Acknowledgements
This work was supported in part by CenSSIS, the Center for Subsurface Sensing and Imaging Systems, under the Engineering Research Centers Program of the NSF (Award Number EEC-9966821) and:
- Engineering Research Centers Innovations Program (Award Number EEC-0946463)
- CIBER, the Center for Integrative Biomedical Computing, an NIH / NCRR funded P41 center, grant P41-RR12553-10, which develops and maintains SCIRun and supports the third author of this poster.

Goals
Create an implementation of conjugate gradient (CG) in SCIRun that:
- Accelerates the performance of the algorithm on a GPU
- Is transparent to the scientist
- Is maintainable and replaceable

Introduction
SCIRun is a problem solving environment developed at the University of Utah.

- The conjugate gradient, minimal residual and Jacobi methods are some of the many algorithms in SCIRun that are suitable for acceleration using GPU hardware.
- Algorithms involving calculations on sparse data are challenging to accelerate on the GPU.
- A reproducible and adaptable code structure has been designed to allow up to 16x speedup via the GPU while remaining maintainable and replaceable.
- NVIDIA’s CUDA C and double precision floating point is used to achieve our results.
- System:
  - CPU: 1.86GHz Intel Core 2
  - GPU: NVIDIA GeForce GTX 280

Future Work
- Automation of the architecture choice between GPU and CPU at run-time.
- Acceleration of other algorithms.
- Further optimization of sparse linear solvers for GPU.

Research to Reality
- This project contributes to R3 by implementing general acceleration to be used with other environments.
- GPU acceleration is a viable and inexpensive way to achieve speedup in parallel computations.
- SCIRun is a powerful scientific tool that allows scientists to interactively design advanced simulations.

SCIRun Problem Solving Environment
SCIRun is a biomedical problem solving environment that allows scientists to create a network of mathematical functions.

Conjugate Gradient Algorithm
The CG algorithm contains parallelism in the Sparse Matrix-Vector multiply (SpMV) and other vector operations. Using CUDA C language, the parallel architecture of the GPU allows acceleration of algorithms like CG.

Conjugate Gradient Results on GPU

References
- Images from: NVIDIA CUDA Programming Guide and Geek3D.com