Preprint typeset in JINST style - HYPER VERSION

GPU Enhancement of the Trigger to Extend Physics Reach at the LHC

V. Halyo,[*] P. Jindal, P. LeGresley, P. Lujan

Princeton University, Princeton, NJ, USA
E-mail: vhalyo@gmail.com

ABSTRACT: Significant new challenges are continuously confronting the High Energy Physics (HEP) experiments, in particular the Large Hadron Collider (LHC) at CERN, where nominal conditions deliver proton-proton collisions to the detectors at a rate of 40 MHz. This rate must be significantly reduced to comply with both the performance limitations of the mass storage hardware and the capabilities of the computing resources to process the collected data in a timely fashion for physics analysis. At the same time the physics signals of interest must be retained with high efficiency.

The quest for rare new physics phenomena at the LHC and the flexibility of the trigger system leads us to evaluate a Graphics Processing Unit (GPU) enhancement of the compute farm to not only provide faster and more efficient events selection but also include new complex triggers that were not possible before. A new tracking algorithm is evaluated on a NVIDIA Tesla K20c GPU, allowing for the first time the reconstruction of long lived particles at the tracker system in the trigger. Preliminary time performance and efficiency will be presented.

KEYWORDS: ATLAS; CMS; Level-1 trigger; HLT; Tracker system; .

[*]Corresponding Author

Contents

1. Introduction

The stunning performance of the LHC, the successful physics results, and the discovery of a new Higgs like particle will mark in history the first LHC running period not as the beginning of the end but rather the end of the beginning. The first LHC shut down, scheduled for 2 years, provides a remarkable opportunity to improve our detector performance even further. In particular to enhance and extend the physics reach that would be selected and recorded by the trigger .

The Trigger and Data Acquisition (DAQ) systems [**?**] in a modern collider experiment provide essential preliminary online analysis of the raw data for the purpose of filtering potentially interesting events into the data storage system. At the design luminosity of 10^{34} cm^{-2}s^{-1} this is a formidable task which requires real time handling of raw data with beam crossings at 40 MHz, each yielding on average $\sim$20 inelastic proton-proton events, and producing approximately 1 MB of zero-suppressed data. To keep the overall data rate within the current capability of the archival storage system of $\sim$100 MB$/s$, the trigger system must achieve a rejection ratio for background events of at least $\sim$400,000 : 1.

The required level of performance can be achieved by implementing a typical trigger system with a hierarchy of multiple levels, ranging from fast and relatively simple criteria implemented entirely in hardware and firmware, to more sophisticated software based analysis. The CMS trigger is implemented in 2 levels: a hardware Level-1 trigger reducing the event rate to less than 100 kHz, followed by further processing in the High-Level Trigger (HLT) designed to reduce this maximum Level-1 acceptance rate of 100 kHz to a final output rate of 100 Hz. A significant component of the HLT is to apply a specific set of physics selection algorithms on the events read out to accept the events with the most interesting physics content. This compute intensive event selection is executed on a farm of commercial CPU processors that constitute the CMS HLT hardware.

By its very nature of being a computing system the HLT relies on technologies that have evolved extremely rapidly. For many decades one of the important methods for improving the performance of computing devices has been to increase the clock speed of the processor, and the typical CPU clock speed has increased by nearly a factor of 1000. In the past decade, however, it has been observed that it is no longer possible to rely solely on increases in processor clock speed as a means for extracting additional computational power from existing architectures. The underlying reasons are complex, but center around reaching what may be fundamental limitations in semiconductor device physics. For this reason, recent innovations have focused around *parallel* processing, either through systems containing multiple processors, or processors containing multiple cores, etc.

One interesting technology which has continued to see exponential growth is graphics processing. A modern GPU is a massively parallel processor with thousands of executions units to handle highly parallel workloads related to computer graphics. By making these execution units highly programmable, manufacturers have made the massive computational power of a modern GPU available for more general purpose computing, as opposed to being hard wired for specific graphical operations. In certain applications that can be executed in massively parallel fashion, this can yield several orders of magnitude better performance than a conventional CPU.

Furthermore, leading technology vendors have released supported development APIs, such as NVIDIA CUDA ("Compute Unified Device Architecture") [**?**], which allow rapid development of software in nearly standard C code. CUDA allows CPU-based applications to access the resources of a GPU for more general purpose computing without the limitations of using a graphics API. A modern GPU can simultaneously execute many thousands of computations in parallel, and an application using effective collaboration between the CPU and the GPU can see dramatic acceleration in algorithm execution.

A particularly good application for this technology would be executing machine vision and pattern recognition algorithms on detector data. These algorithms are often parallel by their very nature, and the highly controlled data produced by a particle physics detector reduces the pattern recognition task to it's purest form. From the physics perspective, such an enhancement of the trigger capabilities would allow inclusion of new tracking triggers, for example selection of events with multiple displaced vertices at any location in the silicon tracker, which could be a smoking gun for new topological signatures not predicted by the Standard Model. The proposed tracking upgrade will serve to augment the existing trigger with new types of trigger filters to select events with topological signatures that are currently suppressed.

In the following, a description of the physics motivation behind GPU enhancement of the HLT is provided. This includes a description of the existing High Level Trigger, the GPU and CUDA architecture, and the additional capabilities which would be enabled by a hybrid GPU/CPU compute farm for the HLT. The authors propose a new tracking reconstruction algorithm, show preliminary results, and a discussion of proposed new triggers that could trigger on new physics that would be a smoking gun for new physics at the LHC.

2. Physics Motivation

Enhancement of the HLT might permit processing of the tracking reconstruction on the full event at Level-1 event rate up to design luminosity. The proposed HLT tracking would have a far reaching impact on the physics program set by the HLT where the ultimate goal is to select the events of interest for CMS. The new tracking algorithm will introduce new possible trigger paths that are currently not possible due to the extensive processing time that would be required using CPUs alone. The robust parallel processing of the tracking on the hybrid CPU/GPU system will allow reconstruction of not only charged prompt tracks from the interaction point but also reconstruction of displaced vertices in the tracker far beyond what is possible in both CMS/ATLAS. Therefore the tracking algorithm using the GPU will enrich the physics program and allow for the search of new topological signatures that were not possible or suppressed before.

Only a few of the large selection of topological models can be described here. An example of an interesting class of models which would benefit from such triggers areâ hidden-valley models [4], in which a new confining gauge group is added to the standard model. The resulting (electrically-neutral) bound states can have low masses and long lifetimes and could be observed at the LHC. The production multiplicities are often large and events with final states with heavy flavor are common. In addition, displaced vertices and missing energy are possible. Accounting for LEP constraints, LHC production cross-sections were estimated to be typically in the 1-100 fb range, though they can be larger.

New tracking triggers would permit selection on the HLT Higgs like decays with a substantial branching fraction to long-lived neutral particles that may decay at macroscopic distances from the primary vertex [4] in the tracker. The limited experimental constraints on light neutral long-lived particles argues that we probably should not be surprised if a Higgs like reveals itself through such decays. The lifetimes of these resonances are not constrained; decays at centimeter and meter scales are equally possible. Each Higgs decay may produce two or more resonances with the multiplicity possibly varying from event to event. In many models the resonances will decay to the heaviest fermion pair available, with branching fractions similar to those of the standard model Higgs.

There are various models that predict these unusual signatures [4]-[**?**]. It includes either simple models where a scalar was added to the Higgs potential or superpotential depending whether a nonsupersymmetric or supersymmetric framework was used to build the model. Or other models that include coniňĄning hidden valley models which showed qualitatively similar signals though the origin of the signals is quite different. A complex Higgs decay in a hidden-valley model could produce, say, four resonances: one decaying promptly to jets, one escaping the detector giving missing energy, and two decaying to bb, each with a displaced vertex.

Other obvious channels that would benefit from the new tracking algorithm or displaced jets or vertex triggers are inclusive and exclusive b decay channels or various topologies with boosted jets that will be more frequent as we increase our center of mass of the collision at design luminosity. These new triggers will allow us to be sensitive to a larger parameter space including lower mass Higgs that could have evaded detection in previous experiments.

3. Trigger System

The LHC delivers proton-proton collisions to the CMS [?] detector at a rate of 40 MHz. This rate must be significantly reduced to comply with the performance limitations of the mass storage hardware and the ability of the offline computing resources to process the collected data in a timely fashion for physics analysis. At the same time the physics signals of interest must be retained with high efficiency. CMS features a two-level trigger system to reduce the rate to approximately 100 Hz. The Level-1 trigger [?] is based on custom hardware and designed to reduce the rate to about 100 kHz, corresponding to 100 GB/s, assuming an average event size of 1 MB. The High Level Trigger (HLT) [?],[?] is purely software-based and must achieve the remaining rate reduction by executing sophisticated offline-quality algorithms. HLT algorithm sequences consist of algorithms that are executed in order of increasing complexity and the execution of a path is stopped unless evidence for the signal of interest is found. This optimization means more sophisticated and time consuming reconstruction algorithms are seldom applied. The ability to select the events of interest is the foundation of our quest for rare new physics phenomena, and thanks to the flexibility of the CMS HLT software and hardware computing farm the PI is able to propose an enhancement for the HLT a few years down the road after startup.

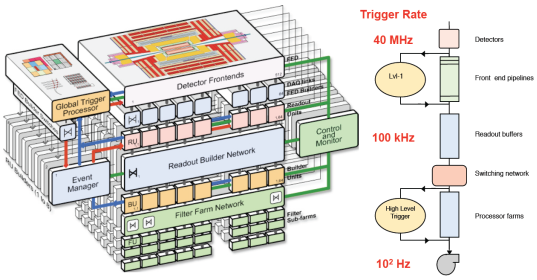


Figure 1. CMS DAQ Architecture. The size of the event builder (72 Readout Units, 288 Builder Units) represents one âĂIJsliceâĂİ; the system can be equipped with up to eight slices.

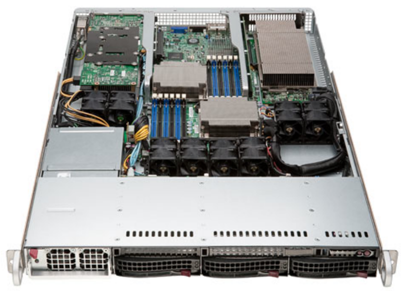


Figure 2. Integrated GPU/CPU 1U server (image courtesy of NVIDIA).

The architecture of the CMS HLT [?],[?] combines the flexibility gained by using the offline reconstruction [?] with the robustness required for reliable online operation of the DAQ. Figure 1 shows a schematic of the architecture of the CMS DAQ system. Event fragments are read out and stored in Readout Units (RU) for each event accepted by the Level-1 trigger. The fragments are subsequently assembled into complete events by an Event Builder through a complex of switched networks into âĂIJevent buffersâĂİ (BU). The full event content is then handed to one of the HLT Filter Units (FU). The FU execute a series of physics reconstruction and filter algorithms and events that are found to be sufficiently interesting for offline analysis are forwarded to the Storage Manager (SM). The decoupling of physics algorithm execution from data flow allows each FU to continue operation, recover the content of the problematic event, and forward it to be stored unprocessed. The FU architecture consists of two separate applications, the ResourceBroker, which exchanges data with the DAQ, and the EventProcessor, which integrates the reconstruction software. Given the involved event sizes and rates, reformatting raw events consisting of a large number of small data fragments into the physical memory of a Filter Unit requires high bandwidth I/O, while the subsequent HLT processing is mostly CPU intensive. The EventProcessor previously mentioned encapsulates the event processing machinery of the CMS reconstruction, providing the full flexibility required to execute complex physics algorithms.

The farms of machines where these event selections are executed is a natural place where the GPUs could be integrated. A simple solution is shown in Figure 2 with a 1U server containing dual multicore CPUs and two NVIDIA Tesla GPUs. While this would provide a single, integrated 1U GPU/CPU server this of course implies replacing the existing servers, evaluating the new power consumption, and the cost of these servers relative to any other options.

4. Inner Tracker

Both the CMS and ATLAS inner tracker include a silicon pixel detector and a silicon strip detector. All tracker layers provide two-dimensional hit position measurements, but only the pixel tracker and a subset of the strip tracker layers provide three dimensional hit position measurements. In the following only the CMS tracker would be considered however the detector performances are comparable in CMS/ATLAS experiments and the tracking algorithm and results discussed are applicable for both of these experiments as well. The CMS pixel detector includes three barrel layers and two forward disks on either end of the detector surrounded by ten strip detectors barrel layers plus three inner disks and nine forward disks at each end of the detector. Owing to the strong magnetic field and the high granularity of the silicon tracker, promptly produced charged particles with transverse momentum $pT = 100$ GeV/c are reconstructed with a resolution in pT of 1.5% and in transverse impact parameter d0 of 15 mm. The track reconstruction algorithms are able to reconstruct displaced tracks with transverse impact parameters up to 25 cm from particles decaying up to 50 cm from the beam line. The performance of the track reconstruction algorithms has been studied with data [11]. The silicon tracker is also used to reconstruct the primary vertex position with a precision of $\sigma_d 20$ mm in each dimension.

5. Fast Tracking Algorithm

One possible application of the GPU enabled parallelism would be to accelerate the performance of the existing Kalman fitter used for tracking reconstruction. Since the iterative Kalman process for each track is largely independent of the other tracks, parallelization should not be conceptually difficult. However, an additional motivating force behind the use of GPU in the HLT is to be able to investigate the possibility of running trigger algorithms which are dramatically and qualitatively different in nature to enhance the discovery potential.

As an example, the Hough Transform is a well known algorithm used in various machine vision applications that differs from the Kalman approach in that it does not operate on localized features of a data set. Rather the technique is in some sense more holistic, operating on an entire image as a whole. The technique is described at length in existing literature, but the concept as it would be applicable to the tracking problem is to consider the parameterization of any given track. Any given hit observed in the detector, for example the simulated straight line track data shown in Figure 3, can correspond to many different possible tracks in parameter space as shown in Figure 4. When integrated over the entire data set, peaks appear in parameter space at the values corresponding to the actual, physical trajectories as shown in Figure 5.

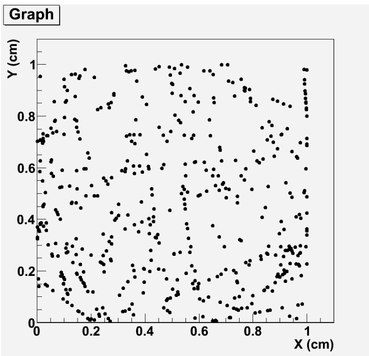


Figure 3. Simulated event made up of 50 straight line tracks, with 10 hits per track.

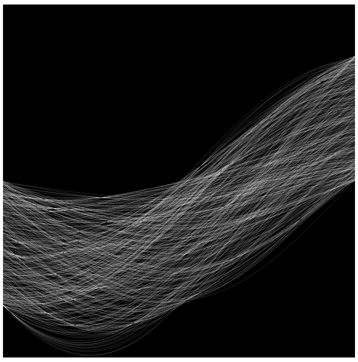


Figure 4. Each hit results in a sinusoidal line in the parameter space. Locations where many of these sinusoidal lines cross are likely to be tracks in the original data.

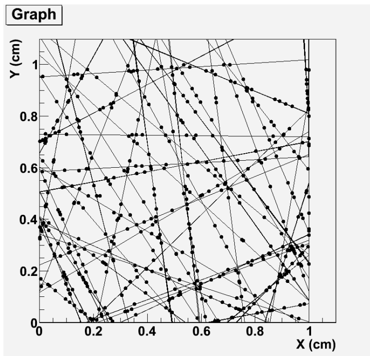


Figure 5. Candidate tracks identified from finding local maxima in the parameter space.

6. Preliminary Results

As was briefly described previously, the traditional Kalman approach implicitly depends on various assumptions and constraints on the phase space of the seed tracks, with one of the most significant results being suppression of displaced vertices. It is hoped that a more holistic algorithm can be used to supplement the existing tracking algorithm in a significant way. As a demonstration of this a simple tracking algorithm based on the Hough Transform was developed and run against a stand alone Monte Carlo simulation with a simple tracker geometry file. The preliminary investigation was performed using Intel multicore CPUs and NVIDIA Tesla C2075 and K20c GPUs.

The Hough Transform algorithm is not trivial to implement in parallel because the computation of the parameter space is subject to race conditions whereby it is not necessarily safe to have multiple threads arbitrarily make updates as hits are processed. In the interest of time and having an unbiased performance comparison for the GPU implementation it would have been preferable to use the CPU implementation of the Hough Transform from the Intel Performance Primitives (IPP) library. However, it was observed that the implementation in IPP is not parallelized and so to provide a more fair performance comparison a parallel implementation of the Hough Transform for the CPU has been developed as part of this work. This CPU implementation is parallelized using OpenMP threads and speedups of 2.6 - 3.6 were achieved on a Intel Core i7-3770 (Ivy Bridge, Quad Core, 3.4 GHz) processor. The CPU used for the performance results is physically a quad core CPU but supports up to two threads per physical core using Simultaneous Multithreading (SMT), with the Intel implementation of SMT known as HyperThreading (HT). SMT did not show any performance advantage for this application because the speedup actually dropped as the number of threads exceeded the number of physical cores. For the GPU implementation, performance results for the two most recent generations of NVIDIA GPUs are shown in Figure 6. Note that the GPU implementation is up to 100x faster than the best multithreaded CPU implementation.

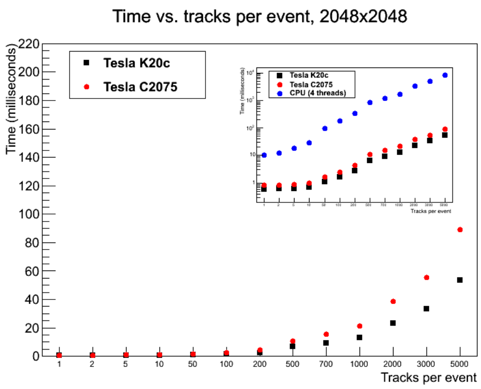


Figure 6. Performance comparison of the Intel CPU (using 4 threads), NVIDIA Tesla C2075, and NVIDIA Tesla K20c.

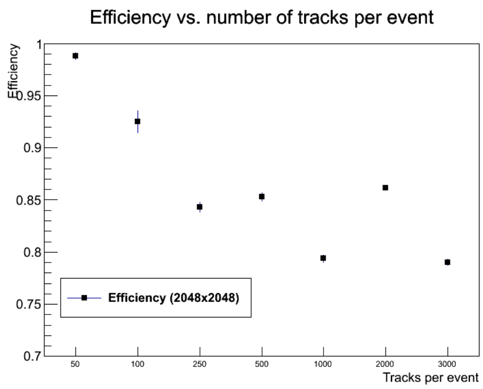


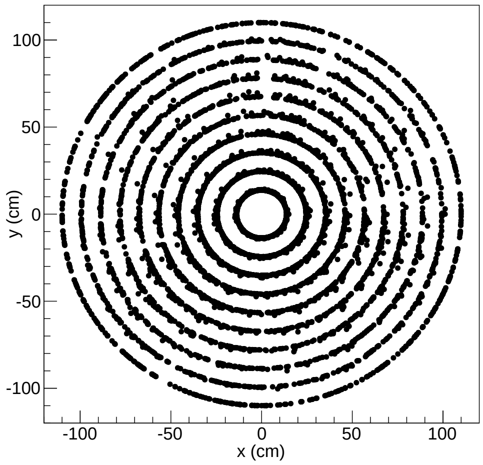
Figure 7. Efficiency as the number of tracks per event is varied.

Although the Hough Transform is most often associated with finding straight lines in data it is applicable to any feature which can be represented by a finite number of parameters. For example, circles of unknown location and radius and location can be identified in a three dimensional parameter space representing the x and y location of the center plus a third parameter for the radius. Figures 8 and 9 show an example of results for our Hough Transform implementation used for identifying curved tracks. Similar to straight tracks, this curved track implementation is in terms of a two dimensional parameter space by constraining the tracks to pass through the interaction point. The computational cost of the Hough Transform is strongly dependent on the total number of parameters while the details of the particular parameterization has essentially no effect on the performance.

While the Hough Transform is a critical part of the proposed tracking algorithm, it is not the only aspect of the computation. After candidate tracks have been identified from analysis of the parameter space it may be necessary to perform a fit of the hits to the candidate tracks. This fitting operation is well suited to parallelization on the GPU, but the performance results in Figures 10 and 11 shows that a multithreaded CPU implementation requires only a few milliseconds. In comparison the Hough Transform on the GPU requires many tens of milliseconds so the fitting operation is relatively low priority for performance optimization. If or when fitting became a significant performance bottleneck it could certainly be run on the GPU to further increase performance.

7. New Tracking Capabilities

The ability to handle displaced vertices provides the opportunity to identify interesting new physics such as jets and black holes. To demonstrate this capability consider the tracks in Figure 12 which originate from a displaced vertex. Again the Hough Transform is applied to the corresponding hits



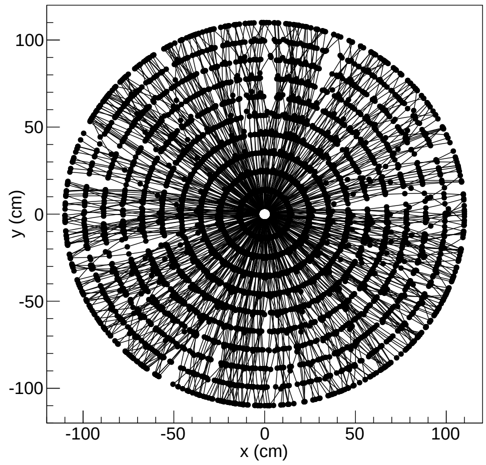
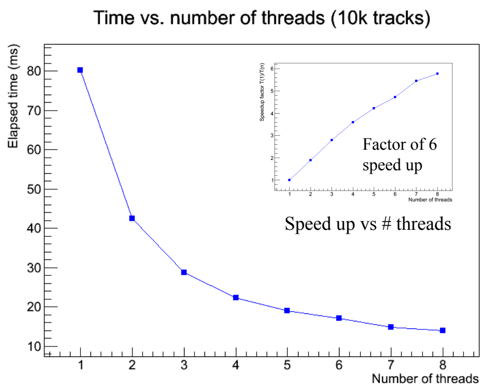


Figure 8. Monte Carlo data for curved tracks originating from the interaction point.

Figure 9. Curved tracks identified using the Hough Transform.



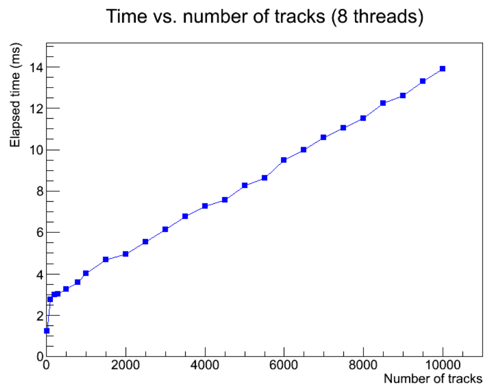


Figure 10. CPU speedup for the fitting computation.

Figure 11. Time required for the fitting computation as the number of tracks is varied.

making up these tracks to generate the parameter space in Figure 13. Recall that the parameter space identifies candidate tracks as locations where many sinusoids cross one another. In Figure 13 the locations of these crossings are visually apparent as the bright spots in the parameter space. However, the analysis is taken one step further by noting that the crossings all lie along a single sinusoidal curve in the parameter space. This single sinusoidal curve has been isolated in Figure 14 and is significant because that one sinusoidal curve in the parameter space corresponds to a point in the original x-y space. More importantly that point in the original x-y space corresponds to the vertex of the displaced jet. And interestingly the most convenient way to implement this process is to use the Hough Transform a second time, this time applying it not to the hits in the x-y space but the crossings in the parameter space. Using the appropriate parameterization to look for sinusoidal curves instead of straight lines the curve in Figure 14 is identified and the vertex of the jet is now located. Figures 15 - 17 illustrate the same technique applied to an event with multiple displaced jets.

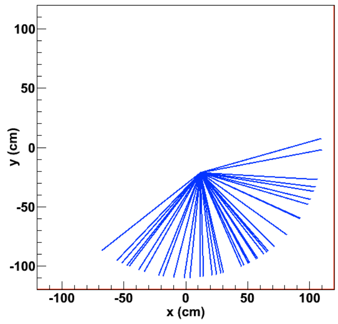


Figure 12. Tracks originating from a displaced jet.

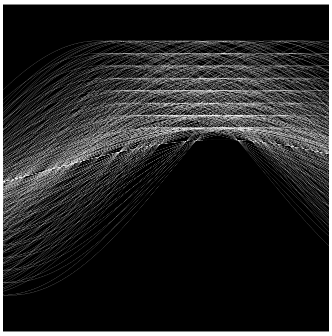


Figure 13. Parameter space shows the maxima are on a single sinusoid.

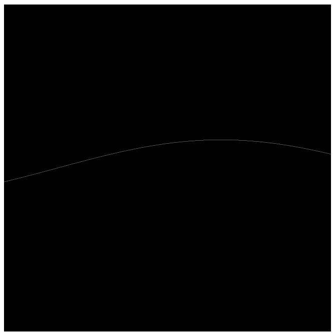


Figure 14. Second Hough Transform identifies the sinusoid corresponding to the jet vertex.

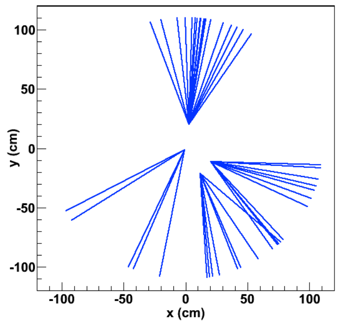


Figure 15. Tracks originating from multiple displaced jets.

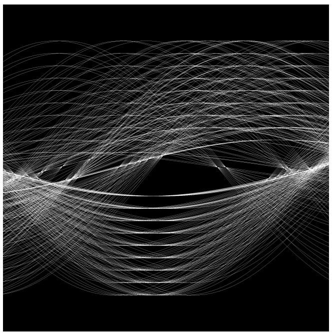


Figure 16. Parameter space is difficult to visually interpret.

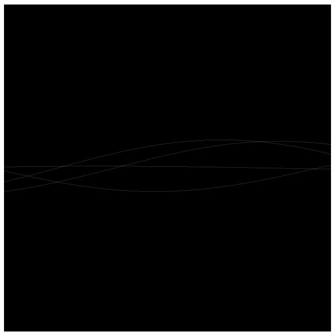


Figure 17. Second Hough Transform identifies the sinusoids corresponding to the jet vertices.

8. Summary

Possible sentences for future work to better optimize CPU performance:

To better understand the relative performance of CPUs and GPUs we intend to implement vectorization on the CPU to fully utilize the Advanced Vector Extensions (AVX) capability present on modern CPUs from Intel and AMD. Although we feel the GPU will still show a compelling performance advantage for this class of computations, the 256 bit AVX capabilities should increase the CPU performance and allow us to better quantify that advantage.

References

[1] ATLAS Collaboration, *Technical Design Report for the ATLAS Muon Spectrometer*, CERN/LHCC/97-22, May 1997.

[2] The ATLAS collaboration, *The ATLAS Experiment at the CERN Large Hadron Collider*, JINST **3** S08003 (2008)

[3] The CMS collaboration, *The CMS Experiment at the CERN Large Hadron Collider*, JINST **3** S08003 (2008)

[4] M. J. Strassler, K. M. Zurek, *Echoes of a hidden valley at hadron colliders*, Phys.Lett.B651:374-379,2007