POSSIBLE REAL-TIME FAST BACK PROJECTION ALGORITHM USING GRAPHICAL PROCESSING UNITS

Aaron Rogan, Richard Carande and David Cohen
Neva Ridge Technologies Boulder, CO 80301

For decades, the Back Projection (BP) algorithm has stood above all other processing algorithms in terms of simplicity as well as image fidelity. The algorithm is also the least restrictive in the sense that the output pixels and input sampled data are not constrained to be planar as they are in all Fourier based algorithms. The flexibility of the BP algorithm removes the burden of having to perform an additional orthorectification stage after image formation in an effort to compensate for the elevation of targets within the imaged scene. Even with all its capabilities, the BP algorithm has long been regarded as an impractical imaging algorithm for any real time applications. The time domain approach is simply to slow of an algorithm requiring $O(N^3)$ operations as opposed to the more attractive $O(N^2 \log N)$ alternative that Fourier based algorithms provide.

In the past fifteen years, various modified versions of the back projection algorithm have used factorization, beamforming, and various other techniques to significantly speed up the traditional BP algorithm [1, 2, 3]. Some implementations of the Fast Back Projection (FBP) algorithm now can perform near the same rate as their Fourier counterparts achieving close to $O(N^2 \log N)$ running times[3]. In most cases, however, the FBP is still roughly an order of magnitude slower than its Fourier counterparts, but it is fair to say that the FBP algorithms have significantly closed the previous gap that existed between the time domain and Fourier based approaches.

With sensor technologies rapidly improving, the need to process increasingly larger data sets is becoming the main bottleneck in many real time applications associated with persistence surveillance such as VideoSAR and volumetric SAR imaging [GOTCHA reference]. This is true whether one employs a time domain or a Fourier domain algorithm. Understandably, the performance improvements afforded by FBP algorithms prove attractive for such environments; however, the FBP algorithm is still incapable of meeting the strict requirements of some of the aforementioned real time applications.

The BP and FBP algorithms are commonly referred to as “embarrassingly parallel” algorithms [4]. It is fairly trivial to distribute the pulses to individual processor and simultaneously process the returns for each output pixel in the scene. A similar approach is currently being exercised to process the large volumes of data collected with the Airforce's GOTCHA staring SAR system. Using a variant of the FBP algorithm in conjunction with what is currently listed in the top 500 fastest supercomputers in the world, AFRL Dayton has demonstrated being able to process the spotlight data collected by the sensor to 1m ground postings in real time [need a reference for this]. The frame rate they achieve is sufficient for VideoSAR capabilities.

The problem that arises with AFRL's approach is that it is not very practical. To achieve the frame rates necessary for VideoSAR capabilities requires a total of 2048 of the supercomputer's processors. It would not be feasible to outfit frontline deployed sensors with the same processing power afforded by such a physically large system. However, the rapid advancement of graphical processing units (GPU) over the past decade has afforded many scientific fields orders of magnitudes improvement in performance for a large variety of applications. This is just as true for the BP and FBP algorithm and it is exactly what we leverage in an effort to produce a portable solution for VideoSAR.

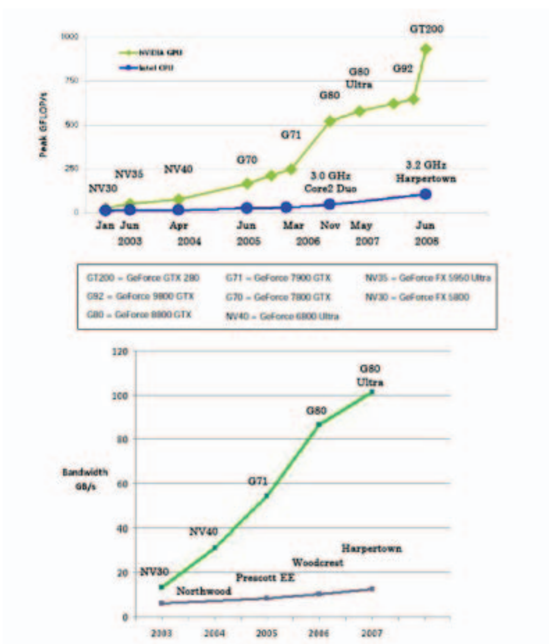
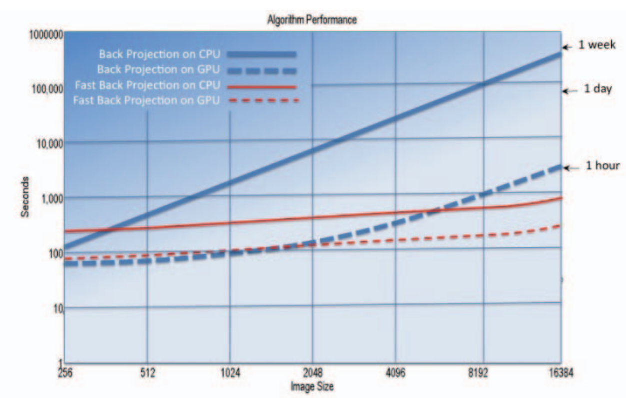


Figure 1: CPU vs. GPU benchmarks. (courtesy NVIDIA)

The past decade has seen the GPU performance further outpace CPU performance with each new release as illustrated in Figure 1. It is especially evident in 2006 when the general purpose GPU (gpGPU) began to mature. NVIDIA's Geforce 8800 was the first card to support the C programming language making it significantly easier for scientists to directly access the power of the GPU. Since that time, the development of the application programming interface (API) CUDA as well as the open computing language (OpenCL) has made the power of the GPU more readily available beyond just the gaming community. The last generation of NVIDIA cards, namely the GT285, contained 240 CUDA cores that could be used for parallel processing. The GT295 was a dual GPU card that housed essentially two GT285 chips on a single board. This provided an unprecedented 480 cores on a single PCI express card.

The next generation cards are shaping up to be even more impressive. NVIDIA's Fermi architecture will house 512 CUDA cores per GPU [5]. Although not mentioned as of yet, it is very likely that a dual GPU card will be released at some point in the product line's lifetime providing a total of 1024 CUDA cores on a single PCA express card. With motherboards such as Gigabyte's GA-8n-SLI or Asus's P5W64-WS that each support up to four PCI express x16 slots it would be possible to house as many as four next generation Fermi cards bringing the total number of CUDA cores in a single ATX tower to be 2048. That amount of parallelism in a small form factor box could provide the necessary bridge to obtaining real time performance for large volumes of SAR data outside of the realm of the laboratory.

Currently, we have parallelized the BP and FBP algorithms for running on NVIDIA's Tesla C1060 with 240 CUDA cores. With just this single card and a non-optimized algorithm we have measured substantial improvements in the performance of both the parallel BP and FBP algorithms compared to their serial counterparts as illustrated in Figure 2. By leveraging the hundreds of cores, we were able to obtain more than two orders of magnitude improvement in the performance of the BP algorithm. The FBP did not produce as compelling performance improvements solely due to the non-optimized algorithm that is currently being used. Significant changes that leverage the vast performance differences between the global and shared memory on the NVIDIA cards could provide substantial improvements in performance. GPU optimizations as well as additional algorithm adjustments indicate that the parallel FBP algorithm operating on a single card could possibly be more than an order of magnitude faster than the serial FBP algorithm. Bear in mind that this is all on current generation GPUs.



The next generation GPUs will offer more cores at faster clock rates which potentially can provide even further improvements to the algorithms performance. Currently, the largest bottleneck in the parallel FBP performance is the data shuffling between the CPU and GPU. This will also be mitigated some in the next generation cards that will house 50% more memory per GPU than is currently available. By distributing the processing across as many as 2048 cores located in a single tower, it will be possible to achieve more than two orders of magnitude improvement in performance compared to the same FBP algorithm running serially today. With this computational power available, there is no doubt that VideoSAR capabilities can be readily deployed across the globe in the near future.

In our talk, we will demonstrate the vast improvements that can be achieved using a parallel GPU algorithm over the more traditional CPU approach. By comparing image quality as well as processing time, the clear advantage of utilizing the GPU will be made apparent. We will also demonstrate the tremendous costs that can be saved when implementing the more compact GPU supercomputer approach over the traditional CPU supercomputer. The result will have threefold savings in the form of costs, space, and computational time.

Finally, we will present recent results of applying GPU technology to Interferometric SAR processing and discuss the challenges and opportunities in that area.

References:

[1] Ulander, L. M. H., Hellsten, H., and Stenstrom, G., "Synthetic-aperture radar processing using fast factorized backprojection" IEEE Transactions on Aerospace and Electronic Systems, 39(3), 760-776, (2003).

[2] Xiao, S. Munson, D. C., Jr., Basu, S., and Bresler, Y., "An N^2 log N backprojection algorithm for SAR image formation", 34th Asilomar Conference on Signals, Systems, and Computers, 3-7, (2000).

[3] McCorkle, J., and Rofheart, M., "An order N^2log(N) backprojector algorithm for focusing wide-angle wide-bandwidth arbitrary-motion synthetic aperture radar", Proc. SPIE 2747, 25-36, (1996).

[4] Ben Cordes and Miriam Leeser, "Parallel Backprojection: A Case Study in High-Performance Reconfigurable Computing," EURASIP Journal on Embedded Systems, vol. 2009, Article ID 727965, 14 pages, 2009.

[5] *NVIDIA's Next Generation CUDATM Compute Architecture: FERMI,* white paper, Nvidia Santa Clara, California, 2009, available at http://www.nvidia.com/content/PDF/fermi_white_papers/NVIDIA_Fermi_Compute_Architecture_Whitepaper.pdf.