Nuclear Inst. and Methods in Physics Research, A 957 (2020) 163427

Contents lists available at ScienceDirect

Nuclear Inst. and Methods in Physics Research, A

journal homepage: www.elsevier.com/locate/nima

A fast method for particle tracking and triggering using small-radius silicon detectors

Ashutosh V. Kotwal

Department of Physics, Duke University, Durham, NC 27708, USA

ARTICLE INFO

Keywords:
Charged particle tracking
Track trigger
Unsupervised machine learning
Field-programmable gate array
Electronics

ABSTRACT

We propose an algorithm, deployable on a highly-parallelized graph computing architecture, to perform rapid reconstruction of charged-particle trajectories in the high energy collisions at the Large Hadron Collider and future colliders. We use software emulation to show that the algorithm can achieve an efficiency in excess of 99.95% for reconstruction with good accuracy. The algorithm can be implemented on silicon-based integrated circuits using field-programmable gate array technology. Our approach can enable a fast trigger for massive charged particles that decay invisibly in the tracking volume, as in some new-physics scenarios related to particulate dark matter. If production of dark matter or other new neutral particles is mediated by metastable charged particles and is not associated with other triggerable energy deposition in the detectors, our method would be useful for triggering on the charged mediators using the small-radius silicon detectors.

The reconstruction of charged-particle trajectories is one of the important experimental tasks in collider physics. Most precision measurements and searches for new physics require reconstruction of all charged-particle trajectories with transverse momentum above a threshold. This task is usually performed with sophisticated software which requires significant computing resources. This approach has been successful but faces challenges on two related fronts. Higher data rates will require corresponding increases in resources. Secondly, rare signals involving charged particles may be difficult to identify rapidly if they are not accompanied by other triggerable energy deposits in the detectors. Triggering capability for such events has to rely on fast "track triggers". We propose an algorithm to perform rapid reconstruction and momentum estimation of charged-particle trajectories using the silicon detectors at small radius in collider experiments. The speed is achieved by embedding the algorithm in a highly-parallelized graph computing architecture, build using commercial field-programmable gate arrays (FPGAs).

Any search for new physics which involves rapid identification of high-momentum charged particles would benefit from our proposed method. While many final states also contain signatures detectable in the calorimeter and muon detection subsystems by existing methods, a class of models predict collision events containing only "disappearing tracks", i.e. charged particles that decay upstream of these detector subsystems and leave no trace in them to enable event identification. In particular, we are interested in short-lived charged particles that travel $\mathcal{O}$ (30 cm) before decaying invisibly. Such particles leave a trace only in the small-radius tracking detectors and disappear before even traversing the large-radius tracking detectors, which extend to $\mathcal{O}$ (1 m) radius. In the latter case our proposed methodology would be most beneficial.

An example is provided by models of particulate dark matter (DM) whose interaction is mediated by metastable charged particles. The principles of relativistic quantum field theory and group theory, embodied in the standard model (SM) of particle physics [1–3], have been astonishingly successful in describing all known fundamental particles and their interactions including the spectacular discovery of the Higgs boson [4–7] at the Large Hadron Collider (LHC) [8,9]. In parallel, the discovery of dark matter via its gravitational interactions on the galactic and cosmological distance scales [10–12] has revolutionized our understanding of large-scale structure formation since the Big Bang. All cosmological data are consistent with DM comprising about 84% of the matter in the universe [13]. Dwarf galaxies, comprising mostly of DM, have recently been discovered [14]. Since DM cannot be accounted for in the SM, it is plausible that DM comprises of one or more new species of particles [15]. Production of such particles at the LHC has been discussed in many theories of particulate DM [16]. Elastic scattering of DM particles off atomic nuclei in cryogenic materials, and astrophysical detection of visible particles produced in the mutual annihilation of DM particles in outer space, are being pursued by dedicated underground, ground-based and satellite-based experiments [13].

We consider, as an example, the production of the metastable charged mediators which subsequently decay to DM particles. The known properties of the weak and electromagnetic interactions make it very plausible that, if the DM particle is the lightest particle in a symmetry group multiplet, other members of the multiplet carry electric charge [17,18]. Therefore the LHC may serve as a DM factory

E-mail address: ashutosh.kotwal@duke.edu.

https://doi.org/10.1016/j.nima.2020.163427
Received 2 November 2019; Received in revised form 9 January 2020; Accepted 11 January 2020
Available online 15 January 2020

through copious production of the heavier, charged partners of the DM particles. It is expected that the parent and daughter particles are almost degenerate in mass, therefore the decay produces negligible associated energy. Should the parent particles be produced in pairs, as required by known conservation laws, and they decay to DM within the volume of the experimental apparatus, the identification of such collision events in a time interval commensurate with the collision rate is not possible with current technology. The challenge is the identification of metastable charged mediators within the short time interval of $\mathcal{O}$ (4 μs) which is necessitated by the high collision rate, in the presence of thousands of other charged particle trajectories produced simultaneously.

The trajectories of charged particles produced at the LHC are intercepted by highly granular silicon sensors, which at central pseudorapidities are placed in concentric cylinders surrounding the colliding beams [19,20]. Starting in the mid-2020's, the upgraded LHC running at high luminosity (HL-LHC) will produce up to 200 proton–proton (pp) collisions every 25 ns [21], each collision producing about seven charged particles per unit of pseudorapidity [22]. The position of each charged particle is recorded by about 10 silicon sensors along its trajectory [8,9], resulting in a point cloud of ≈110,000 points recorded every 25 ns in a tracking detector covering eight units of pseudorapidity.

Reconstruction refers to the task of partitioning the point cloud into sets of points, one point on each sensor layer, such that the elements of each set are the points created by a single charged particle. Ideally, the entire point cloud is partitioned into the ≈11,000 sets associated with the respective particle trajectories. Current reconstruction algorithms require ≈ 4 s on a single CPU core to reconstruct an event with 20 concurrent pp collisions [23]. With an improved HL-LHC detector and upgraded software, the CPU time can be reduced by about a factor of two on the same CPU core, for an event with 200 concurrent pp collisions [23]. A significantly faster technique, implemented using associative memories to match the recorded patterns with stored patterns, is under construction for ATLAS [24,25]. Due to its inherent large parallelism, this custom electronic device has a much shorter latency of 50–100 μs, which will be reduced further to 24 μs [26,27]. However, it requires the simulation and storage of billions of patterns for matching, and is incapable of running on the $\mathcal{O}$ (4 μs) time scale. A similar proposal [28,29] has also been made for CMS. In the ATLAS proposal [26,27], the search would be restricted to regions of interest already defined in the vicinity of other triggerable objects in the event. In another set of CMS proposals [30–35], a special silicon detector geometry is exploited by arranging sensor layers as closely-spaced pairs to obtain track stubs. These stubs are used as local tangents in the track-finding. These stub-based approaches use the large-radius ($25 < r_{\text{sensors}} < 110$ cm) silicon detector and cannot be used to trigger on disappearing tracks. Techniques based on deep learning [36,37] are applications of supervised machine learning and require training data.

We propose an algorithm wherein the logic is executed on a massively-parallel graph computer [38]. Since this method requires no training on preclassified data, it represents a type of unsupervised learning that partitions the input points into clusters. Compared to certain other methods of unsupervised learning, our method is not statistical and does not require large datasets to learn from, does not involve diagonalization of large matrices (unlike principal component analysis), and does not require initialization values (unlike k-means clustering). Furthermore, our method only uses nearest neighbor information, whereas k-means clustering and the expectation–maximization algorithm require a global analysis of all data points. Thus, these unsupervised learning methods cannot be processed in a parallelized, distributed computing architecture as our method can. We demonstrate a novel algorithm that can identify charged particle trajectories in a large point cloud, and how such an algorithm lends itself to implementation on a silicon integrated circuit using FPGA technology. We verify the accuracy of the algorithm and the scalability of the FPGA implementation for the task of triggering on metastable charged mediators.

Charged particles traversing an axial magnetic field in a cylindrical detector execute a helical trajectory, parameterized by functions $\phi(r)$ and $z(r)$, where ϕ denotes the azimuthal angle around the cylinder axis, and z and r denote the longitudinal and radial coordinates respectively. The particle's momentum perpendicular to the beam (z) direction, $p_T \propto BR$ where B denotes the strength of the magnetic field and R is the helix radius. Defining the curvature $c = (2R)^{-1}$ and ϕ_0 denoting the particle's azimuthal angle at emission, we have

$$\sin(\phi - \phi_0) = cr \tag{1}$$

If each point is recorded by a two-dimensional pixel detector, the measurement is denoted by h with attributes of ϕ and z respectively. The z coordinate is given by

$$z - z_0 = \frac{\lambda}{c}(\phi - \phi_0) = \frac{\lambda}{c}\sin^{-1}(cr) \tag{2}$$

where the constants λ and z_0 specify the cotangent of the polar angle and the z-position of the particle at emission, respectively. It is convenient to normalize z by the longitudinal length of the sensor layers in order to convert it to a dimensionless angle commensurate with the azimuthal angle ϕ. Upon reconstruction, the points associated with each particle's trajectory can be used to calculate λ, z_0, ϕ_0 and c (equivalently, R), and the latter yields a measurement of the particle's p_T using the knowledge of B.

1. Graph computing algorithm

The point cloud may be represented by a two-dimensional matrix of points $h_{i,l}$ where l denotes the sensor layer and i denotes the point's ordinal number in that layer. For simplicity we assume that all charged particles create energy deposits in all sensor layers. In future work, we will investigate relaxing this assumption so that multiple charged particles may be associated with the same point in a given layer (i.e. particle trajectories may intersect at any sensor layer). Each point $h_{i,l}$ is associated with a ϕ measurement coordinate (in the case of one-dimensional sensors) or (ϕ, z) measurement coordinates (in the case of two-dimensional sensors). These coordinates represent the silicon pixel clusters to be provided *a priori* by the detector readout, similar to the scheme proposed for CMS [30]. A cluster is defined as a group of neighboring pixels having energy deposits above some threshold.

One may convert the $h_{i,l}$ matrix into a graph by associating links $w_{ij,l}$ between each point $h_{i,l}$ and all possible points $h_{j,l+1}$ in the adjacent outer layer of sensors. According to the procedure of Ref. [38] for computing derivatives on a graph, the values of the link weights are set to be inversely proportional to the radial distance between consecutive layers. The combinatorial problem of reconstruction is equivalent to pruning the $w_{ij,l}$ matrix (i.e. eliminating all spurious links) until the surviving links partition the point cloud into sets of linked points and each set is associated with a physical particle trajectory.

Viewing each point as a node in the graph, we postulate that a massively-parallel graph computer can be constructed using modern FPGA technology. Each node has a small, dedicated processor that stores a slice of the matrix $h_{i,l}$, and is optimized for addition, multiplication and sorting operations.

The solution to the combinatorial problem is obtained by considering the graph operator $\Box_{ijk,l}$ at each node (i,l), which is a function of first and second derivatives computed using the triplet of points $h_{i,l}$, $h_{j,l+1}$ and $h_{k,l-1}$. From Eq. (1) we find

$$\begin{aligned} \phi' &\equiv \frac{d\phi}{dr} = c\sec(\phi - \phi_0) \\ \phi'' &\equiv \frac{d^2\phi}{dr^2} = \tan(\phi - \phi_0)(\phi')^2 \end{aligned} \tag{3}$$

Particles of interest have high p_T ($c \to 0$), implying $\phi' \to c$ and $\phi'' \to rc^3$. Therefore, $[\phi'' - r(\phi')^3] \to 0$. Similarly, Eq. (2) yields

$$z' \equiv \frac{dz}{dr} = \lambda[1 - (cr)^2]^{-\frac{1}{2}}$$

$$z'' \equiv \frac{d^2 z}{dr^2} = rc^2\lambda[1-(cr)^2]^{-\frac{3}{2}} \tag{4}$$

Therefore $z' \rightarrow \lambda$ and $z'' \rightarrow rc^2\lambda$ for high-p_T particles, and $[z'' - r(\phi')^2 z'] \rightarrow 0$.

The graph operator $\Box_{ijk,l}$ can be computed at each node (i,l) (starting at $l = 2$ because the first layer has no preceding layer to compute derivatives with respect to) for all combinations of links to the previous and the next layer,

$$\Box_{ijk,l} = \phi''_{ijk,l} - r_l(\bar{\phi}'_{ijk,l})^3 + z''_{ijk,l} - r_l(\bar{\phi}'_{ijk,l})^2\bar{z}'_{ijk,l} \tag{5}$$

where the first derivatives are computed as the link-weighted differences of ϕ or z values at the two nodes connected by the link,

$$\begin{aligned} \phi'_{ij,l} &= w_{ij,l}(\phi_{j,l+1} - \phi_{i,l}) \quad ; \quad \phi'_{ki,l} = w_{ki,l-1}(\phi_{i,l} - \phi_{k,l-1}) \\ z'_{ij,l} &= w_{ij,l}(z_{j,l+1} - z_{i,l}) \quad ; \quad z'_{ki,l} = w_{ki,l-1}(z_{i,l} - z_{k,l-1}) \\ \bar{\phi}'_{ijk,l} &= (\phi'_{ij,l} + \phi'_{ki,l})/2 \quad ; \quad \bar{z}'_{ijk,l} = (z'_{ij,l} + z'_{ki,l})/2 \end{aligned} \tag{6}$$

and the second derivatives are computed as the respective differences of first derivatives at the middle (shared) node,

$$\begin{aligned} w_{ijk,l} &= 2/(w^{-1}_{ij,l} + w^{-1}_{ki,l-1}) \\ \phi''_{ijk,l} = w_{ijk,l}(\phi'_{ij,l} - \phi'_{ki,l}) &\quad ; \quad z''_{ijk,l} = w_{ijk,l}(z'_{ij,l} - z'_{ki,l}) \end{aligned} \tag{7}$$

We find the criterion for valid trajectories to be $\Box_{ijk,l} \rightarrow 0$ for both one- and two-dimensional sensors, at each point of the graph. In case the measurement resolutions in the two directions are unequal, Eq. (5) can be optimized by de-weighting the terms with the worse resolution.

Eq. (5) encodes the list of all combinatorial trajectories through node (i,l). Valid local trajectories will produce small values of $\Box_{ijk,l}$ while invalid trajectories will produce large values of $\Box_{ijk,l}$. Each node is equipped with a sorting unit that ranks the values of $\Box_{ijk,l}$ over the triplets (i,j,k) in ascending order. Using the sorted list of triplets, we produce a sorted list of links such that the rank of a link is given by the order in which the link appears for the first time in the ordered list of triplets. The higher the rank of a link, the larger the values of $\Box_{ijk,l}$ it contributes in combination with any other link. Invalid links are iteratively removed by pruning the link (either ij or ik) with the highest rank (i.e. the worst link). The iterations at each node (i,l) are terminated when the node is left with one link each to the next and previous layer.

For N charged particles, the computational time cost of this algorithm involves $\mathcal{O}(N)$ iterations, each iteration removing the worst link at each node. Finding the worst link requires sorting $\mathcal{O}(N^2)$ terms. The sorting is performed once before the iterations, as each iteration simply drops the last link in the list. In future work, the iterations will be optimized to prune the worst percentile in each iteration, reducing the number of iterations to $\mathcal{O}(\log N)$. Note that initially there were $\mathcal{O}(N^2M)$ links to be pruned, where M represents the number of sensor layers. The hardware cost is the number NM of graph nodes, each equipped with the calculator-sorter unit.

Specialized algorithms and FPGA implementations for sorting large lists and finding the minimum or maximum in a list exist [39–42]. Since our algorithm can proceed from course-grained to fine-grained sorting to find the best link combinations, we can process 16^k numbers in k sequential steps using a 16-input sorter. It has been shown [43] that a 16-input sorter for 32-bit integers can be implemented on a Xilinx Spartan 6 LX45 FPGA using 5% of its logic resources. In comparison to this FPGA's 43,000 logic cells, a more modern Virtex-7 2000T FPGA has 4.4 million logic cells, a factor of 100 increase in available hardware resources. Thus, modern FPGAs can accommodate $\mathcal{O}$ (2k) sorter units, which we show below to be adequate for our implementation. The exponential growth of data processing worldwide has created a growing field of R&D into FPGA-based co-processors and accelerators for data sorting and ranking, to augment traditional CPU-based search algorithms. Our methodology is well-situated to take advantage of these technical developments in FPGA sorter architectures.

Additional hardware will be required for reading out and preprocessing the raw detector hits into clusters and routing this information to the track-finding circuits. We expect these tasks to be similar to the readout and routing requirements of the other proposals for ATLAS and CMS track triggering [24–35,44], and synergistic solutions for these requirements can be pursued.

2. Results

The success of this algorithm is demonstrated by the following emulation. We generate point clouds, shown in Fig. 1, from the intersections of 100 particles traversing five silicon sensor layers spaced 5 cm apart in a 2 T magnetic field, over an azimuthal domain of width one radian. This detector geometry is representative of the upgraded ATLAS [45] silicon pixel barrel detector for the HL-LHC, which will be placed at the center of a cylindrical magnetic spectrometer of approximate radius 1 m. The CMS tracking volume is smaller but with a higher magnetic field of 3.8 T [46]. Since the pp collision region has a relatively small longitudinal length of $\approx$ 0.5 cm, a projective slice of the silicon detector of longitudinal width 15 cm would certainly contain a particle's complete trajectory in the z-direction. Therefore, a cylindrical detector of length 1.5 m (approximately the length of the HL-LHC ATLAS pixel barrel detector, covering four units of central pseudorapidity) illuminated by 5600 particles would result in about 100 particles contained in a wedge of azimuthal width one radian and longitudinal width of 15 cm.

The transverse momentum (p_T) spectrum in the emulation is realistically soft, modeled as an exponentially-falling distribution in p_T^2, $\frac{dN}{dp_T^2} \sim e^{-(p_T/\beta)^2}$. The coefficient β is chosen so that the p_T spectrum peaks at $p_T = 250$ MeV [47], as shown in Fig. 2. This model is in reasonable agreement with the measured p_T spectrum [22] of soft particles from minimum-bias pp collisions, but does not provide enough statistics for studying the algorithm performance at high p_T. Therefore we embed a second distribution of the form $\frac{dN}{dp_T} \sim p_T^{-2}$ for $p_T >$ 500 MeV, which is also soft but with a harder tail to model high-p_T particles. The combined spectrum enables the study of the algorithm as a function of p_T.

In this first study, we have not included noise hits, sensor inefficiencies and sensor resolution, in order to understand the viability of our approach with a perfect pixel detector. In subsequent studies we plan to investigate the impact of these effects on algorithm performance.

For simplicity we show the results of the emulation using the azimuthal point coordinates only. The reconstructed trajectories found by the algorithm are shown in Fig. 1. All 100 trajectories are reconstructed, and the high quality of reconstruction is demonstrated in Figs. 3 and 4. Each point on a reconstructed trajectory is compared to its progenitor particle trajectory, and the number of correctly and wrongly assigned points per track are shown in Figs. 3 and 4 respectively. These rates as functions of curvature are shown in Fig. 5. These results are averaged over 160 emulated events.

We conclude that the track-finding efficiency exceeds 99.95%. The probability per track to lose a correct point is $<$ 0.1% and to have a spurious point assigned is (1.6 ± 0.3)%, with no significant momentum dependence. Thus, in the unique cases mentioned in the introduction, where new physics manifests as "disappearing tracks" with no identifiable signatures at radii beyond ≈ 30 cm, our methodology can identify such tracks with high efficiency using the small-radius tracker only. The tracks will be robust, with only about one in 100 tracks having a spurious hit assigned. Since there are five sensor layers, the rate of hit loss is $<$ 0.02% and the rate of spurious hit assignment is (0.32 ± 0.06)%.

3. Discussion

Our algorithm can be implemented on a single FPGA integrated circuit containing an array of 100 × 5 arithmetic-sorter units, capable of processing 100 tracks in a wedge with 5 points per track. The

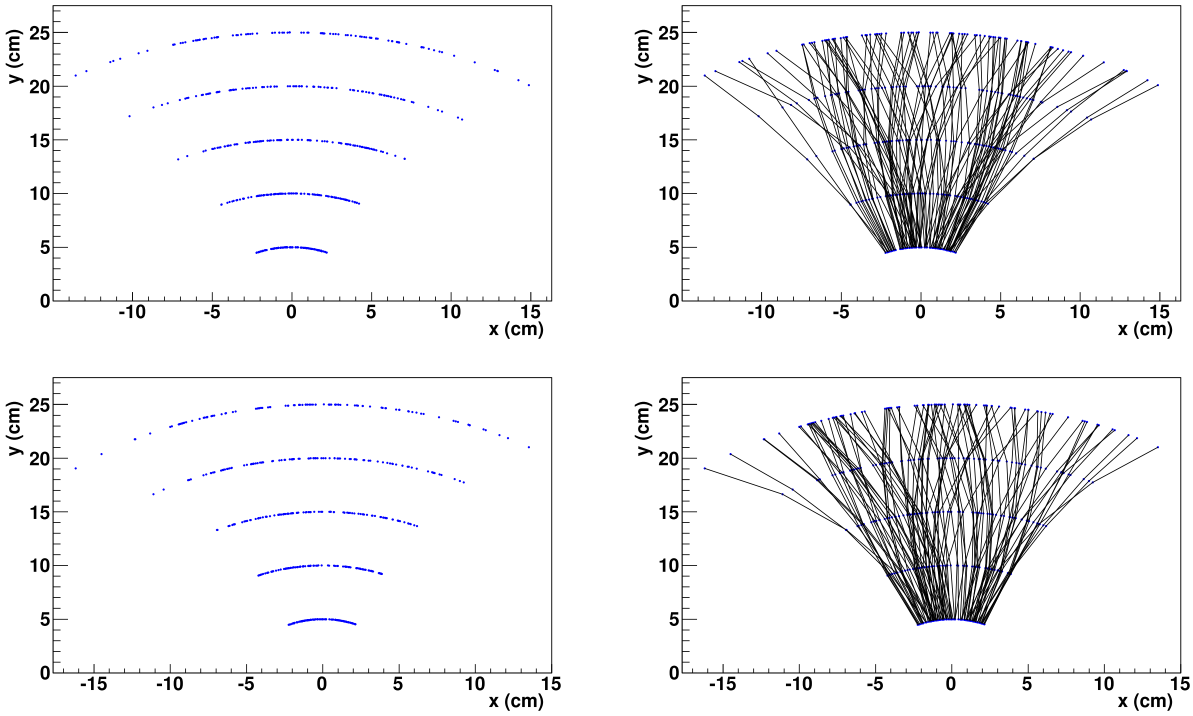

Fig. 1. (left) Two examples of the point cloud generated by 100 particles in an azimuthal sector of the silicon pixel detector of width one radian. The silicon sensors are placed in concentric circles with radial separation of 5 cm. (right) The reconstruction of 100 tracks from the respective point clouds.

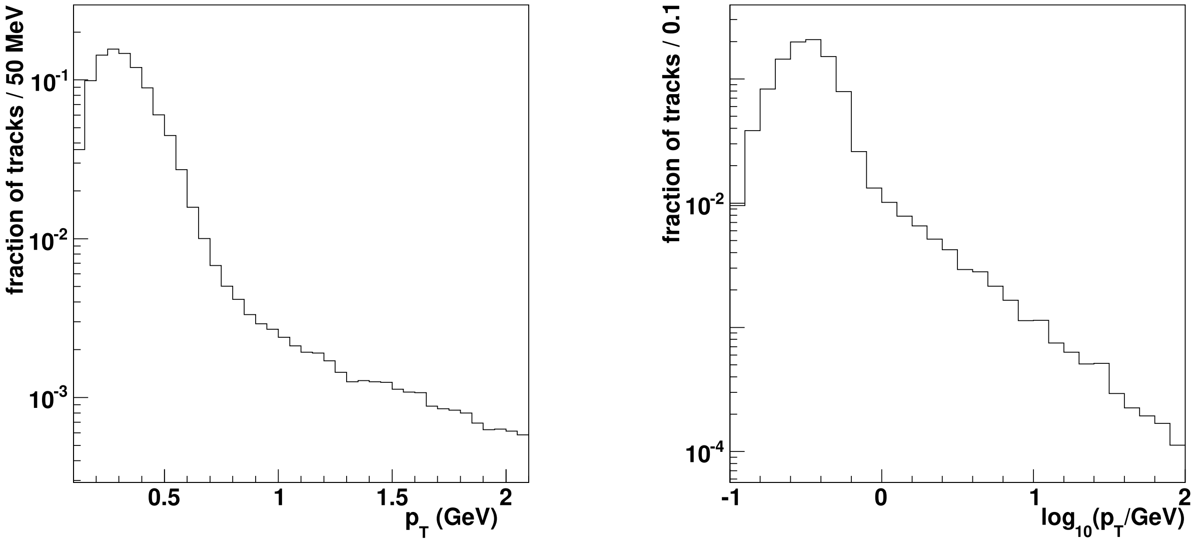

Fig. 2. The p_T spectrum of the generated tracks used to create the point clouds of Fig. 1.

implementation is eminently possible on modern FPGAs containing billions of transistors. As mentioned above, a Virtex-7 2000T FPGA with its 6.8 billion transistors can accommodate $\mathcal{O}$ (2k) arithmetic-sorter units, adequate for the 500 units needed for this implementation. Furthermore, in comparison with the 28 nm transistor feature size used in the manufacture of the Virtex-7 series, the latest Versal series of FPGAs from Xilinx are manufactured using 7 nm feature size, and contain 50 billion transistors. The steadily increasing logic resources available can only increase the processing speed and versatility of our method's implementation. To process $100^2 = 10{,}000$ link pairs would require running the 16-input sorters for four sequential iterations, which is acceptable from the latency perspective. According to Ref. [48], the latency for a 16-input sorter is 40 ns for a 220 MHz clock frequency on a Xilinx Virtex-5 FX130T FPGA, leading to an estimate of 160 ns for a progressive course- to fine-grained search for the best links.

Following the task of track pattern recognition, triggering schemes typically require a second step of track-fitting and momentum estimation [24–35,44]. In our approach, momentum estimation is already completed since the evaluation of the derivatives in Eq. (6) provides an estimate of each track's curvature. Therefore, subsequent processing steps for track-fitting and momentum estimation are not required. The trigger curvature threshold could be stored in the FPGA and applied on the found tracks to produce a trigger decision directly.

We also note that $\mathcal{O}(120)$ such FPGA chips will be able to perform full particle reconstruction for the LHC experiments, since one FPGA is used to process a pseudorapidity interval of ≈ 0.4 and 1/6 of the azimuth. Currently, traditional software codes running on computer clusters are expected to require a factor of five more computing power than the budget allows [49]. Our approach provides a promising solution

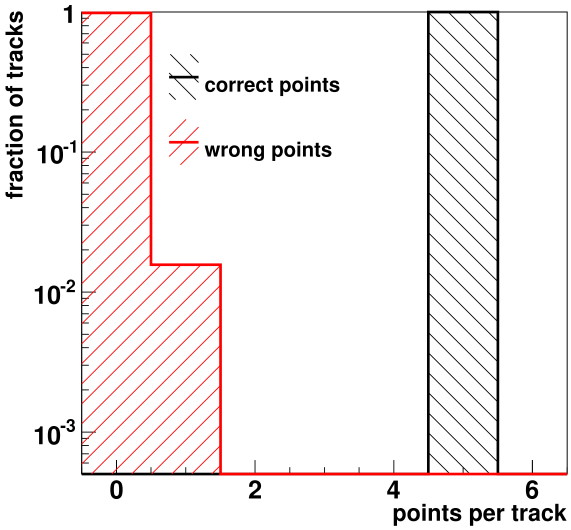

Fig. 3. The distributions of the number of correctly and wrongly assigned points for particle $p_T > 500$ MeV and five sensor layers.

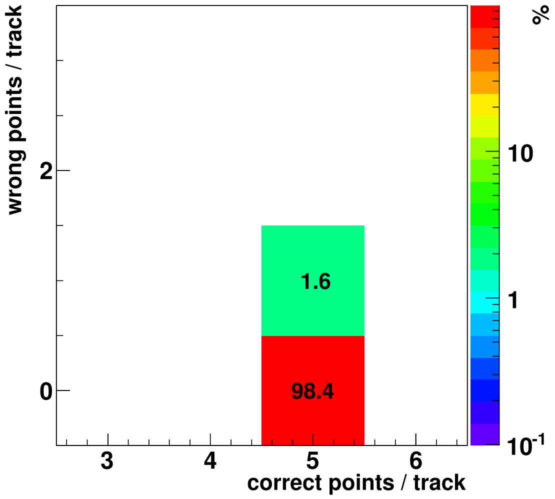

Fig. 4. The two-dimensional distribution of the number of correctly and wrongly assigned points, corresponding to Fig. 3, for particles with $p_T > 500$ MeV and five sensor layers. The bin contents indicate the percentage of all tracks with a particular combination of correct and wrong points.

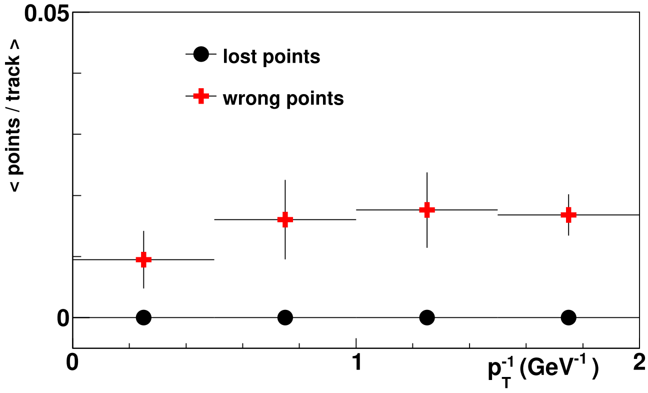

Fig. 5. The mean number of lost points and wrongly assigned points as a function of particle curvature (inverse transverse momentum p_T), for five sensor layers.

to this significant problem, in addition to providing track-triggering capability.

The event-processing time for our FPGA circuit implementation can be reduced dramatically if the task is restricted to higher p_T particles. If the minimum p_T of reconstructed particles was raised to 1 GeV, the width of the azimuthal wedge processed can be reduced by a factor of 10, to 0.1 radians, and still contain the complete trajectory of the particle. This reduces the number N of tracks to be processed by a factor of 10, thereby reducing the number of combinations to be sorted by a factor of 100 at each processing node. The number of nodes required is also reduced by a factor of 10. Thus our algorithm and processing circuit has the flexibility to optimize speed versus p_T threshold in order to meet timing and FPGA requirements.

We obtain processing-time estimates using the studies in Ref. [48], which were based on the Xilinx Virtex-5 FX130T FPGA. For an even–odd/bitonic merge sorting network, an FPGA implementation has the number of stages $S(n) = \mathcal{O}(\log^2 n)$, where $n = N^2$ integers are completely sorted. In Ref. [48], a latency of 100 ns was achieved for $n = 64$ 32-bit integers and clock frequency $f_{\text{clk}} = 220$ MHz. For a triggering device with a particle $p_T > 1$ GeV threshold, $N = 10$, $n = 100$ and the latency extrapolates to 130 ns, which is significantly smaller than the 4 μs upper limit set by the LHC experiments. Thus, from the timing perspective, our approach is viable as a high−p_T triggering device for disappearing tracks. For a pipelined implementation, the latency scales as $S(n)/f_{\text{clk}}$ [48] and can be extrapolated to other FPGA clock frequencies. The speed is achieved by sorting directly in hardware and by the highly parallelized, distributive nature of this method's computations.

In subsequent studies we plan to investigate the effects of noise hits, sensor inefficiencies and sensor resolution. We emphasize that the algorithm has no tunable or initialization parameters and requires no training, unlike other methods of supervised or unsupervised machine learning. Our approach can be described as the partitioning of a point cloud into graphs which minimize the total Dirichlet energy. This approach is viable for finding all tracks using only the small-radius silicon pixel detectors, and for fast triggering on the high momentum ones, including those that decay and disappear immediately thereafter.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

CRediT authorship contribution statement

Ashutosh V. Kotwal: Conceptualization, Data curation, Formal analysis, Funding acquisition, Investigation, Methodology, Project administration, Resources, Software, Supervision, Validation, Visualization, Writing - original draft, Writing - review & editing.

Acknowledgments

We thank Raj Iyer, Henry Greenside, Alex Cloninger and Arijit Banerjee for helpful discussions. We acknowledge support from the U.S. Department of Energy, Office of High Energy Physics grant no. DE-SC0010007.

References

[1] S. Glashow, Partial symmetries of weak interactions, Nuclear Phys. 22 (1961) 579.
[2] A. Salam, J.C. Ward, Electromagnetic and weak interactions, Phys. Lett. 13 (1964) 168.
[3] S. Weinberg, A model of leptons, Phys. Rev. Lett. 19 (1967) 1264.
[4] P.W. Anderson, Plasmons, gauge invariance, and mass, Phys. Rev. 130 (1963) 439.

[5] F. Englert, R. Brout, Broken symmetry and the mass of gauge vector mesons, Phys. Rev. Lett. 13 (1964) 321.

[6] P.W. Higgs, Broken symmetries and the masses of gauge bosons, Phys. Rev. Lett. 13 (1964) 508.

[7] G.S. Guralnik, C.R. Hagen, T.W.B. Kibble, Global conservation laws and massless particles, Phys. Rev. Lett. 13 (1964) 585.

[8] G. Aad, et al., ATLAS Collaboration, Observation of a new particle in the search for the standard model Higgs boson with the ATLAS detector at the LHC, Phys. Lett. B 716 (2012) 1.

[9] S. Chatrchyan, et al., CMS Collaboration, Observation of a new boson at a mass of 125 GeV with the CMS experiment at the LHC, Phys. Lett. B 716 (2012) 30.

[10] G. Bertone, D. Hooper, J. Silk, Particle dark matter: Evidence, candidates and constraints, Phys. Rep. 405 (2005) 279.

[11] J.L. Feng, Dark matter candidates from particle physics and methods of detection, Ann. Rev. Astron. Astrophys. 48 (2010) 495.

[12] G. Bertone, D. Hooper, History of dark matter, Rev. Modern Phys. 90 (2018) 045002.

[13] M. Tanabashi, et al., Particle Data Group, The review of particle physics, Phys. Rev. D 98 (2018) 030001.

[14] L.E. Strigari, Dark matter in dwarf spheroidal galaxies and indirect detection: a review, Rep. Progr. Phys. 81 (2018) 056901.

[15] G. Bertone, Particle Dark Matter, Cambridge University Press, 2010.

[16] A. Boveia, C. Doglioni, Dark matter searches at colliders, Annu. Rev. Nucl. Part. Sci. 68 (2018) 429.

[17] J. Bramante, N. Desai, P. Fox, A. Martin, B. Ostdiek, T. Plehn, Towards the final word on neutralino dark matter, Phys. Rev. D 93 (2016) 063525.

[18] M. Low, L.T. Wang, Neutralino dark matter at 14 TeV and 100 TeV, J. High Energy Phys. 1408 (2014) 161.

[19] G. Aad, et al., ATLAS Collaboration, The ATLAS experiment at the CERN large Hadron collider, JINST 3 (2008) S08003.

[20] S. Chatrchyan, et al., CMS Collaboration, The CMS experiment at the CERN LHC, JINST 3 (2008) S08004.

[21] M. Testa, Upgrade of ATLAS and CMS for high luminosity LHC: detector performance and physics potential, Nuovo Cim. C 40 (1) (2017) 8.

[22] M. Aaboud, G. Aad, B. Abbott, et al., Charged-particle distributions at low transverse momentum in $\sqrt{s} = 13$ TeV pp interactions measured with the ATLAS detector at the LHC, Eur. Phys. J. C 76 (2016) 502.

[23] ATLAS Collaboration, Fast Track Reconstruction for HL-LHC, ATL-PHYS-PUB-2019-041.

[24] A. Andreani, et al., The fast tracker real time processor and its impact on muon isolation, τ and b-jet online selections at ATLAS, IEEE Trans. Nucl. Sci. 59 (2012) 348.

[25] N. Kimura, for the ATLAS Collaboration, Fast tracker: a hardware real time track finder for the ATLAS trigger system, JINST 9 (2014) C04012.

[26] A. Cerri, on behalf of the ATLAS Collaboration, L1Track: A fast Level 1 track trigger for the ATLAS High luminosity upgrade, Nucl. Instrum. Methods Phys. Res. A 824 (2016) 263.

[27] M. Mårtensson, on behalf of the ATLAS Collaboration, Fast pattern recognition with the ATLAS L1Track trigger for the HL-LHC, PoS Vertex2016 069.

[28] D. Sabes, L1 track triggering with associative memory for the CMS HL-LHC tracker, JINST 9 (11) (2014) C11014.

[29] G. Fedi, et al., A real-time demonstrator for track reconstruction in the CMS L1 track-trigger system based on custom associative memories and high-performance FPGAs, PoS TWEPP-17 (2018) 138.

[30] R. Aggleton, et al., An FPGA based track finder for the L1 trigger of the CMS experiment at the high luminosity LHC, JINST 12 (2017) P12019.

[31] C. Amstutz, et al., An FPGA based track finder at L1 for CMS at the High Luminosity LHC, CERN Conference Report CMS CR-2016/112.

[32] E. Bartz, et al., FPGA-Based Tracklet Approach to Level-1 Track Finding at CMS for the HL-LHC, in: EPJ Web of Conferences, vol. 150, 2017, p. 00016, arXiv:1706.09225.

[33] Z. Tao, for the CMS Tracker Group, Level-1 Track Finding with an all-FPGA system at CMS for the HL-LHC, arXiv:1901.03745.

[34] G. Hall, for the CMS Collaboration, A time-multiplexed track-trigger for the CMS HL-LHC upgrade, Nucl. Instrum. Methods Phys. Res. A 824 (2016) 292.

[35] T. James, for the TMTT Collaboration, Track finding for the level-1 trigger of the CMS experiment, Springer Proc. Phys. 212 (2018) 296–302.

[36] S. Farrell, et al., Novel deep learning methods for track reconstruction, arXiv:1810.06111.

[37] A. Tsaris, et al., The hep.trkx project: Deep learning for particle tracking, J. Phys.: Conf. Ser. 1085 (2018) 042023.

[38] M. Bronstein, et al., Geometric deep learning: going beyond Euclidean data. arXiv:1611.08097.

[39] C.L. Wey, M.D. Shieh, S.Y. Lin, Algorithms of finding the first two minimum values and their hardware implementation, IEEE Trans. Circ. Syst. I 55 (11) (2008) 3430–3437.

[40] G. Masera, L.G. Amaru, M. Martina, High speed architectures for finding the first two maximum/minimum values, IEEE Trans. Very Large Scale Integration 20 (12) (2012) 2342–2346.

[41] S. Goren, G. Dundar, B. Yuce, H.F. Ugurdag, A fast circuit topology for finding the maximum of N k-bit numbers, Symp. Comput. Arith. (2013) 59–66.

[42] C. Torres-Huitzil, Resource efficient hardware architecture for fast computation of running max/min filters, Sci. World J. (2013) 108103, http://dx.doi.org/10.1155/2013/108103.

[43] V. Sklyarov, I. Skliarova, High-performance implementation of regular and easily scalable sorting networks on an FPGA, Microprocess. Microsyst. 38 (5) (2014) 470–484, http://dx.doi.org/10.1016/j.micpro.2014.03.003.

[44] M. Trovato, for the ATLAS and CMS Collaborations, Track trigger at the high luminosity LHC, PoS LHCP 2018 (2018) 259.

[45] T. Flick, for the ATLAS Collaboration, The phase II ATLAS pixel upgrade: the inner tracker (ITk), JINST 12 (01) (2017) C01098.

[46] D. Contardo, A. Ball, for the CMS Collaboration, The Phase-2 Upgrade of the CMS Tracker, CERN-LHCC-2017-009, CMS-TDR-014.

[47] We work in units where speed of light is set to unity, thereby equivalencing the units of momentum and energy.

[48] R. Mueller, J. Teubner, G. Alonso, Sorting networks on FPGAs, VLDB J. 21 (2012) 1, http://dx.doi.org/10.1007/s00778-011-0232-z.

[49] S. Campana, et al., An ATLAS distributed computing architecture for HL-LHC, J. Phys. Conf. Ser. 1085 (2018) 032029.