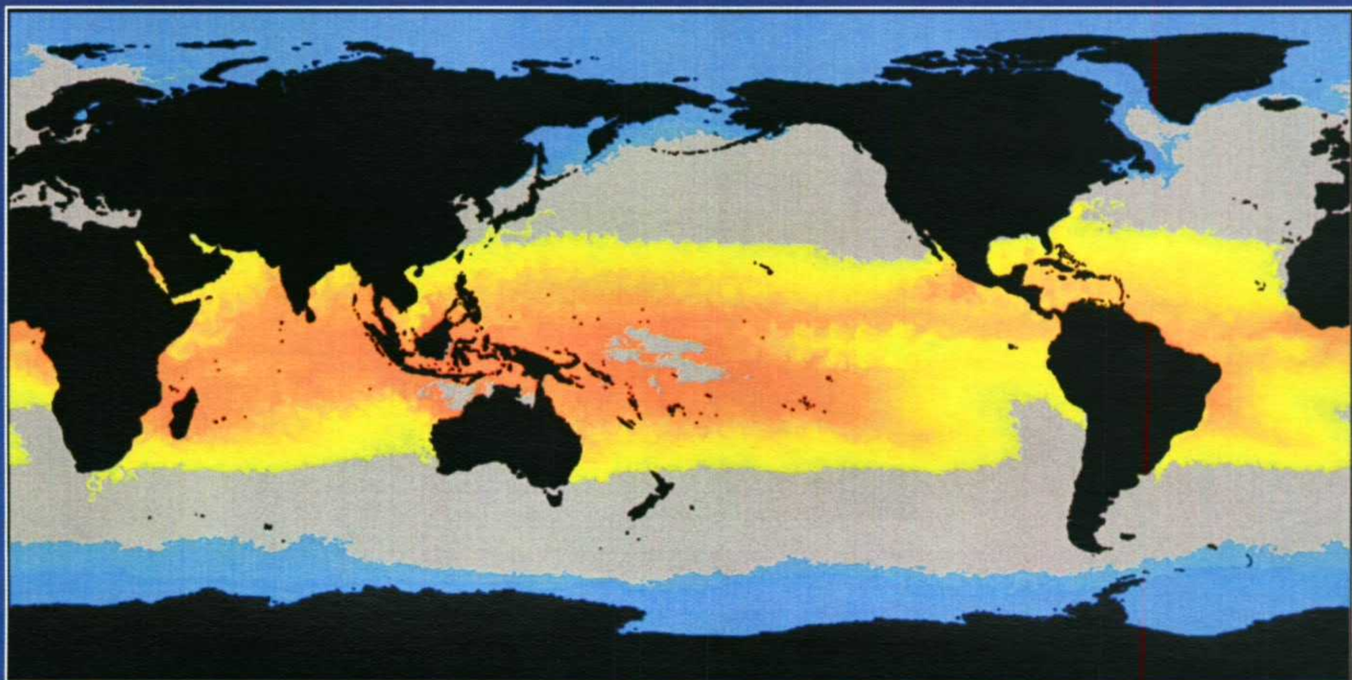
Example

```
// Color map potential temperature from blue
// (cold) to red (hot) using hsva() at all
// locations where there is water.
where (land == 0)
  image = hsva(240 - 240 * norm(pt), 1, 1, 1);
else
  image = 0; // black
```



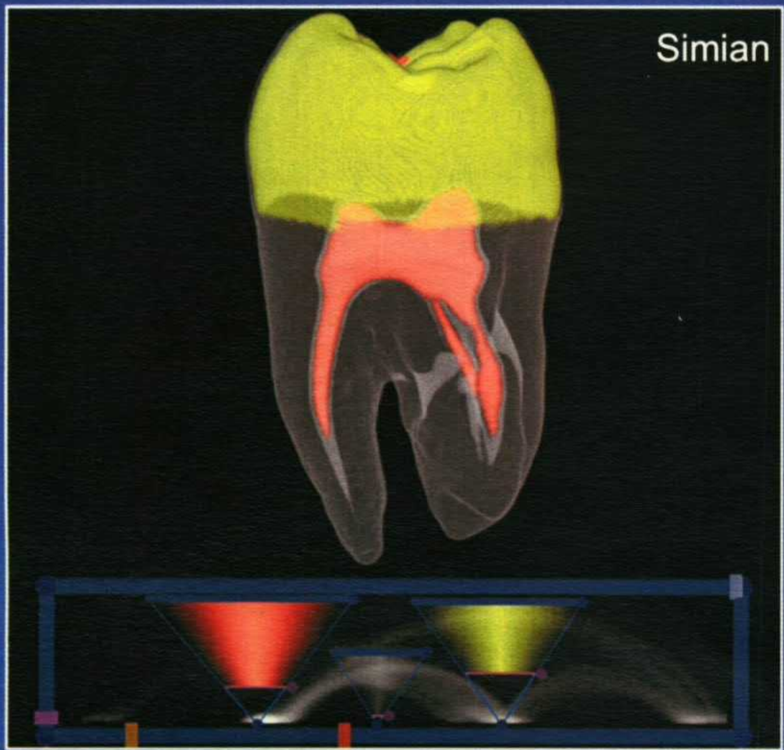
Example

```
where (land == 1)
  image = 0;  // render land as black
else where (pt < 2.375 || (pt >= 21 && pt < 29.5))
  image = hsva(240 - 240 * norm(pt), 1, 1, 1);
else
  image = 0.6; // outside range colored  gray
```



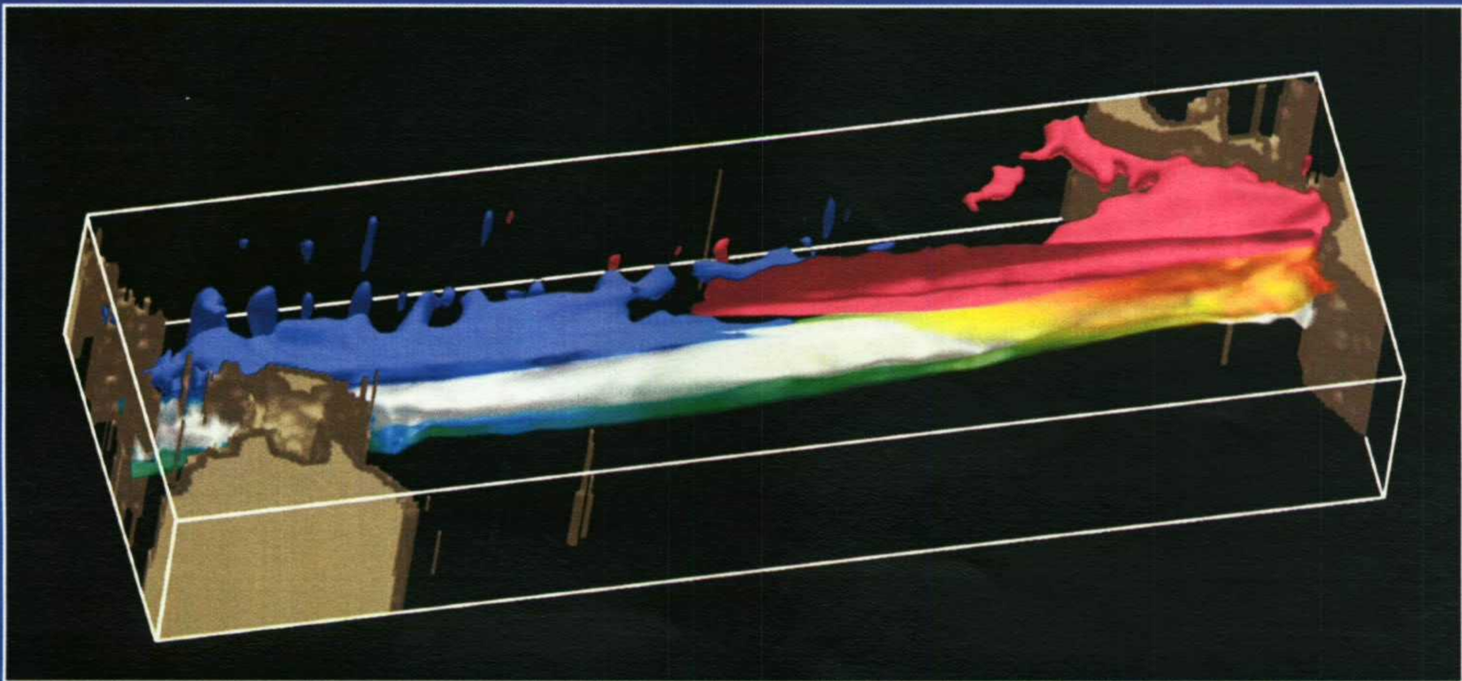
Multi-dimensional Transfer Functions

- Possible to program multi-dimensional transfer functions like the triangle widget from *Kniss et al. Visualization 2001*.
  - Language allows alternative to UI for higher dimensions
  - Lighting can also be implemented



Multivariate

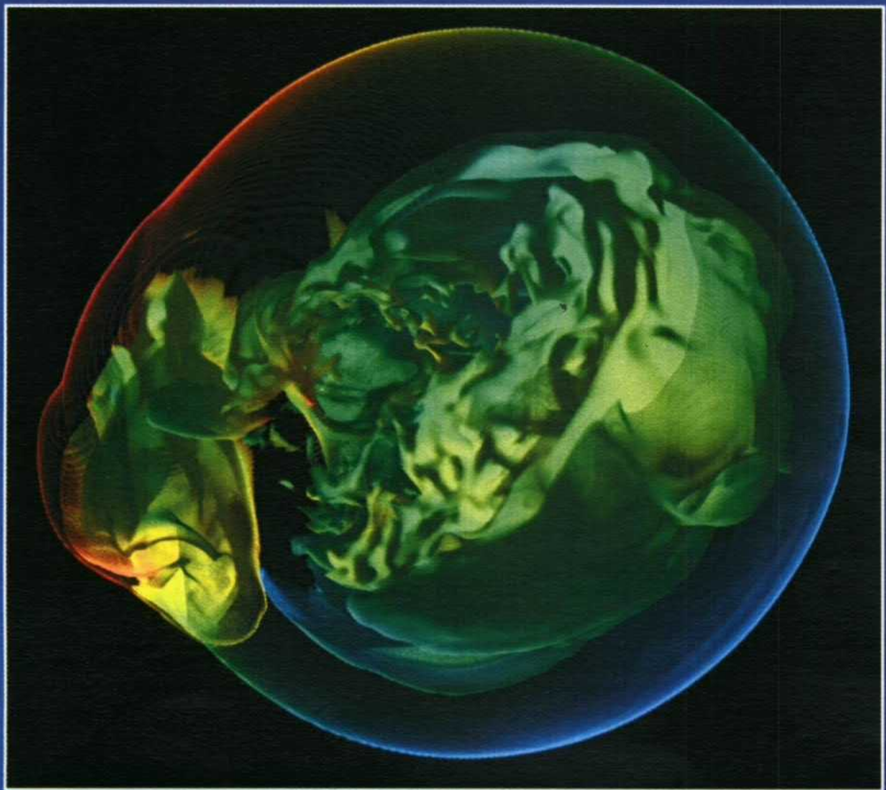
- More detailed multi-dimensional transfer functions
  - Ocean temperature differences exceeds 2.5° C (magenta), less than -2.5° C (blue) and clipped south of equator
  - Two thermoclines (values between 19° and 21° C): El Nino year (white), normal year (colored by sea surface height differences)
  - Land masses and lighting
  - 16 lines of code



Volume rendered data showing features consistent with an El Nino event.

Derived Fields

- The computation of derived values can be done as part of the visualization process.
  - Adds flexibility and performance



Volume rendered computed entropy of a core collapse supernovae simulation.

Computation times:

- CPU: *3.3 sec*
- CPU w/ SSE: *0.42 sec*
- GPU: *0.12 sec*

Amortization of texture download cost plays a key role.

Conclusion

- Allow scientists to work in the original data domain using a simple data-parallel language
  - Much easier than the traditional transfer function editor for quantitative exploration

- Leverage of the GPU as a coprocessor
  - Has both pros and cons

Future Work

- Additional mathematical/statistical operations
  - Important for analysis
- Parallelism to meet data set size requirements
- More flexible development environment
  - Incorporation of Mio - *Andrew Riffel, et al., Graphics Hardware 2004.*
- Other coprocessors?

Acknowledgements

- Funded by Mathematical, Information and Computational Sciences Program -- DOE Office of Science.
- John Blondin, Tony Mezzacappa - Terascale Supernova Initiative
- JoAnn Lysne, Matthew Hecht, Mat Maltrud - COSIM
- Nick Triantos - NVIDIA
- Mark Segal - ATI