2021 IEEE/ACM HPC for Urgent Decision Making (UrgentHPC) | 978-1-6654-1130-1/21/$31.00 ©2021 IEEE | DOI: 10.1109/UrgentHPC54802.2021.00008

Supporting Multi-messenger Astrophysics with Fast Gamma-ray Burst Localization

Jacob Wheelock
Dept. of Electrical and Systems Engineering
Washington Univ. in St. Louis
jacobwheelock@wustl.edu

William Kanu
Marion Sudvarg
Zhili Xiao
Jeremy D. Buhler
Roger D. Chamberlain
Dept. of Computer Science and Engineering
Washington Univ. in St. Louis
{wkanu,msudvarg,xiaozhili}@wustl.edu
{jbuhler,roger}@wustl.edu

James H. Buckley
Dept. of Physics
Washington Univ. in St. Louis
buckley@wustl.edu

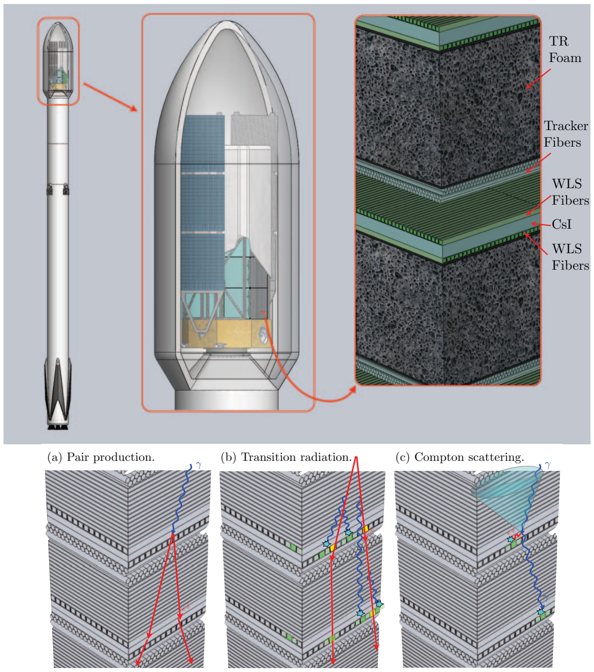
***Abstract*—Multi-messenger astrophysics is amongst the most promising approaches to astronomical observations. A significant challenge, however, is the fact that many instruments have a narrow field of view, so transient events are often missed by these instruments. The Advanced Particle-astrophysics Telescope, currently under development, promises to provide low-latency detection and localization for an important class of astronomical events, thereby enabling the full observational capabilities of narrow field-of-view instruments to be brought to bear. We examine the computational pipeline for detection and localization of Compton events utilizing computational accelerators, both FPGAs and GPUs.**

***Index Terms*—Advanced Particle-astrophysics Telescope (APT), Compton scattering, FPGA, GPU**

I. Introduction

The Advanced Particle-astrophysics Telescope (APT) is a future gamma-ray/cosmic-ray mission that will combine a pair tracker and Compton telescope in a single monolithic design [1]. One of the major features of APT is that by incorporating multiple Compton imaging over a very large effective area, the instrument will achieve orders of magnitude improvement in sensitivity to photons of energies of one to a few MeV compared to existing experiments. The multilayer design also makes it possible to achieve a much larger field of view (FoV) than conventional Compton telescopes. This feature is particularly beneficial in the newly emerging area of multi-wavelength and multi-messenger astrophysics, in which transient signals from multiple modalities are combined to learn more about the physical universe.

The information carried by photons, gravitational waves, and neutrinos about individual cosmic sources is inherently complementary [2]. In addition, many of the instruments used to acquire signals in these modalities are limited to a narrow field of view (e.g., sub-1° FoV is common), while large FoV instruments (such as gravity wave detectors) provide limited-precision localization of detected signals. APT is intended to support sub-1° localization precision in the MeV energy range.



(a) Pair production. (b) Transition radiation. (c) Compton scattering.

Fig. 1. Top: APT in Falcon-9 faring. Bottom: Detection modes. [1]

This combination of wide instantaneous field of view and narrow localization capability makes APT an important component in the detection of γ-ray transients such as neutron-star mergers. Fast detection and localization of these events – ideally within less than one second of their light arriving at the detector – is crucial to capture sufficient data about their evolution. The ability to urgently respond to events and re-task terrestrial instruments therefore dramatically increases the fraction of the available observation window of opportunity

This work supported by NSF awards CNS-1763503 and CNS-1814739 and NASA awards 80NSSC19K0625 and 80NSSC21K1741.

978-1-6654-1130-1/21/$31.00 ©2021 IEEE
DOI 10.1109/UrgentHPC54802.2021.00008

that is utilized, which increases the quality of the resulting science.

The full APT instrument will have a cross-sectional area of 3 m × 3 m and a height of 2.5 m, consisting of 20 xy tracker and imaging CsI calorimeter (ICC) layers (see Figure 1). Two such 3 m modules fit in the faring of a Falcon 9 rocket. With a Falcon-heavy rocket, it should be possible to lift the instrument into a sun-earth Lagrange orbit, where the obscuration of the sky by the earth is minimized and the benefit of the large (nearly 4π-steradian) FoV can be exploited.

NASA has recently funded a full suborbital mission, the Advanced Demonstration of the APT (ADAPT), targeting a long-duration flight on a 60 million-cubic-foot balloon flight from Antarctica in 2024-2025. ADAPT will validate the APT technical approach while potentially providing prompt degree-scale localization of several bright GRBs that occur during the long-duration Antarctic flight.

In this paper, we evaluate the benefits of using computational accelerators, specifically Field-Programmable Gate Arrays (FPGAs) and Graphics Processing Units (GPUs), in the computational pipeline of Compton event detection, reconstruction, and localization within APT [3]. These tasks are budgeted for no more than 1 s of latency in the original instrument plans; however, further latency improvement will translate directly into improved scientific observations. Here, we report estimated latency of 200 ms for the entire computational pipeline.

II. APT Mission and Compton Pipeline

A. Advanced Particle-astrophysics Telescope (APT)

The APT mission is intended to have a broad impact on astroparticle physics; including the following primary science drivers [1]:

1) probing weakly interacting massive particle (WIMP) dark matter across the entire natural mass range and annihilation cross section for a thermal WIMP;
2) providing a nearly all-sky instantaneous FoV, with prompt sub-degree localization and polarization measurements for gamma-ray transients such as neutron-star mergers; and
3) making measurements of rare ultra-heavy cosmic ray nuclei to distinguish between n-star merger and supernovae r-process synthesis of the heavy elements.

In this paper, we are primarily interested in the second driver listed above. In particular, the goal is to detect and localize short-term gamma-ray bursts (GRBs) so that the source direction can quickly be communicated to other instruments, specifically those with a limited field of view.

Multi-messenger astrophysics seeks to learn about gravitational collapse of end-of-life massive stars, neutron stars and neutrino emission, supernova and supernova remnants, pulsars and pulsar wind nebulae, stellar mass black holes, and other physical phenomena [4]. The core idea is that different observational modalities teach us different things about each of these phenomena.

The range of observational modalities is quite large, including optical telescopes, radio telescopes, X-ray telescopes, γ-ray telescopes, neutrino telescopes, and gravity wave detectors [4]. While some of these instruments have a wide FoV (e.g., the gravity wave detectors), many do not. For example, optical telescopes with a wide aperture often have a very limited FoV (typically $< 1°$) [5].

Huerta et al. [6] recently published a set of recommendations for maximizing the potential for scientific discovery that includes substantial computational requirements, particularly for processing of the resultant data sets in real time [7]. The sooner we can localize the source of a GRB, the faster we can target other instruments to the same source. It is this rapid targeting capability that we are working to realize with APT.

B. Compton Pipeline

Figure 2 illustrates the detection mechanism in the APT instrument [1]. Incident photons strike thin (∼5 mm) CsI:Na crystal tiles, generating scintillations from Compton scatterings that deposit some of the photon's energy. Wavelength shifting (WLS) fibers (2 mm square) covering the tiles are used to collect and shift the UV/blue re-emission of deposited energy from the CsI:Na and pipe a fraction of the isotropically re-emitted light to silicon photomultipliers (SiPMs) on the periphery.

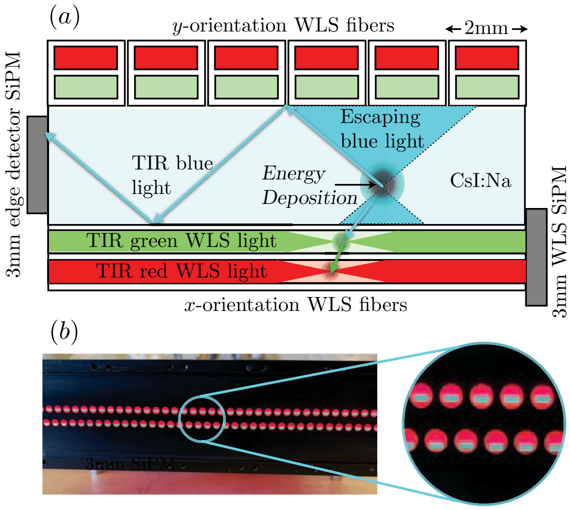


Fig. 2. (a) Imaging CsI detector principle of operation. (b) End-on view of fibers. TIR is totally internally reflected light. From [1]

Centroiding the light collected by the orthogonal WLS fibers bonded to either side of the CsI crystals provides the xy coordinates of the interaction. Which layer detects the energy deposition provides the z coordinate. The instrument will use analog-pipeline waveform digitizers [8], which enable only triggered events to be digitized. This dramatically decreases overall energy requirements, as the instrument is not digitizing unused noise.

In the Compton regime, the computational pipeline that follows digitization is comprised of event detection (determining the nominal xyz coordinates of each energy deposition), reconstruction (determining the sequence of energy depositions by each photon), and localization (estimating the origin direction from a burst of photons). Each of these is described in turn in the sections below.

III. Event Detection using an FPGA

When the analog waveform digitizer ASICs trigger an event, the information that is passed into the event detection stage is the collection of contiguous fibers that have a signal, with the ith fiber represented by f_i, and the intensity signal on each of these fibers, represented by s_i. This happens in both the x and y dimensions. In this stage of the computation, we must determine the centroid in each dimension of each energy deposition. Adapting the techniques from Hyde [9], here we investigate the use of high-level synthesis (HLS) to perform the centroid computation, which is shown in Figure 3 and expressed in Equation (1).

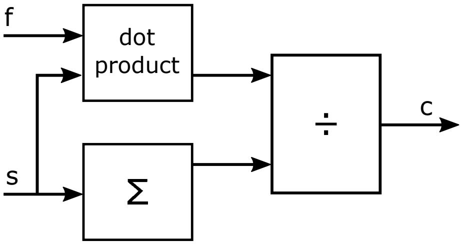


Fig. 3. Centroiding computation.

$$c = \frac{\sum_i s_i \cdot f_i}{\sum_i s_i} \tag{1}$$

Here, c is in units of fibers, but that is easily converted to mm as the width for each fiber is 2 mm (see Figure 2).

Given that an FPGA will be used to interface with the analog ASICs, we are interested in performing the centroiding computation on the same FPGA. Flight hardware has not yet been chosen, so we utilize a desktop card for evaluation purposes (a Xilinx Alveo U250 accelerator card, which includes an UltraScale+ architecture FPGA). The number of contiguous fibers with a signal for each energy deposition can vary. In this work, we explore a maximum of 24 contiguous fibers, guided by the simulation models of Chen et al. [10].

The area limitations are completely dominated by DSP blocks (see Table I). For 12 instances of a 24-fiber version of Figure 3, 864 DSP blocks are required, comprising 7% of the FPGA. This corresponds to 3 DSP blocks per fiber. All other resources are underutilized by comparison.

TABLE I
FPGA Resource Usage

Instances	LUTs	BRAM	Registers	DSPs
12	25,529 (1.5%)	23 (0.9%)	67,528 (2.3%)	864 (7.0%)

The 12-instance design can be clocked at 300 MHz and completes a single event computation in 68 cycles, or 0.23 μs. This design therefore supports an event throughput of 53 Mevents/s, well above the anticipated photon arrival rate.

Figure 4 shows a scatter plot of the positional error (separately for x and y) that results from the above technique. Incident γ-ray photon interactions and ground-truth positions are provided by [10]. The large majority of the errors are less than 2 mm, the width of a single WLS fiber.

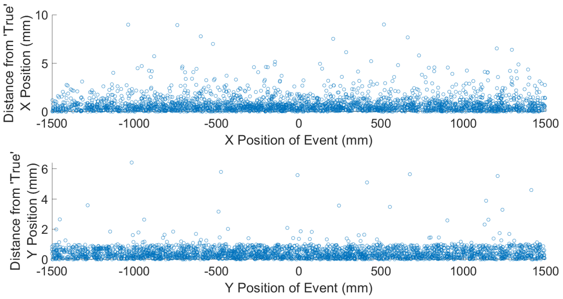


Fig. 4. Positional error as a function of position.

When used in a different context (a microscope design), Hyde [9] reported significant positioning errors near the boundaries of the sensor. The geometry of our instrument is sufficiently different that, at least qualitatively, we do not see this issue in our application. Our current investigation is focused on the few outliers with large positioning errors — both understanding their cause and mitigating their downstream impact.

IV. Reconstruction on a Multicore Processor

Reconstruction takes the Compton scattering events detected from one incident γ-ray photon by the centroiding stage, including both their locations and their deposited energies, and infers the order in which they occurred in the detector. The goal of this inference is to discover the locations of the most likely first and second scatterings in order, which are used along with the first scattering's deposited energy to constrain the direction in the sky from which the photon arrived at the detector. The time between scatterings is too small to directly observe their sequence; instead, we consider every possible ordering, each implying a trajectory of the photon in the instrument, and select the trajectory that best explains the observed locations and energy depositions.

Our basic approach to trajectory reconstruction from Compton scatterings follows Boggs and Jean [11], who described a figure of merit for a putative trajectory based on agreement between the angles at which the photon scatters (given the trajectory) and the corresponding energy deposits, which provide independent estimates of each scattering angle via the Compton law. In [3], we describe an algorithmic approach to accelerate the testing of all possible trajectories over a set of scattering events. To eliminate redundant computation and ensure rapid analysis even of photons with multiple scatterings, we implement a tree search over possible photon trajectories, which merges the computations for trajectories sharing a common ordered prefix of scatterings as shown in Figure 5, to find one with the highest figure of merit. Further acceleration is obtained by pruning the search tree to eliminate trajectories that cannot or are unlikely to have the highest merit.

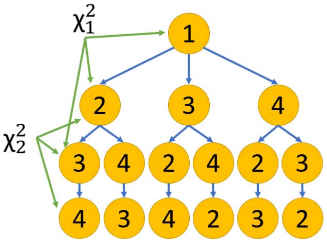


Fig. 5. A subtree for an event with $N = 4$ scatterings, representing all permutations that begin with scattering 1. The χ^2 terms are components of the figure of merit, each computed from three successive scatterings.

We implemented reconstruction on a four-core ARM Cortex-A53 processor, specifically a Raspberry Pi 3B+, to mimic the kind of low-power embedded processor that meets the power requirements of APT and ADAPT. We simulated γ-ray burst events resulting in thousands to hundreds of thousands of incident photons input to reconstruction and found that reconstruction of all photons for such events typically took 10 ms or less, as shown in Figure 6. Our high observed efficiency arises partly from algorithmic improvements as described above and partly from accelerating the computation using four cores to process different photons in parallel.

V. Localization using a GPU

Localization of a γ-ray burst combines information from multiple reconstructed photons to infer the most likely direction in the sky from which the burst came. From each reconstructed photon i, we obtain a unit vector c_i parallel to a line through the locations of its first two scatterings in the detector, as well as an inferred angle ϕ_i between c_i and another unit vector s representing the burst's source direction in the sky. The pair (c_i, ϕ_i) thus defines a circle on the unit sphere centered on the detector, with the source s assumed to lie somewhere on this circle. In practice, measurement uncertainty leads to an uncertainty σ_i^2 in the angle ϕ_i, so that the circle is in fact an *annulus* with a Gaussian cross section. Given a collection of hundreds to thousands of such annuli, the goal is to infer a single most plausible source direction s that gave rise to them. The task is complicated by the fact that some photon trajectories (perhaps even a majority) are reconstructed incorrectly. Hence, s must be inferred robustly from a plurality of annuli, with the remainder discounted as noise.

As described in [3], localization divides naturally into two stages. First, a large number of candidate directions are tested, and the best candidates are then combined to produce a rough approximation to the correct source direction. Second, this initial approximation is iteratively refined using a linear least-squares approach. The final source direction is typically accurate to within a few degrees of arc, though sub-degree accuracy is desired in practice and is routinely achieved for bursts that produce sufficiently many photons.

While both reconstruction and localization can be implemented on a multicore, we found that these two parts of our computation exhibited very different performance characteristics. As Figure 6 illustrates, when run on our four-core ARM Cortex-A53 processor, the time spent in reconstruction is almost negligible compared to that spent in localization. Hence, we sought to further accelerate localization using wide-SIMD parallelism – specifically, targeting a graphics processor (GPU). We note that low-power devices exist (e.g., the NVIDIA Jetson family) that combine a GPU with an ARM multicore, so targeting different parts of the burst detection to heterogeneous resources is eminently practical for our application. In what follows, we describe our parallelization strategy and estimate the performance of the resulting computation on a representative low-power GPU device.

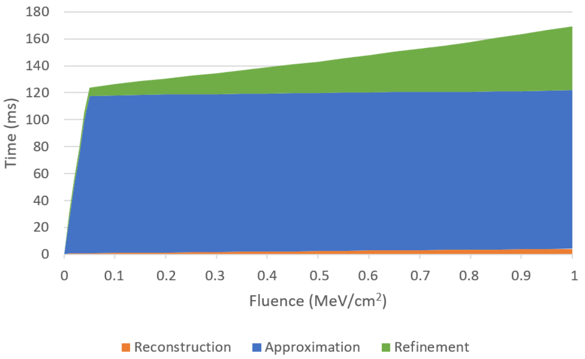


Fig. 6. Pipeline stage execution times on a four-core ARM Cortex-A53 CPU for a range of input sizes. Fluence is a measure of energy density in the sample and is proportional to input size; for our instrument, a fluence of 0.03 MeV/cm^2 corresponds to 5000 incident photons. [3]

A. Approximation

In order to derive meaning from the potentially large number of annuli produced from reconstruction, we seek a candidate source direction where multiple photons' annuli overlap. We first sample a small number of annuli (c_j, ϕ_j) from the input and, for each annulus, construct a set of equally-spaced directions s_{jk} on the circle of radius ϕ_j around center c_j, as illustrated in Figure 7. We hypothesize that sampling sufficiently many input photons results in a high probability of picking at least one correctly reconstructed photon; for this photon, some direction on its implied circle should be close to the true source direction.

We evaluate each candidate source direction s_{jk} using an estimated likelihood of all photons' annuli given this direction.

Specifically, we compute

$$L(s_{jk} \mid (c_i, \phi_i)) \propto \frac{1}{\sigma_i} e^{-(\phi_i - \beta_i)^2 / 2\sigma_i^2},$$

where $\beta_i = \arccos(s_{jk} \cdot c_i)$. The overall likelihood of s_{jk} is simply the product of its likelihoods given each input annulus. This product is computed in the log domain for efficiency, with an additional "short-circuiting" optimization to rapidly discard as noise annuli that do not pass anywhere near s_{jk}. For each sampled annulus (c_j, ϕ_j), we retain the single most likely source direction s_j from among the tested s_{jk}, then average these directions, weighted according to their likelihoods, to obtain the approximate source direction s.

Approximation involves a large number of likelihood computations, all of which can be performed independently. In practice, we sample 20 input annuli and test 720 directions s_{jk} per annulus. For each annulus, we must then perform an argmax reduction over the likelihoods for all its tested directions to find the most likely candidate, before finally computing a weighted average of these candidates. A natural SIMD decomposition of this work is to divide the sampled annuli evenly across GPU blocks (i.e., across processors of the GPU), and within each block to divide individual directions s_{jk} evenly among threads. For one annulus, each thread sequentially computes likelihoods for its assigned directions, maintaining the most likely direction found. Finally, all SIMD threads in a block collaboratively compute the argmax reduction for the annulus. The candidates s_j for each sampled annulus are returned to the host processor, where the weighted average is performed sequentially.

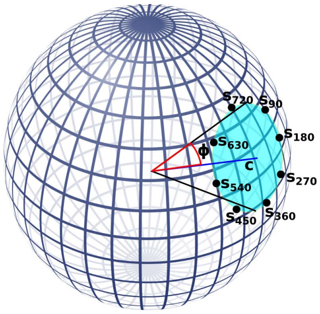


Fig. 7. For initial approximation, candidate source directions are evenly spaced around a randomly selected annulus.

In contrast to our ARM multicore implementation, whose parallelism is limited to the number of available cores (less than ten), the GPU allows computation of many tens of likelihoods in parallel per processor core, with additional performance obtained from hiding the latency of the data accesses associated with each likelihood computation.

B. *Refinement*

After obtaining our initial guess at the source direction s, we seek to refine it. In principle, given three circles (c_j, ϕ_j) that meet at a common point, it is possible to *trilaterate*, that is, to infer a source direction by solving for the unit vector s that satisfies

$$s \cdot c_j = \cos \phi_j$$

for all j. In practice, however, we are given many more than three annuli, each with some uncertainty, so the problem is both overdetermined and impossible to solve exactly. We therefore seek a unit vector s that solves this system in a least-squares sense, with the constraint for annulus (c_j, ϕ_j) weighted by $1/\sigma_j$. The unit-vector constraint makes the problem nonlinear, but this particular nonlinear constraint can be accommodated efficiently by reducing the problem to a quadratic eigenvalue problem [12], [13].

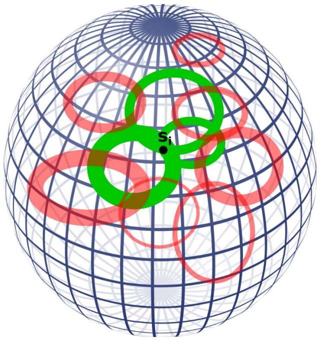


Fig. 8. At refinement step i, only annuli (shown in green) within three standard deviations of the current estimated source vector $\mathbf{s}_i$ are used as constraints for the least-squares problem. Other annuli (shown in red) are ignored.

The iterative nature of the computation arises from the desire for robust estimation of s in the presence of noise. Given an initial candidate s from the approximation stage, we discard all annuli for which s lies at least three standard deviations from their circles, i.e., for which $|\arccos(s \cdot c_j) - \phi_j| > 3\sigma_j$, as illustrated in Figure 8. We solve the resulting reduced least-squares problem as described, producing a new source estimate s', then again discard annuli passing too far from s' and compute a new least-squares estimate. This process is iterated 20 times.

The dominant costs of each iteration of refinement are, first, the selection of which annuli to retain, and second, reduction of the resulting least-squares problem to a quadratic eigenvalue problem. If the first step retains N annuli, then the second involves forming an $N \times 3$ matrix A and an N-vector b from the centers c_j and angle cosines $\cos \phi_j$ from all retained annuli, respectively, and then computing $A^T A$ and $A^T b$. The resulting eigenvalue problem has small constant size. We first parallelize selection of annuli across as many threads as possible, distributed over enough blocks to occupy all GPU processors. We then perform a device-wide parallel scan and compaction to form A and b and finally parallelize the multiplications needed to form the eigenvalue problem, which we solve on the ARM using the Eigen library [14].

We relied on CUDA's CUB library for scans and reductions but otherwise coded all GPU portions of the computation explicitly in CUDA. We attempted to use the cuBLAS library for the linear-algebraic computations but found that it was substantially slower than our own naïve multiplication kernel for the problem sizes tested (N of several thousand).

C. Performance

We estimate the performance of our GPU implementation of localization on an NVIDIA Jetson Xavier NX board, a 10-watt device whose GPU has six processors (SMs) built on the Volta architecture. Because we do not have access to a Xavier NX device, we extrapolate from performance measurements taken on an NVIDIA RTX 2080, which has 46 processors built on the newer Turing architecture. We conservatively estimate that, due to differences in clock rate and architecture, one processor of the RTX 2080 may run up to twice as fast as one processor of the Xavier NX, implying that, if (as is the case) the GPU time dominates the overall computation time, the Xavier will run roughly $46/6 \times 2 \approx 15.33$ times slower than the RTX 2080.

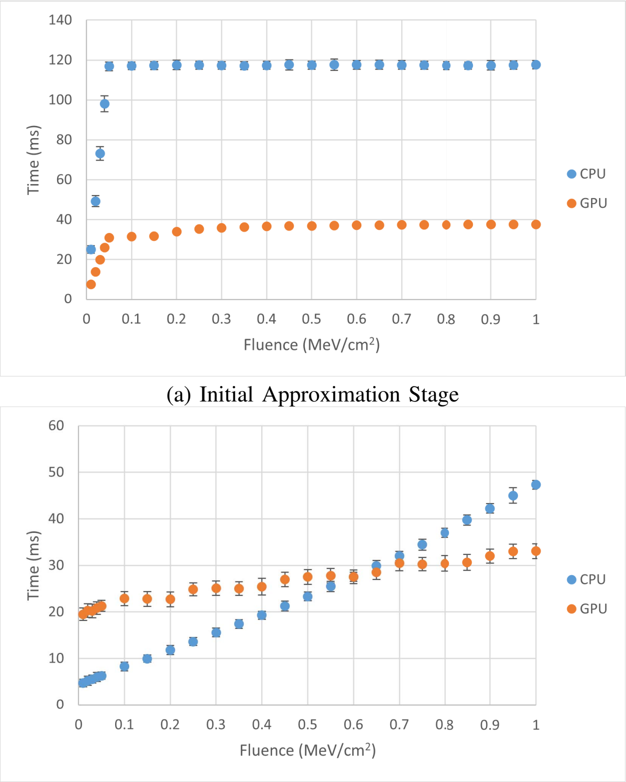
We tested our code on a dataset of 10^6 γ-ray photons from a long GRB simulated in [15], with additional instrument noise modeled with APTSoft [10]. We measured execution times for both localization stages across a range of input sizes. For each input size in our domain, we randomly sampled 200 subsets of photons, performed reconstruction, then measured execution time for both the the initial source approximation and the iterative refinement stages of our pipeline.

Figure 9 illustrates the estimated times for approximation and refinement on the Xavier NX's GPU compared to the measured times on our four-core ARM Cortex-A53. For the initial approximation stage, we observe significant speedup with GPU acceleration. For the iterative refinement stage, our estimated times on the GPU demonstrate significant constant-time overhead, but execution times increase more slowly with fluence on the GPU compared to the CPU. Since running time is dominated by the CPU-bound eigenvalue problem and memory copies between the CPU and GPU (which are less expensive on a shared-memory platform like the Xavier NX), we expect the constant-time overhead to be closer to that of the CPU implementation. Overall, we expect a $2.5 - 3.5\times$ speedup by moving localization to the GPU.

VI. Conclusions and Future Work

For multi-messenger astrophysics, directing terrestrial telescopes with a narrow field of view onto transient celestial events is an urgent task. For many such events (e.g., neutron-star mergers), γ-ray burst detection in the Compton regime is a promising approach to provide early detection and localization of these events, and the Advanced Particle-astrophysics Telescope is being designed for this very purpose.

Utilizing FPGAs for event detection, low-power processor cores for reconstruction, and GPUs for localization, our initial investigations indicate that the entire computational pipeline can be performed with an estimated latency of under 200 ms, well under the 1 s budget postulated in the initial telescope proposals.



(a) Initial Approximation Stage

(b) Iterative Refinement Stage

Fig. 9. Comparison of execution times between the ARM Cortex-A53 CPU and the NVIDIA Jetson Xavier NX board (estimated) for a range of input sizes. Error bars denote the standard deviation over 200 trials.

There is substantial additional work to be done before we can confidently launch even the prototype instrument on our upcoming balloon flight. In the event detection stage, we need to interface the FPGA logic with the analog-pipeline waveform digitizer ASICs and assess the importance of the outliers that are clearly present in Figure 4. Between event detection and reconstruction, we need to insert a compute stage that distinguishes Compton events from pair-production events that occur at higher energies. Algorithmic approaches to perform this task are currently under investigation. Within the reconstruction stage, we are investigating whether a maximum-likelihood formulation would give better results than the current algorithm due to Boggs and Jean [11]. For the localization stage, we need to measure the performance on flight-capable hardware (this is true for event detection as well).

Throughout the entire pipeline, we need to further investigate both the sources of noise in the instrument and assess mechanisms for mitigating that noise. This includes stray γ-rays that are not part of the burst, other sources that can trigger stray scintillations in the CsI layers or the fibers, dark photon counts in the SiPMs, and electronics noise in the analog subsystem. While early assessments are quite positive, significant work remains before the instrument flies.

References

[1] J. H. Buckley, S. Alnussirat, C. Altomare, R. G. Bose, D. L. Braun, J. Buhler, E. Burns, R. D. Chamberlain, W. Chen *et al.*, “The Advanced Particle-astrophysics Telescope (APT) Project Status,” in *Proc. of 37th Int’l Cosmic Ray Conference — PoS(ICRC2021)*, vol. 395, 2021, pp. 655:1–665:9.

[2] P. Mészáros, D. B. Fox, C. Hanna, and K. Murase, “Multi-messenger astrophysics,” *Nature Reviews Physics*, vol. 1, no. 10, pp. 585–599, 2019.

[3] M. Sudvarg, J. Buhler, J. H. Buckley, W. Chen, Z. Hughes, E. Ramey, M. L. Cherry, S. Alnussirat, R. Larm, C. Chungata *et al.*, “A Fast GRB Source Localization Pipeline for the Advanced Particle-astrophysics Telescope,” in *Proc. of 37th Int’l Cosmic Ray Conference — PoS(ICRC2021)*, vol. 395, 2021, pp. 588:1–588:9.

[4] A. Neronov, “Introduction to multi-messenger astronomy,” in *Journal of Physics: Conference Series*, vol. 1263, no. 1. IOP Publishing, 2019, p. 012001.

[5] M. R. Ackermann, J. T. McGraw, and P. C. Zimmer, “An overview of wide-field-of-view optical designs for survey telescopes,” in *Proc. of Advanced Maui Optical and Space Surveillance Technologies Conference*, 2010.

[6] E. A. Huerta, G. Allen, I. Andreoni, J. M. Antelis, E. Bachelet, G. B. Berriman, F. B. Bianco, R. Biswas, M. C. Kind, K. Chard *et al.*, “Enabling real-time multi-messenger astrophysics discoveries with deep learning,” *Nature Reviews Physics*, vol. 1, no. 10, pp. 600–608, 2019.

[7] D. George and E. Huerta, “Deep neural networks to enable real-time multimessenger astrophysics,” *Physical Review D*, vol. 97, no. 4, p. 044039, 2018.

[8] K. Bechtol, S. Funk, A. Okumura, L. Ruckman, A. Simons, H. Tajima, J. Vandenbroucke, and G. Varner, “TARGET: A multi-channel digitizer chip for very-high-energy gamma-ray telescopes,” *Astroparticle Physics*, vol. 36, no. 1, pp. 156–165, 2012.

[9] J. Hyde, “Data processing electronics for an ultra-fast single-photon counting camera,” Master’s thesis, Washington University Dept. of Electrical and Systems Engineering, St. Louis, MO, Aug. 2020. [Online]. Available: https://doi.org/10.7936/nk7z-0p22

[10] W. Chen, J. Buckley, S. Alnussirat, C. Altomare, R. Bose *et al.*, “The Advanced Particle-astrophysics Telescope: Simulation of the Instrument Performance for Gamma-Ray Detection,” in *Proc. of 37th Int’l Cosmic Ray Conference — PoS(ICRC2021)*, vol. 395, 2021, pp. 590:1–590:9.

[11] S. Boggs and P. Jean, “Event reconstruction in high resolution Compton telescopes,” *Astronomy and Astrophys. Supp. Series*, vol. 145, no. 2, pp. 311–321, 2000.

[12] W. Gander, G. H. Golub, and U. V. Matt, “A constrained eigenvalue problem,” *Linear Algebra Appl.*, vol. 114-115, pp. 815–839, 1989.

[13] F. Tisseur and K. Meergergen, “The quadratic eigenvalue problem,” *SIAM Rev.*, vol. 43, pp. 235–286, 2001.

[14] G. Guennebaud, B. Jacob *et al.*, “Eigen v3,” 2010. [Online]. Available: http://eigen.tuxfamily.org

[15] S. Agostinelli, J. Allison, K. Amako, J. Apostolakis *et al.*, “Geant4 — a simulation toolkit,” *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 506, no. 3, pp. 250–303, 2003.

Appendix: Artifact Description

Summary of the Experiments Reported

The localization algorithms were run on a Washington University server that has a Intel(R) Xeon(R) CPU E5-2620 v4 for the CPU and an NVIDIA RTX 2080 for the GPU. CUDA 11.4, Eigen Version 3.3.9 and gcc (GCC) 9.2.1 20191120 (Red Hat 9.2.1-2) were used on this machine for testing.

Artifact Availability

Software Artifact Availability

All author-created software artifacts are maintained in a public repository under an OSI-approved license. The URL is below.

Hardware Artifact Availability

There are no author-created hardware artifacts.

Data Artifact Availability

The data artifacts are not yet publicly available.

Proprietary Artifacts

None of the associated artifacts, author-created or otherwise, are proprietary.

List of URLs and/or DOIs where artifacts are available

https://sbs.wustl.edu/ADAPTsoftware.html

Baseline Experimental Setup and Modifications Made for the Paper

Relevant hardware details

Server - see details below.

Operating Systems and Versions

Rocky Linux 8.4 (Green Obsidian)

Compilers and Versions

gcc (GCC) 9.2.1 20191120 (Red Hat 9.2.1-2), nvcc 11.0

Libraries and Versions

CUDA 11.4, Eigen 3.3.9

Key Algorithms

GEMM, Linear Least-Squares

Output from scripts that gather execution environment information

```
+ uname -a
Linux lotus.engr.wustl.edu 5.10.16-1.el8.
    elrepo.x86_64 #1 SMP Thu Feb 11
    17:44:06 EST 2021 x86_64 x86_64 x86_64
    GNU/Linux
```

```
+ lscpu
Architecture:          x86_64
CPU op-mode(s):        32-bit, 64-bit
Byte Order:            Little Endian
CPU(s):                32
On-line CPU(s) list:   0-31
Thread(s) per core:    2
Core(s) per socket:    8
Socket(s):             2
NUMA node(s):          2
Vendor ID:             GenuineIntel
CPU family:            6
Model:                 79
Model name:            Intel(R) Xeon(R) CPU
    E5-2620 v4 @ 2.10GHz
Stepping:              1
CPU MHz:               1494.107
CPU max MHz:           3000.0000
CPU min MHz:           1200.0000
BogoMIPS:              4199.94
Virtualization:        VT-x
L1d cache:             32K
L1i cache:             32K
L2 cache:              256K
L3 cache:              20480K
NUMA node0 CPU(s):     0-7,16-23
NUMA node1 CPU(s):     8-15,24-31
Flags:                 fpu vme de pse tsc
    msr pae mce cx8 apic sep mtrr pge mca
    cmov pat pse36 clflush dts acpi mmx
    fxsr sse sse2 ss ht tm pbe syscall nx
    pdpe1gb rdtscp lm constant_tsc
    arch_perfmon pebs bts rep_good nopl
    xtopology nonstop_tsc cpuid aperfmperf
     pni pclmulqdq dtes64 monitor ds_cpl
    vmx smx est tm2 ssse3 sdbg fma cx16
    xtpr pdcm pcid dca sse4_1 sse4_2
    x2apic movbe popcnt tsc_deadline_timer
     aes xsave avx f16c rdrand lahf_lm abm
     3dnowprefetch cpuid_fault epb cat_l3
    cdp_l3 invpcid_single pti intel_ppin
    ssbd ibrs ibpb stibp tpr_shadow vnmi
    flexpriority ept vpid ept_ad fsgsbase
    tsc_adjust bmi1 hle avx2 smep bmi2
    erms invpcid rtm cqm rdt_a rdseed adx
    smap intel_pt xsaveopt cqm_llc
    cqm_occup_llc cqm_mbm_total
    cqm_mbm_local dtherm ida arat pln pts
    flush_l1d
```