NVIDIA 2009 GPU Technology Conference

GPU Acceleration of the Scientific Data Analysis Package GDL

A. Jaffey[1,2], A. Kobashi[1,2], M. C. M. Cheung[1]

[1]Lockheed Martin Solar and Astrophysics Laboratory, [2]Stanford University

Abstract

GDL is an open-source scientific data analysis package compatible with the syntax of IDL, the interactive programming language commonly used by members of the astronomy community. The interactive nature of GDL allows users to perform mathematical operations on large numerical data arrays and then to conveniently display the results in graphical form. Many of the mathematical operations supported by GDL involve independent operations on the individual elements of arrays (matrices), which makes them suitable for parallelization by means of multithreading.

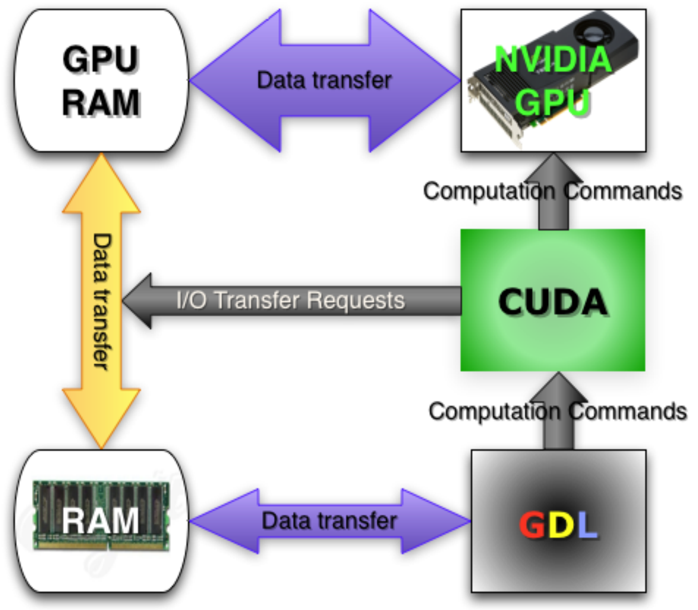
Introduction

GNU Data Language (http://gnudatalanguage.sourceforge.net) is an open-source compatible implementation of the popular scientific data analysis package IDL.

While emulating many of the features of IDL, GDL provides a more flexible solution:

- **It is released under the GNU General Public License**
- **It is easily adaptable to non-standard environments**

The open-source nature of GDL allowed us to easily modify it to support the use of graphics card acceleration using the CUDA interface.



Key features

Our adaptation of GDL provides acceleration of several features useful to scientists:

- **A FFT (Fast Fourier Transform) function that uses the CUDA FFT Library**
- **A function for spatio-temporal filtering of solar photospheric observations**
- **A function for deconvolution of images from solar telescopes**

See boxes below for detailed descriptions of each of these features and respective performance improvements.

In the future, we plan to extend GPU-acceleration to other common GDL functions such as matrix operations.

Interfacing Between GDL and CUDA

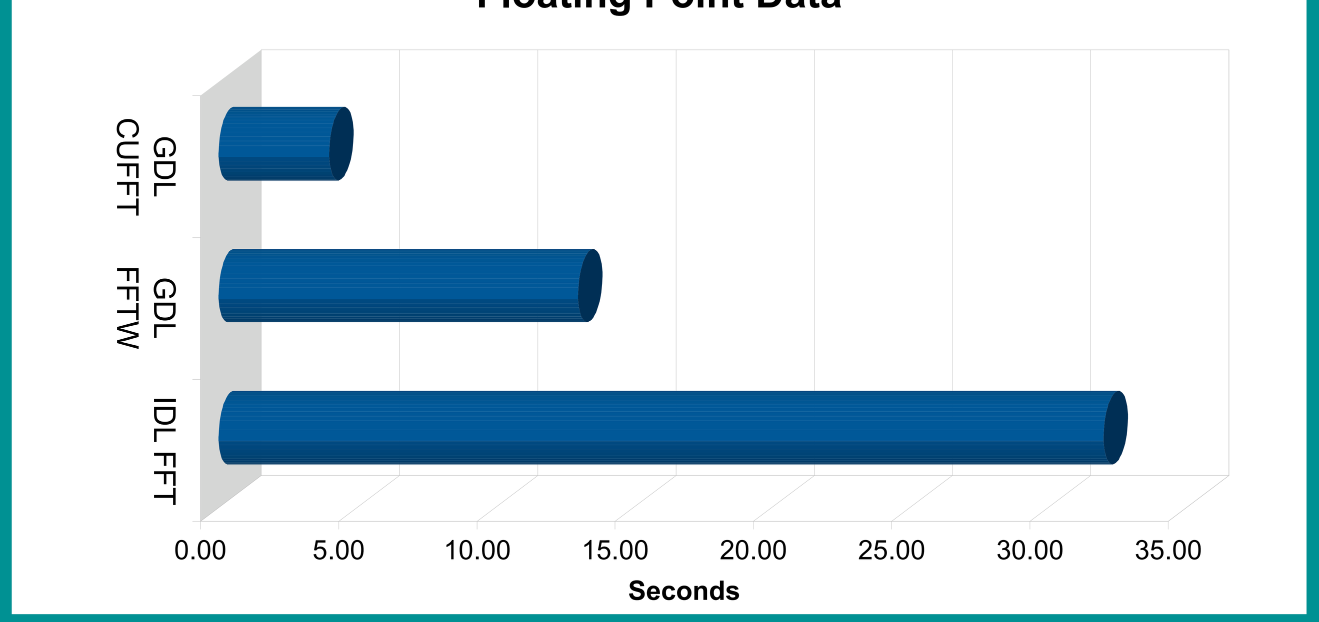
GDL is implemented in C++, and uses special container data types to represent the underlying data. As GDL is weakly typed, special care must be taken to ensure that data to be passed to CUDA is in the correct format (in our case, single-precision floating point). The manner in which arrays are indexed must also be considered since GDL arrays are stored in column-major order while CUDA and C++ normally operate on multidimensional arrays in row-major order.

CUDA provides a C library that we have linked with GDL, in order to use a variety of GPU functions. For example, using the CUDAFFT library, we implemented a CUFFT function in GDL to provide higher performance on large Fourier transformation computations.

On smaller data sets, the time needed to transfer data between the computer and card memory comprises a large fraction of the program's run time. In this case, it is often faster to run code only on the CPU. However, with larger quantities of data that are becoming increasingly prevalent, the multithreaded nature of GPU processing enables much faster computations, with only a small increase in I/O overhead. Due to this restriction, one should benchmark computations on specific data sizes using both CPUs and GPUs to determine the optimal configuration.

All tests and applications presented here were carried out on NVIDIA's Tesla C1060 GPU with 240 cores, on a Linux machine with an Intel Core 2 Duo processor running at 3 GHz.

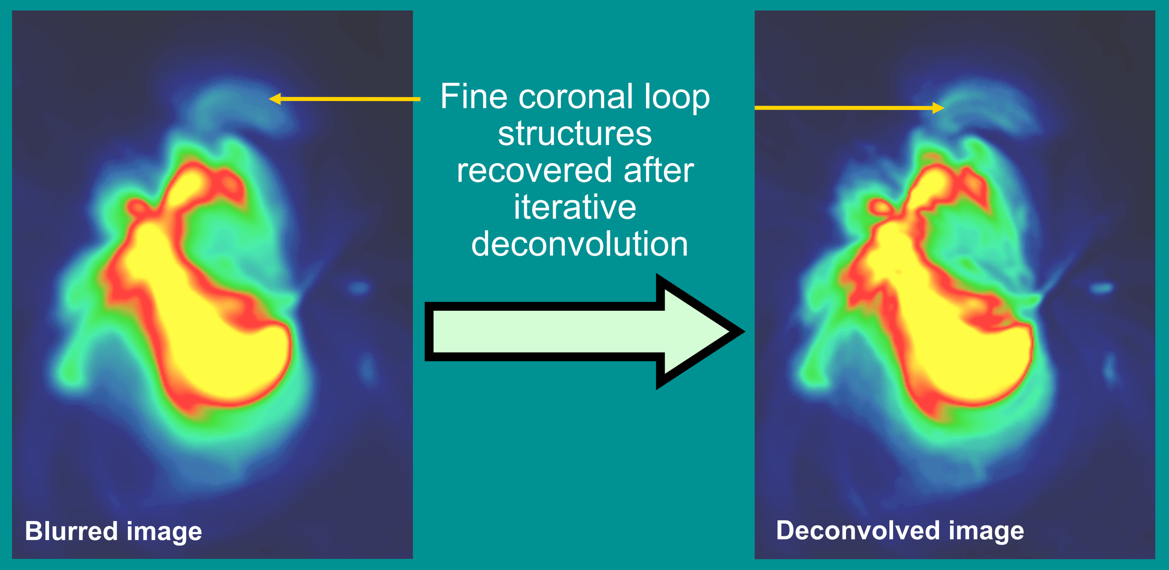
Speed Comparison of FFT on 1024x512x256 Array of Floating Point Data



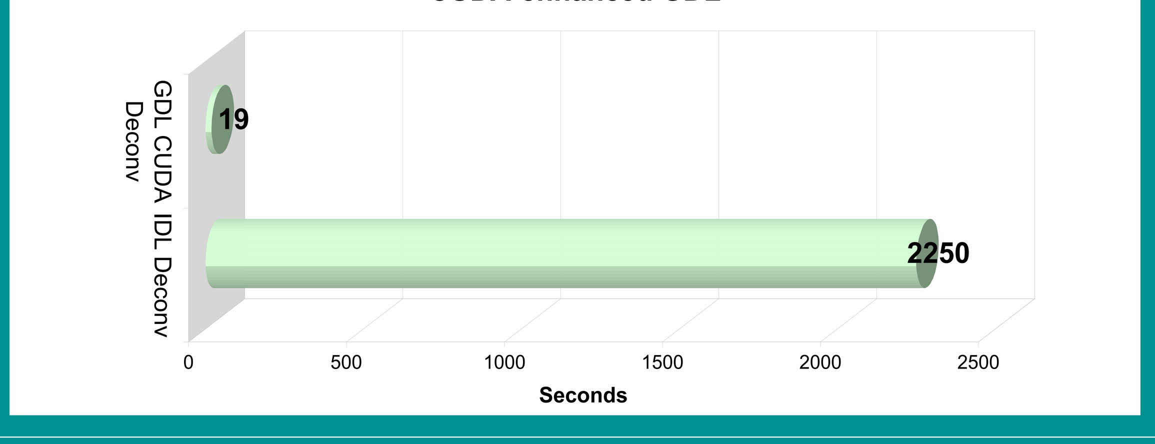
Applications to Solar Physics

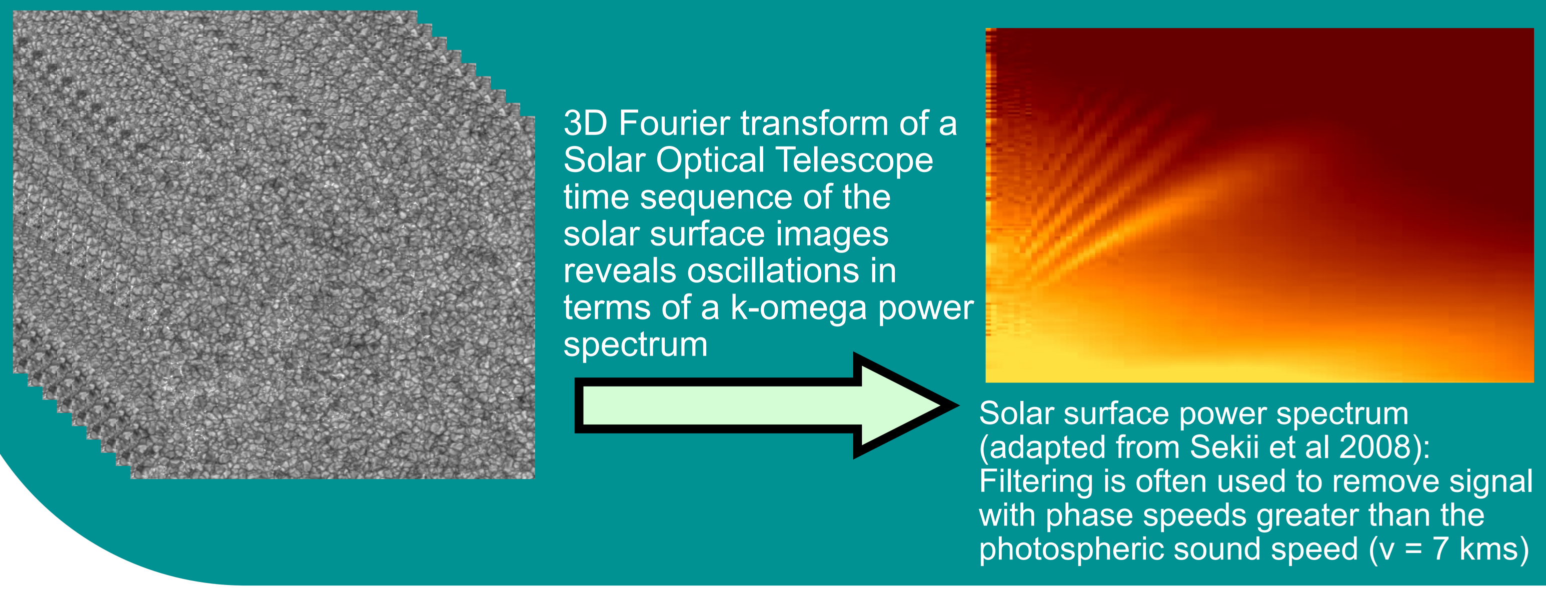
Deconvolution of images from solar telescopes

NASA's upcoming Solar Dynamics Observatory will continuously deliver 4096x4096 (at a data rate of 2 TB / day) images of the full disk of the Sun at different wavelengths. Deconvolution of these images will facilitate scientists to pick out the finest structures in the solar atmosphere. This example demonstrates how GPU-enabled GDL will dramatically accelerate this task. In the example on the right, an EUV (Extra-Ultra-Violet image) of the solar corona from NASA's STEREO mission was artificially convolved with a Gaussian kernel to blur the image. Applying the Richardson-Lucy deconvolution iterative method 100 times results in the deconvolved image on the far right.



Speed Improvement of Richardson-Lucy Deconvolutions using CUDA-enhanced GDL





Spatio-temporal filtering of solar photospheric observations

Continuous observations of the solar photosphere reveal resonant oscillations at characteristic ridges in Fourier-space. The power spectrum also reveals signal at supersonic phase speeds. Filtering of these signals is often performed to isolate certain types of waves for the study of different phenomena. This example illustrates example of a GDL function that performs such a task.

Acknowledgements

This work is supported by the NASA grants for the Hinode/Solar Optical Telescope (NNM07AA01C) and Solar Dynamics Observatory / Atmospheric Imaging Assembly (NNG04EA00C) projects.

IDL is a registered trademark of ITT Visual Information Solutions (http://www.ittvis.com/)

We express our gratitude to the developers of GDL for making it freely available to the public.