The Use of Graphics Hardware in Computing Wind Velocity from Aerosol Lidar Data

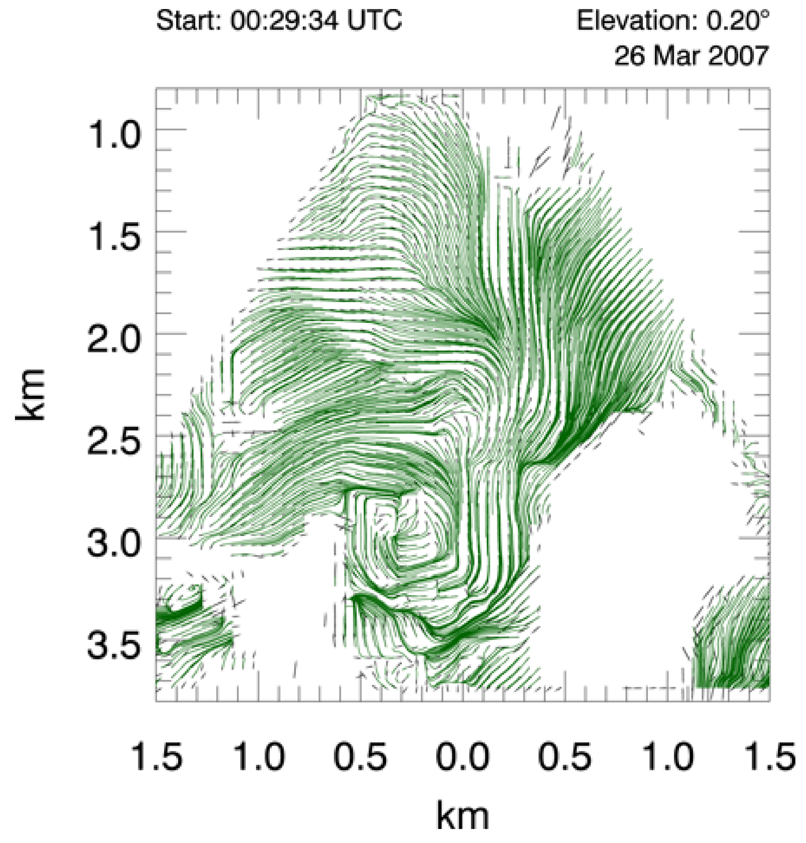
Chris Mauzey Jennifer Lowe Shane D. Mayor

Department of Physics, California State University Chico

Motivation

The Atmospheric Lidar Research Group is investigating the use of GPUs for the fast calculation of wind velocities from Raman-shifted Eye-safe Aerosol Lidar (REAL) data. Prof. Shane Mayor has been researching cross correlation as a method to measure wind velocities in aerosol profiles, and has amassed 3 months worth of lidar data from a previous experiment called CHATS[1]. To compute the wind vector fields from all CHATS data would require an impractical amount of time using conventional workstations. Our goal is to accelerate our current software by executing its most time-consuming algorithms on GPUs. The success of the method being presented will lead to the future goal of computing wind velocities from REAL's data in real time.

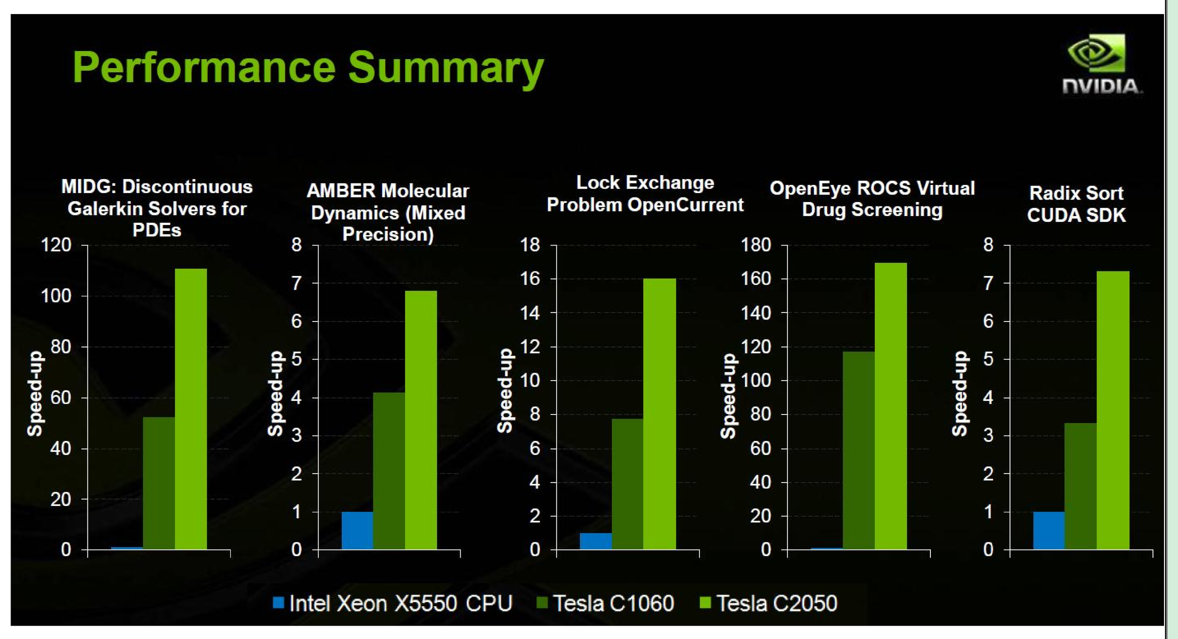
Fig. 1: Example of wind vector fields



GPU Computing

The graphics processing unit (GPU) is a processor that was designed specifically for real-time rendering of computer graphics, found in most computers as integrated display chips or video cards. However, recent trends in programmable graphics hardware have lead researchers to experiment with non-graphics related applications that run on GPUs. General purpose computing on GPUs, or GPGPU, has become a hot topic in high performance computing due to the massively parallel nature of graphics hardware. NVidia, a graphics hardware company, has been leading the GPGPU movement with their Compute Unified Device Architecture (CUDA) technology. CUDA C provides an extension to the C language that can easily harness the computing performance of nVidia's GPUs through massively parallel kernels.

Fig. 2: GPU speed-up over CPU for various applications. *Courtesy of nVidia.*



Methodology

Fig. 3: Chris Mauzey installing a Tesla C2070 into the workstation used to compute these results.

Application

Our application extracts REAL scan data from files and generates aerosol profiles, which are used to calculate wind velocity. Each scan consist of many data points that are in polar coordinates with the REAL at the origin. The data along the projections of the scan undergo a 1D high-pass median filter. The data is then interpolated from a polar grid to a rectangular grid. The resolution of this grid being determined by what grid cell size is used: the lower the cell size, the higher the resolution. With the aerosol profiles in uniform rectangular grids, subsections of the image are extracted to calculate the velocity at that subsection. Using the subsection of a previous profile, the cross correlation function is computed. Cross correlation determines where features in the previous profile have been displaced in the current profile, and with the known time difference between the profiles the velocity is calculated.

Software

This application was originally implemented by Prof. Mayor in Interactive Data Language (IDL). IDL does not natively support GPU computing, but the company Tech-X Corp has provided GPULib, a library that serves as an interface between IDL and CUDA. This library provides various routines that are accelerated by the GPU. However, the current version of GPULib lacks a median filter routine. The median filter used in this application was implemented in CUDA C and used by IDL as an external function.

Hardware

Our workstation, shown in Figure 3, contains the following:

- 1 six-core 3.33GHz Intel Xeon 5680
- 12GB 1333MHz DDR3 RAM
- 2 nVidia 3GB Tesla C2050
- 1 nVidia 6GB Tesla C2070

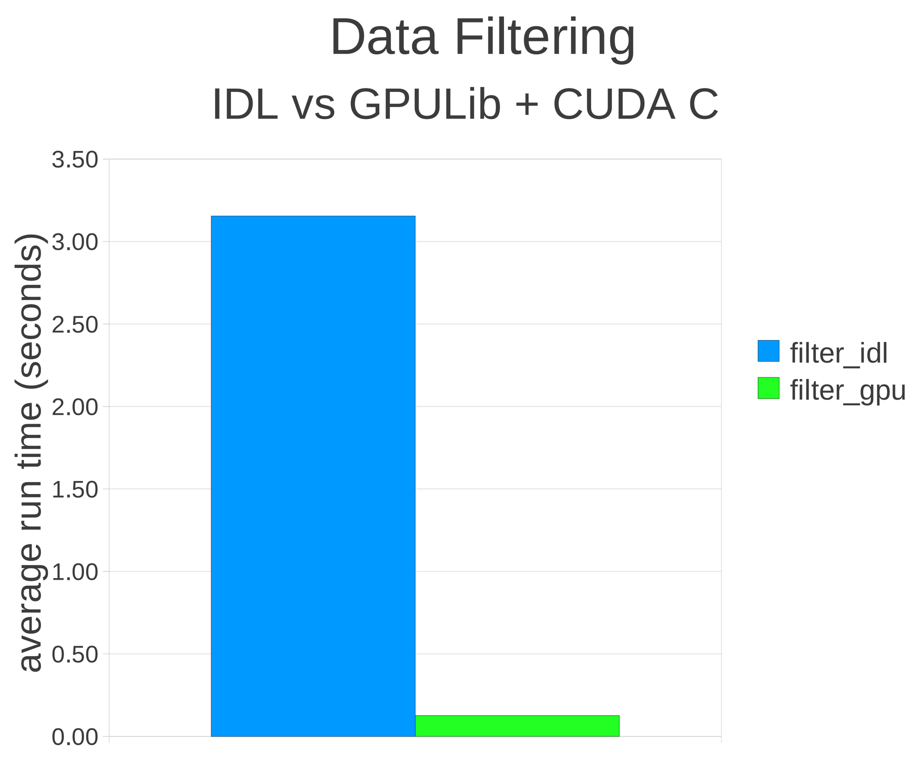
For the GPU performance test, only the C2070 was used.

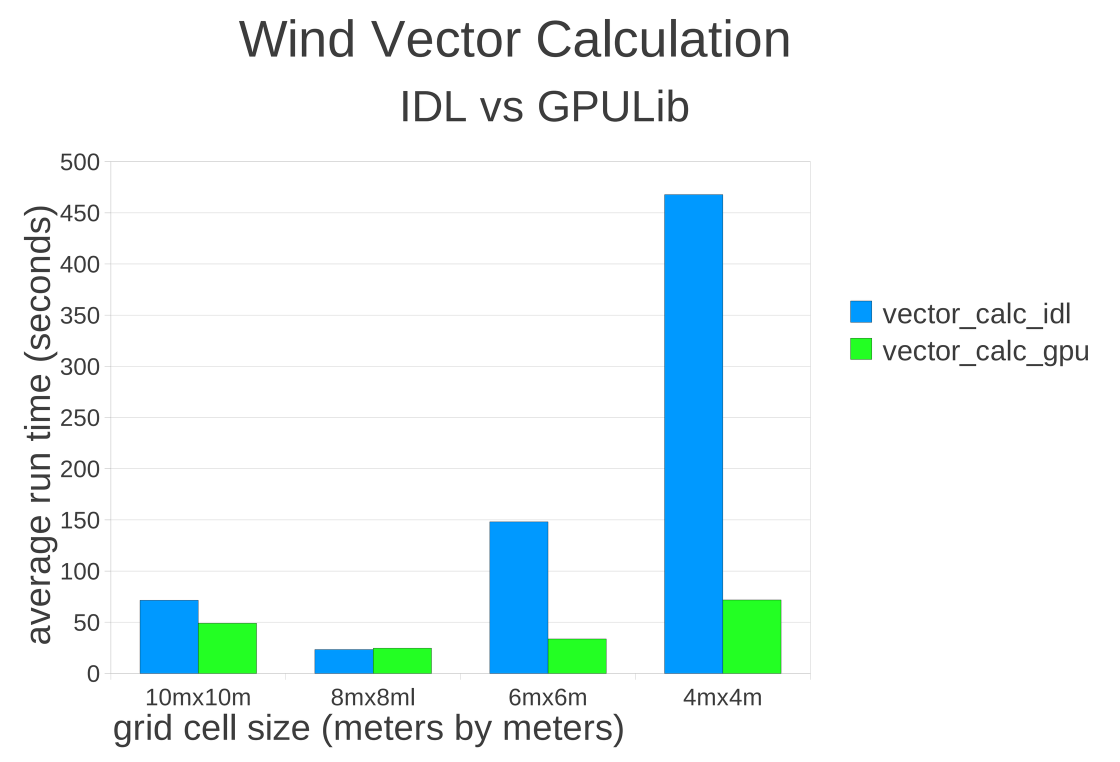
Performance Test

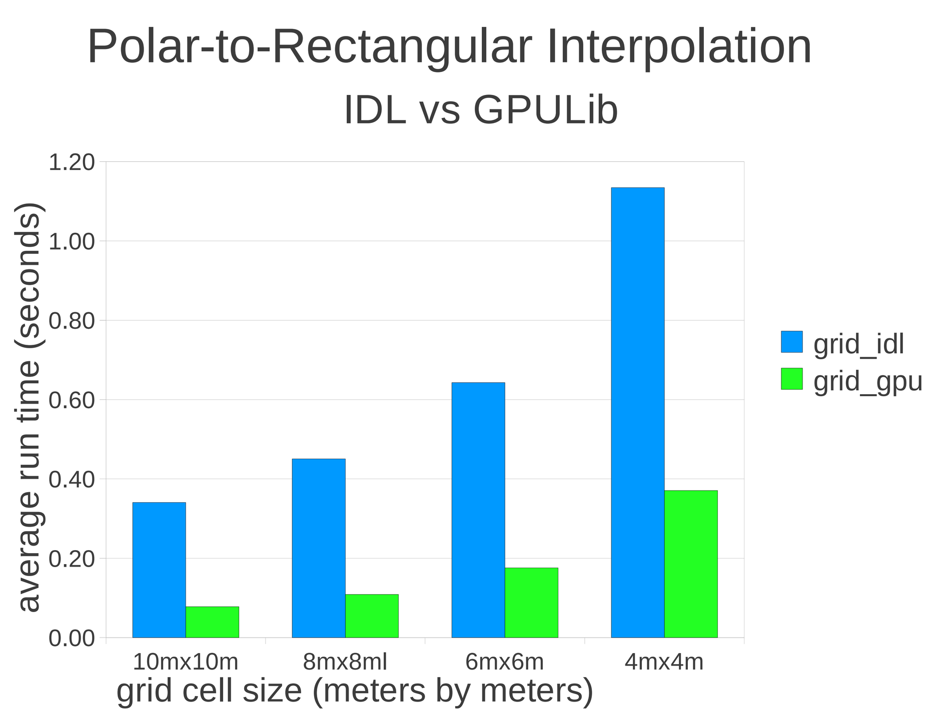
The test case used will generate velocity fields where vectors are spaced every 50m apart from each other in a 6km by 5km area. Each vector will be calculated from a 1km by 1km block with the vector in the center. The data used contains scans from March 26, 2007. It has 159 scans, each with 150 projections, and with each projection containing 7500 data points. The grid square cell size goes to 10m, 8m, 6m, and 4m.

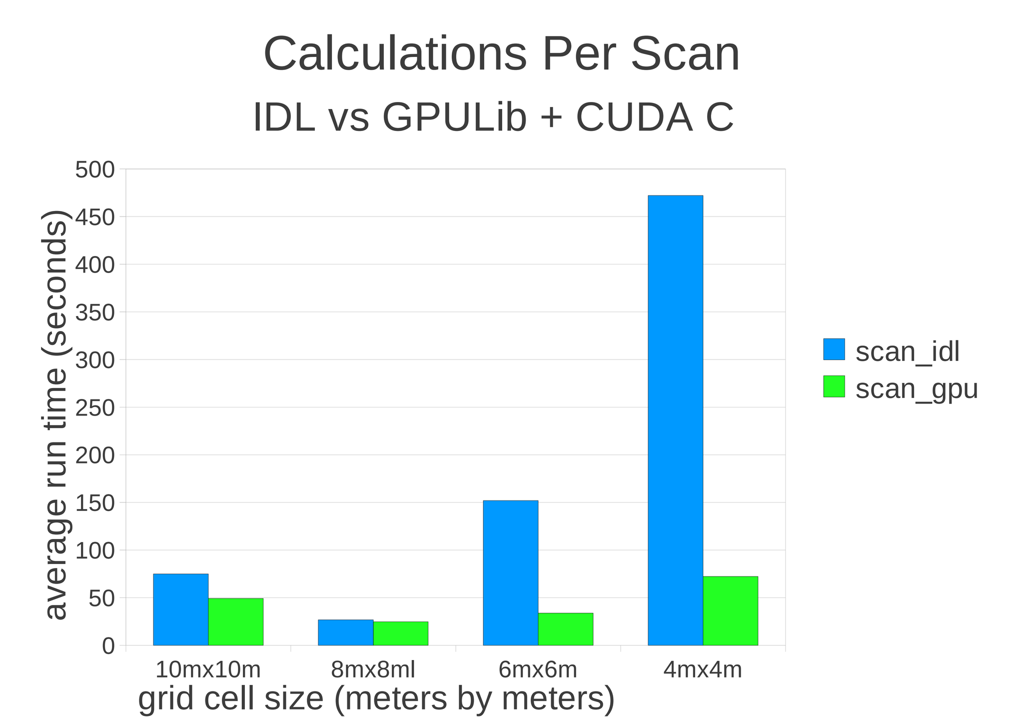
Two versions of the application will be tested: one with only IDL routines, and another with GPULib routines and a custom CUDA C median filter.

Results









The filtering of the lidar scan data shows the greatest GPU:IDL speed ratio of approximately 25:1. Interpolation of data from polar to rectangular coordinates shows a GPU speed-up of 3x-4x over IDL. The performance measurements for the wind vector calculations shows that it takes the majority of the application's run time, with the GPU version being 5x-6x faster than IDL. Interestingly, the 10mx10m spacing takes roughly twice as much time as the 8mx8m, where the GPU is marginally slower than IDL for wind vector calculations.

Conclusion

We have observed some impressive speed-ups in the GPU accelerated version of the application. Although speed-ups were not very impressive for low resolution aerosol profiles, they improved at higher resolution. The wind vector calculation part of our application is an area with more room for improvement. Greater performance might be achieved using custom-made CUDA accelerated routines for cross correlation.

References

1. Mayor, S. D., 2011: Two-component horizontal motion vectors from scanning eye-safe aerosol lidar and cross-correlation. Presented at the 5th Symposium on Lidar Atmospheric Applications. American Meteorological Society, 26 January 2011, Seattle, WA.

Acknowledgements

This work was funded by Grant 0924407 from the National Science Foundation’s Physical and Dynamic Meteorology Program.

Thanks to nVidia for donating a Tesla C2070 card for our research. Thanks to Tech-X Corp for GPULib.