COMPUTATION OF EARTH SCIENCE PRODUCTS ON SPACEBORNE PLATFORMS

Kevin Fisher[1], J. Anthony Gualtieri[2], Jacqueline LeMoigne[1], James C. Tilton[1]

[1] NASA Goddard Space Flight Center, Greenbelt, Maryland, USA

[2] LuxAnalytica LLC, Schenectady, New York, USA

ABSTRACT

Spaceborne sensors like NASA's Hyperion hyperspectral imager generate huge data volumes, and several near-term trends indicate that data volumes will only increase. Next-generation hyperspectral missions, such as NASA's Hyperspectral Infrared Imager (HyspIRI), will operate at higher duty cycles and higher data rates, and their users will expect products to be generated from the data in near real time [1]. Barring a sudden advance in satellite downlink capacity, these trends point to a need to process data and generate products onboard the spacecraft. Rather than downlink an entire hyperspectral image cube, onboard processing enables satellites to downlink partial or completed scientific data products, which are often one to two orders of magnitude smaller than the original image. In addition, a satellite with onboard data processing resources and direct broadcast transmission equipment could send data products directly to first responders, research scientists or other users on the ground.

Next-generation space-capable data processors will have a combination of reconfigurable gate arrays, digital signal processors and general-purpose CPUs. Correctly programmed and configured, these resources are sufficient to run sophisticated data analysis programs, including hyperspectral image processing algorithms that commonly run on desktop computers [2].

This paper describes how we implemented one such program, the HSEG hierarchical image segmentation algorithm, software commonly used on desktop and parallel processors, on a hardware platform designed to mimic a next-generation space-capable data processor [3]. We also describe our approach to porting the algorithm to and optimizing it for the new platform, and determine the expected performance gains enabled by our design. This extended abstract will describe the HSEG algorithm and hardware platform in greater detail, provide an analysis of the key function within the algorithm that required hardware acceleration, and describe our implementation of that function in hardware.

SYSTEM COMPONENTS

Onboard Processing Platform

We expect that in the near future, spacecraft will be designed with high-performance hybrid processors with general-purpose processors (CPUs) and large-scale field-programmable gate array (FPGA) units with integrated digital signal processing (DSP) resources. The hardware platform used in these experiments is a Xilinx ML507 development platform, based on the Virtex-5 processor. This specific model includes an integrated PowerPC 440 processor and an FPGA with 16,000 configurable logic blocks (CLBs) and 128 DSP blocks, all operating at 400 MHz. The board has a 100 MHz front-side bus to connect to its onboard memory. In our implementation, the PowerPC runs Linux, which allows the HSEG program to run as it would on a desktop computer. By modifying the code (HSEG is open-source software), we can use the FPGA and DSP resources to implement certain functions in hardware.

Hierarchical image segmentation

HSEG ("Hierarchical SEGmentation") is a hierarchical image segmentation program for hyperspectral images. It partitions images in the spatial domain into regions or clusters according to their similarity in the spectral domain. It operates by forming a distinct region around every pixel, then deciding which regions are similar enough to be merged. As regions merge, they become larger, and there become less of them. HSEG records the order in which regions are merged, so that the user can explore a rich representation of the structure within the data.

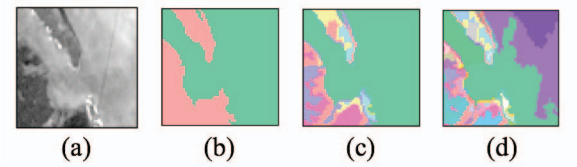

Figure 1. (a) 64x64 pixel hyperspectral image of New Orleans, and HSEG output showing (b) two, (c) five, and (d) eighteen regions.

Dissimilarity Criterion

One key function within HSEG is its dissimilarity criterion, the function it uses to judge how similar two regions are, and therefore at what point they should be merged [4]. The function can be defined in multiple ways, but in our implementation the function resembles the Euclidean distance between two points (a and b) in a space with as many dimensions (n) as there are bands in the hyperspectral image:

$$D(a,b) = \sqrt{\sum_{i=1}^{n} (a_i - b_i)^2} \qquad (1)$$

The larger the distance, the more dissimilar the two points are, and the lower the chance they will be merged. Each region is represented in the calculation by a representative pixel, derived from the representative pixels of the smaller regions that were merged to form the current region. (The recursive definition stops with the single-pixel regions, whose representative pixels are the same as the single pixel they contain.) Larger images contain more pixels, which require more region dissimilarity comparisons. Across a range of image sizes, HSEG spends 85% of its time computing region dissimilarity, making this function a ripe target for hardware optimization.

Hardware implementation

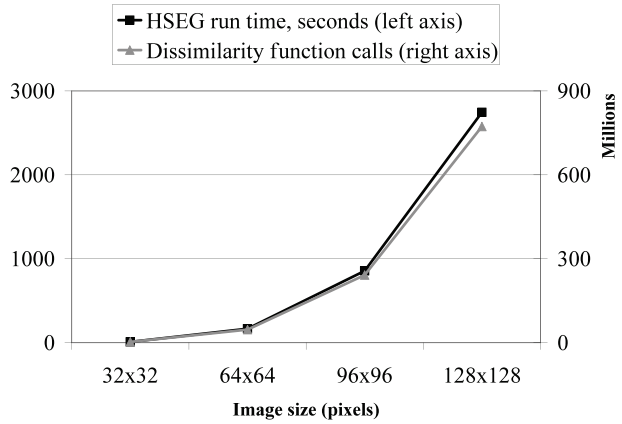

Figure 2. HSEG run time and the number of times the dissimilarity function is called both scale quadratically with the number of image pixels.

The dissimilarity function can be decomposed into the set of discrete arithmetic operations a computer must perform to compute it: one two-element subtract per image band (hyperspectral images typically contain about 200 bands), one large summation of all the differences, and a single square root. Using profiling tools on a desktop computer running HSEG, we find that each call to the dissimilarity function consumes roughly 6480 clock cycles. The goal of hardware acceleration would be to reduce the number of clock cycles needed to perform each call. With so much of the program run time devoted to this one function, even small efficiencies per function call can translate into large performance improvements.

Each of the hardware platform's 128 DSP blocks can perform one subtraction and one multiplication operation in parallel. Allocating all of them to this task, for example, would accomplish the subtraction and multiplication stages of the function in two clock cycles. However, these blocks are also useful for the summation stage: the blocks can be arranged in a binary tree configuration to perform the summation with logarithmic time complexity. The final square root operation, if implemented in hardware, would require a large proportion of the FPGA's logic blocks. Since this function is already implemented on the PowerPC CPU, it may make more sense to leave this stage to software.

Our full paper will present our detailed analysis of hardware acceleration of the HSEG similarity criterion, including the tradeoffs among possible allocations of the platform's limited hardware resources, and an analysis of the effect of data input/output between the CPU and FPGA.

BIBLIOGRAPHY

[1] S. Chien, et al., "Onboard Science Processing Concepts for the HyspIRI Mission," *IEEE Intelligent Systems*, pp. 12-19, November/December 2009.

[2] Tilton, J.C., "Image segmentation by region growing and spectral clustering with a natural convergence criterion," *Proceedings of IGARSS 1998*, pp. 1766–1768, 1998.

[3] Plaza, A. et. al., "Advanced Processing of Hyperspectral Images," *Proceedings of IGARSS 2006*, pp. 1974–1979, 2002.

[4] Gualtieri, J.A. and Tilton, J.C. "Hierarchical Segmentation of Hyperspectral Data," *Proceedings of the 2002 AVIRIS Earth Science and Applications Workshop*, Pasadena, CA, March 5-8, 2002.