PROCEEDINGS OF SCIENCE

Artificial retina tracking at the LHC crossing rate

D. Tonelli[*†]
CERN, CH-1211 Geneva 23, Switzerland
E-mail: diego.tonelli@cern.ch

A. Abba, M. Citterio, F. Caponio, A. Cusimano, A. Geraci, and N. Neri

Politecnico and INFN, Milano, Via Celoria 16, 20133 Milano, Italy

F. Bedeschi, P. Marino, M.J. Morello, G. Punzi, A. Piucci, F. Spinella, and S. Stracka

University, Scuola Normale Superiore, and INFN Pisa, Largo Pontecorvo 3, 56127 Pisa, Italy

L. Ristori

Fermilab, PO Box 500, 60510 Batavia, IL, U.S.A.

We present results of an R&D study for a specialized processor capable of precisely reconstructing, in pixel detectors, hundreds of charged-particle tracks from high-energy collisions at 40 MHz rate. We use an highly parallel pattern-recognition algorithm, inspired by studies of the processing of visual images by the brain as it happens in nature. We describe an efficient hardware implementation in high-speed, high-bandwidth FPGA devices and show the resulting simulated tracking performance. This is the first detailed demonstration of reconstruction of offline-quality tracks at 40 MHz and makes the device suitable for processing Large Hadron Collider events at the full crossing frequency.

Technology and Instrumentation in Particle Physics 2014,
2-6 June, 2014
Amsterdam, the Netherlands

[*]Speaker.
[†]Corresponding author.

PoS(TIPP2014)187

1. Introduction

Charged-particle trajectories (tracks) are the most physics-rich quantities typically available in collider experiments. Tracks encapsulate kinematic, lifetime, and charge information in few parameters, which are usually measured accurately, owing to the high precision of position-sensitive detectors. Tracks are therefore attractive to discriminate in real time the $10^{-3} - 10^{-5}$ fraction of events that are typically stored for further processing in high-rate hadron collisions. However, real-time track reconstruction at high rates is a major challenge, which requires doing pattern recognition in a large combinatorics environment and handling a large information flow. This calls for highly parallel algorithms that only use the minimal subset of information needed to reconstruct tracks efficiently. Dedicated devices have been developed since the early '80s [1]. In the '90s, the Collider Detector at Fermilab (CDF) used pattern-matching algorithms implemented into field-programmable gate-arrays (FPGA) to reconstruct two-dimensional tracks from clusters of aligned hits in a large drift-chamber [2]. In 2001 the silicon vertex trigger [3] implemented fast and efficient pattern-matching using a custom-made processor, the associative memory, that connected the drift-chamber tracks with silicon-detector information and made available two-dimensional tracks with offline-like resolution at 30–100 Hz within 20 μs latency. Track triggers, based on content-addressable memories, were also used in the less demanding environment of proton-electron collisions [4]. Real-time track reconstruction would greatly benefit experiments at the Large Hadron Collider (LHC). As of year 2020, higher LHC energies and luminosities will challenge severely the experiments' data acquisition and event reconstruction capabilities. In addition, the large number of interactions per bunch crossing, and the increased event complexity will reduce the discriminating power of usual experimental signatures, such as charged leptons with a large momentum transverse to the beam (transverse momentum, p_T) or significant unbalances in total event p_T.

We realize a detailed bit-wise simulation of an implementation into FPGAs of a novel neurobiology-inspired pattern-recognition algorithm, the *artificial retina*, which proves particularly suited for real-time tracking in high-luminosity LHC conditions, and use a higher-level simulation to assess the tracking performance in the conditions of the upgraded LHCb experiment.

2. Artificial retina tracking

The artificial retina tracking algorithm [5] was inspired by the understanding of the mechanism of visual receptive fields in the mammals' eye [6], whose functionalities have recently been shown to mirror those employed in high-speed digital data reduction [7]. Each neuron dedicated to vision is tuned to recognize a specific simple shape on a specific region of the mammals' retina, the *receptive field*. The neuron response intensity to a visual stimulus is proportional to the degree of similarity between the shape of the stimulus and the shape for which the neuron is tuned to. Hence, each neuron reacts to the stimulus with different intensity. The brain extracts the first higher-resolution information on the basic geometric features of the stimulus by interpolation between the neuron responses, within a time of approximately 30 ms in humans. For a typical neuron firing frequency of 1 kHz, this corresponds to approximately 30 processing cycles. At clock frequencies of 1 GHz, this approximates the number of cycles/event required for achieving pattern recognition

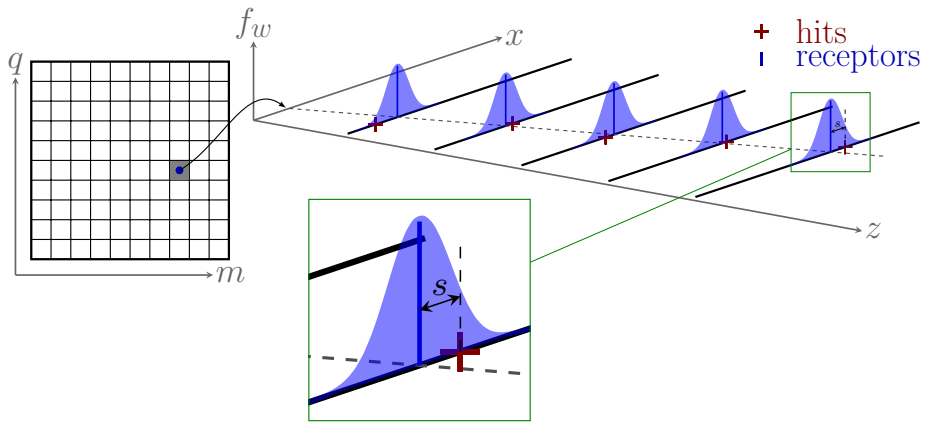

Figure 1: Schematic representation of the detector mapping. The parameter space (left panel) is discretized into cells; to each cell corresponds a track that intercepts the detector layers in a determined sequence of receptors (right panel).

at 40 MHz, and corresponds to a $\mathcal{O}(100)$ increase in processing efficiency over what attained by present or foreseen devices [3, 8] and about a $\mathcal{O}(10^7)$ increase over generic CPU architectures.

The concept is best understood using a simple example: a detector consisting of parallel layers of position-sensitive sensors that only measure one spatial coordinate, x. The trajectories of charged particles in the absence of magnetic field are straight lines identified by their angular coefficient m, and intercept q with the x axis in an arbitrary (z,x) plane. We discretize the parameter space into *cells* that mirror the visual receptive fields. The center of each cell identifies uniquely an ideal track in the detector space that intersects detector layers in spatial points called *receptors*. Therefore the parameter-space cell (m_i, q_j) maps into the set of receptors x_k^{ij}, where $k = 1,\ldots,n$ runs over the detector layers (figure 1). This cell-receptors mapping is done for all cells of the track parameter space. Once the receptors corresponding to all cells are known, the detector can be exposed to real tracks. The distance $s_{ijkr} = \bar{x}_{k,r} - x_k^{ij}$ of the receptors from the observed hits is computed and the response of the (m_i, q_j) retina-cell is calculated, $R_{ij} = \sum_{k,r} \exp\left(-s_{ijkr}^2/2\sigma^2\right)$, where $\bar{x}_{k,r}$ are the coordinates of the rth hit on the detector layer k, while σ is a parameter of the algorithm. R_{ij} represents the excitation of the receptive field. The total response of the retina is obtained by calculating the excitation R of all cells. Tracks are identified by local maxima among cells excited over a suitable threshold (figure 2).

In two dimensions, the algorithm bears analogies with the Hough transformation [9]. Generalization to multiple dimensions, presence of magnetic field, and so forth is conceptually straightforward [10]. After the track finding, determination of track parameters is refined using the excitation centroid of the nearest cells around each local maximum. This, along with the information contained in the smooth pattern-recognition response, recovers the resolution degraded by the discretization of the parameter space and allows for coarse retina granularities with no penalty in performance. The total number of cells is mainly driven by the capability of separating similar tracks. The retina's continuous, analog-like response and intrinsic capability of parallel processing down to hit level offer significant additional advantages.

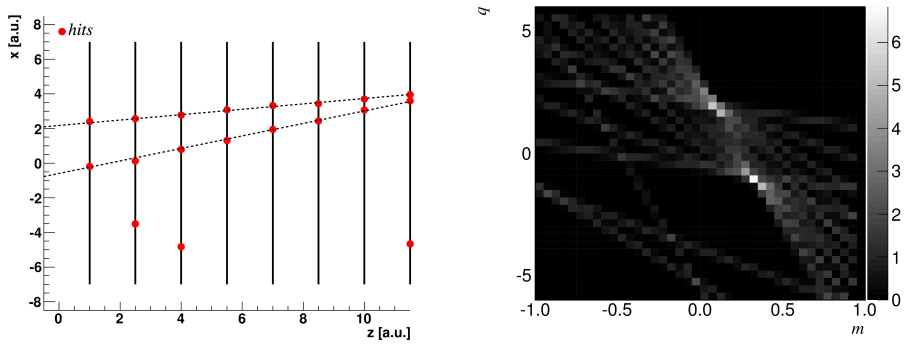

Figure 2: Sketch of a simple event containing only two tracks and a few noise hits (left panel) and response of the retina (right panel).

3. Implementation

We implement the algorithm and simulate its performance in realistic conditions using the LHCb detector upgraded for the 2020 high-luminosity operations as a use case [10]. The upgraded LHCb detector [11] is a single-arm forward spectrometer covering polar angles from 0.8° to 15.4° from the beam. The detector is designed to study particles containing bottom or charm quarks produced in 14 TeV proton-proton collisions occurring every 25 ns at luminosities of $2\text{–}3\times10^{33}$ $\text{cm}^{-2}\text{s}^{-1}$, with 7.6–11.4 average interactions per crossing. We use the standard LHCb upgrade simulation, which includes detailed modeling of a number of experimental effects, including multiple scattering, detector noise and so forth. The goal is to reconstruct, every 25 ns, 100–500 tracks, each associated to 10–20 hits that are scattered over 10–20 parallel detector planes. We choose an implementation that uses information from a small-angle telescope of ten tracking planes made of (i) eight layers of microvertex silicon detectors based on pixel technology [12] and installed in a volume free of magnetic field and (ii) two additional layers of microstrip silicon detectors located upstream of the magnet and immersed in its 0.05 T fringe field (Fig 3) [13]. Such telescope has about 50 mrad of angular acceptance and covers about half of the total tracking acceptance. Complementing it with an additional, similarly sized telescope at large angles covers the full acceptance. The logic is implemented in VHDL language; detailed logic-gate placement and simulation on the high-bandwidth Altera Stratix V device, model 5SGXEA7N2F45C2ES, is achieved using Altera's proprietary Quartus II software.

The implementation of the algorithm poses two chief conceptual and technological challenges. One is achieving an efficient distribution of the detector-hit information to the processing engines that calculate the excitations; the required 40 MHz throughput with several Tbit/s of input data flow makes this a nontrivial task. The other challenge is performing pattern recognition quickly enough to remain within latency constraints. Solutions to either issues typically depend on the geometry of the tracking layout. An efficient option exploits the LHCb geometry, in which straight-line tracks traverse the vertex detector before being curved by the magnetic field and reach the downstream tracking stations. The idea is that tracking performance sufficient for triggering can be achieved by restricting the core of the pattern-recognition task to a region where the magnetic field is weak. This greatly simplifies the switching and the pattern recognition tasks.

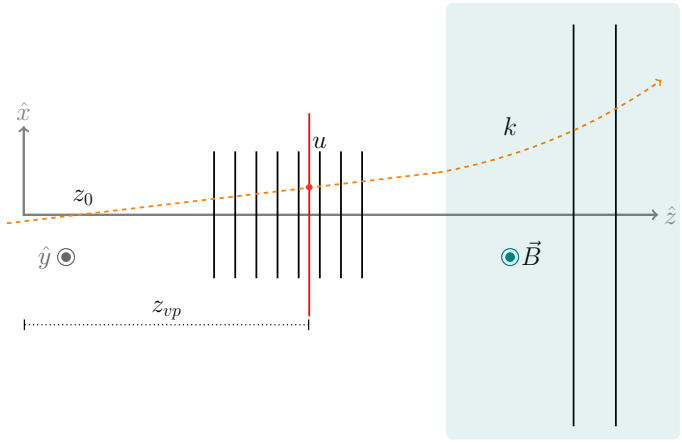

Figure 3: Longitudinal view of the tracker volume layout in (not to scale). The dashed line indicates a track; the parallel black lines indicate planes of silicon detectors used in the retina algorithm; the red line indicates the primary (u, v) plane for track finding.

In this geometry, the switching is facilitated because regions of contiguous detector-hits map into contiguous regions in track-parameter space, which have limited overlap. Hence, a mapping between hits and the parameters of possible tracks is obtained using simulation and results are used to associate a *zip-code* with each possible hit. The zip-code is used by the nodes of the switching network to properly route each hit thus avoiding the need to feed all hits to all receptors.

The LHCb geometry allows a factorization of the five-dimensional pattern recognition task into two separate and simpler steps, independently of nonuniformities of the magnetic field or detector misalignments. First, tracks are assumed to be straight lines originated from a single nominal interaction point and track-finding is performed in a two-dimensional *primary* plane transverse to the beam, whose intersection which each track identifies the track's two primary cartesian parameters u and v. Then, the determination of the momentum p and the coordinates of the origin of the trajectory d and z are treated as small perturbations of the primary two-dimensional track. We divide the primary parameter space (u, v) in a fine grid of cells and, for each, allow for a small number of bins (lateral cells) in the three remaining track parameters. A track is first identified as a cluster over threshold in the primary plane and an estimate of all track parameters is then obtained by balancing the excitation found in the later cells for each compact dimension.

We design an intelligent information-delivery system that routes each hit in parallel to all and only those engines for which such hit is likely to contribute a significant weight. Each hit comes associated with a zip-code. At each stage of the switching network, each node reads the hit's zip-code and, based on a predetermined map loaded locally, routes the hit in parallel to the appropriate nodes (or engines) of the following stage. Each hit is dispatched and duplicated as prescribed by the map. For maximum efficiency, the switching logic is integrated in the same FPGA devices where the processing engines are hosted. The switch consists of a network of nodes, whose basic building blocks are two-way sorters, with two input and two output data streams. Left and right input data are merged and each incoming hit is dispatched to one or both outputs according to its zip-code. If a stall from downstream layers occurs, one or both input streams are held. Such elementary building blocks are combined to build the needed network topology, with

the required switching capability. A $N \times N$ network requires $N\log_2(N)/2$ elements. The modular structure allows scalability and reconfiguration of the system if necessary, and allows distributing the necessary addressing information over the whole network, storing the information only at the nodes where it is required. The switching network occupies approximately 10% of the available logic in the Stratix V and completes its processing in 30 clock cycles.

Each cell in parameter space is defined as a logic module, the engine. Each hit is defined as a 41 bits word encoding the hit's geometric coordinates, zip-code, and timestamp. The engine is implemented as a clocked pipeline (figure 4). The intersections x_k and y_k for each layer k are stored in a read-only memory. The layer identifier associated with each incoming hit selects the appropriate set of x_k and y_k coordinates that are subtracted from the observed hit's $\bar{x}$ and $\bar{y}$ coordinates. The outcomes are squared and summed, and the result R is rounded. A sigma function, common to all engines, is mapped into a (8×256)-bit lookup table. The rounded R is used as address to the lookup table. The outputs of the lookup table are accumulated for each hit of the event. The same hit is cycled seven times in the engine logic, once for the calculation of the excitation corresponding to the coordinates of the cell in the primary plane, and twice for each of the secondary track parameters, treated as perturbations. Hence, seven accumulators are defined for each cell. However, the excitation intensities contributed by each hit to all track parameters are computed in parallel, such that every engine is able to accept one hit every 20 ns approximately. Several variants of the architecture are tested, from simple cases in which hits arrival is time-ordered to more complex scenarios where simultaneous processing of up to 16 events is allowed. Once readout of an event is completed, a signaling word prompts each engine to share the content of its primary cell to the neighboring engines. All engines in parallel compare the excitation in their primary accumulator with the excitations received from the neighboring engines and raise a flag if they identify a local maximum. Then, the coordinates and intensities of local maxima and the intensities of the nearest neighbors are output for track parameter extraction.

Each Stratix V can host up to 900 engines, with approximately 25% of logic available for other uses, including 15% for switching and the logic for center-of-excitation calculation. Hence, a realistic retina-tracker based on a small-angle telescope is implemented with about 22 500 engines in about 32 chips, and a complete system of two-telescopes with about 50 000 engines in 64 chips. A Modelsim simulation yields a maximum clock frequency of 350 MHz.

The clustering logic is a minor expansion that looks at the maximum flag and, if not busy, requires from the engine the content of all accumulators *and* the content of central accumulators of neighboring engines. The center-of-excitation calculation is factorized into two separate processes. The calculation restricted to the primary track parameters u and v implies finding the center of mass of a 3×3 square; the calculation relative to the remaining track parameters (d, p, z) requires computing the center of mass of a $3 \times 3 \times 3$ cube. Only a subset of coordinates in each dimension is relevant for the final result; hence, the problem reduces to processing a smaller number of values. The operation for each coordinate is $u = u_0/d_k + (\sum_{i,j} u_i l_{i,j})/\sum l$, where u_0/d_k is a global translation that depends on the absolute position of the engine and is stored in a lookup table. Two distinct weights are simultaneously produced for (u, v) and (d, p, z), respectively. The computation of the center of excitation takes 11 clock cycles along with another 10 cycles for fanout with a logic that occupies a fraction not larger than 15% of the Stratix V. A single center-of-excitation unit can serve up to 12 engines, with the expected hit occupancies. The determination of the local maximum and

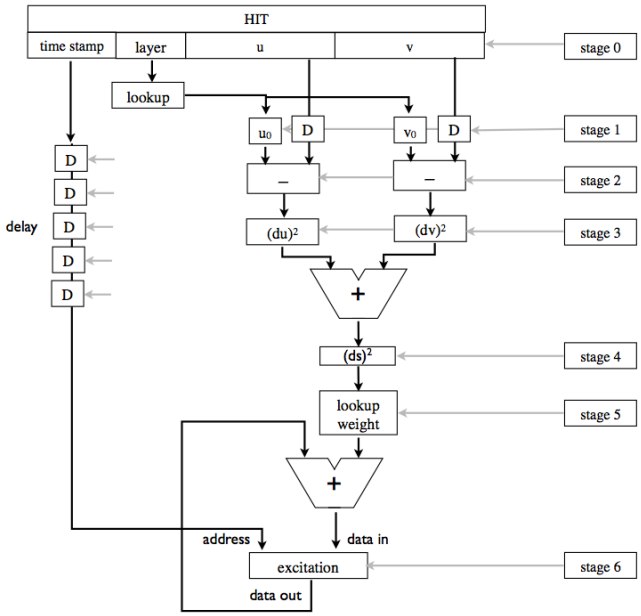

Figure 4: Functional architecture of a receptor-field engine.

center of excitation uses local copies of the accumulators so that the incoming hit flux is never stopped unless large time-fluctuations in the end-event signal occur. In these cases the incoming hits are kept on hold and stored in the switch trees.

4. Performance

Simulation shows that the device sustains an input frequency of 40 MHz of events, with the occupancy predicted by the full LHCb simulation, in the nominal luminosity conditions of the 2020 upgrade, 2×10^{33} cm^{-2}s^{-1}. Contributions to the total latency of 120 Stratix V clock-cycles are dominated by engine processing (60%), followed by switching (25%), and clustering and I/O (8% each). With the attained clock frequencies of up to 350 MHz, tracks are reconstructed with less less than 0.5 μs latency, which is likely to be negligible compared with other latencies typically present in the DAQ. Hence, the response is effectively *immediate* and makes tracks available right after the tracking detectors have been read out. Tests simulating higher track-densities show that the logic needed increases approximately linearly with the average number of hits per event.

Tracking performance is determined using a detailed high-level C++ simulation of the algorithm, which interfaces with the LHCb simulation. All hits from the chosen layers are processed by the retina, and are 880 (1220) on average at $L = 2(3) \times 10^{33}$cm^{-2}s^{-1} resulting in approximately 1.3 (1.95) hits per engine, which are used to form 121 (223) retina clusters over threshold. We restrict to *reconstructable* tracks. These have polar separation of $\theta < 50$ mrad from the beam, which approximates the angular coverage of the chosen layer configuration; are associated with at least three (two) hits on the relevant pixel (microstrip) layers; and have momentum $p > 3\,\mathrm{GeV}/c$ and transverse momentum $p_T > 200\,\mathrm{MeV}/c$. Efficiencies are the fractions of reconstructable tracks

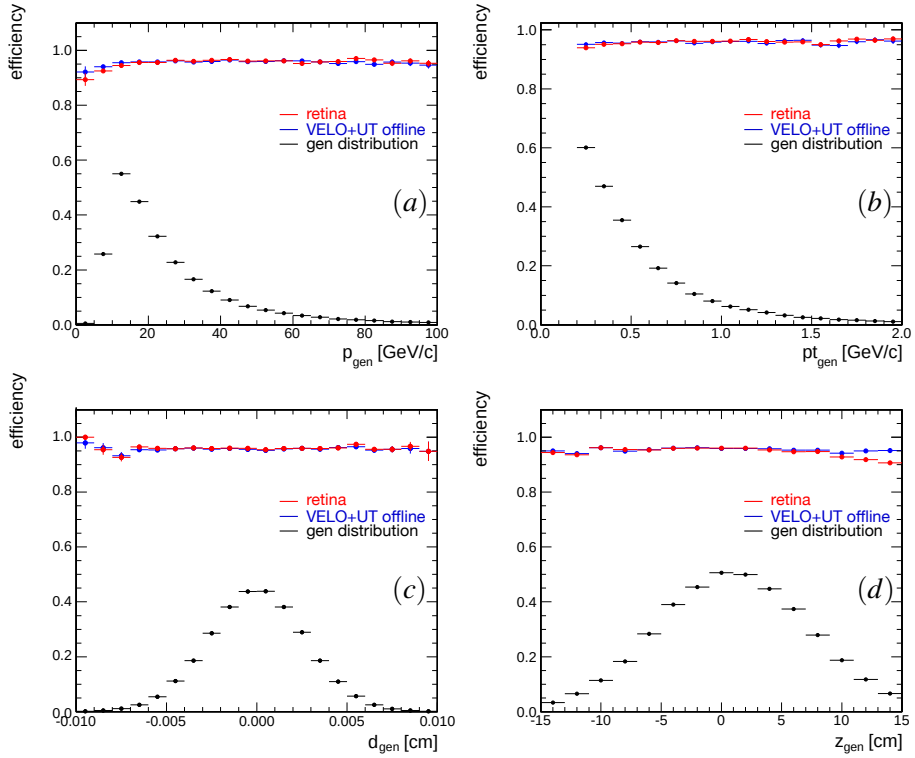

Figure 5: Tracking reconstruction efficiencies of the retina algorithm (in red) and of the offline algorithm (in blue) for simulated generic events at $L = 3 \times 10^{33}\,\mathrm{cm}^{-2}\mathrm{s}^{-1}$ events, as functions of (a) p, (b) p_T, (c) d, (d) z. The distribution of the considered parameter is also reported, in black.

that are reconstructed. Efficiencies are shown in figure 5 as functions of p, p_T, d, and z and compared with the efficiencies of the offline LHCb tracking [10] Retina tracking efficiencies exceed 95% in generic events, which is comparable to the offline-tracking performance. The fraction of falsely reconstructed tracks is 8% (12%) at $L = 2(3) \times 10^{33}\,\mathrm{cm}^{-2}\mathrm{s}^{-1}$, marginally higher than that of the offline algorithm. Efficiencies in excess of 97% are determined for recostructing tracks from benchmark signal modes, such as $B_s^0 \to \phi\phi$, $D^{*\pm} \to D^0\pi^\pm$ and $B^0 \to K^*\mu\mu$.

Resolutions on retina-track parameters are comparable to those from the offline algorithm, taking into account the differences in layer configurations. The resolution of the retina-track curvature is a factor of 1.25 worse than that of the offline, as expected due to the lack of the stereo-layer information in the tested implementation of the system.

5. Summary

We report the first realistic implementation of the artificial-retina tracking algorithm. We simulate the algorithm in high-end FPGA processors and determine performances under the conditions expected for the 2020 upgrade of the LHCb detector as a benchmark. Offline-grade tracks are reconstructed within a latency of less than 0.5 μs making the processor appear to DAQ as an additional detector that directly outputs tracks. This is 400 times faster than any existing or foreseen

HEP device and provides the first demonstration of online reconstruction of offline-quality tracks at 40 MHz.

References

[1] C. Daum *et al.*, *Online event selection with the Famp microprocessor system*, *Nucl. Instrum. Methods A* **217** (1983) 361.

[2] G. Foster *et al.*, *A Fast Hardware Track Finder for the CDF Central Tracking Chamber*, *Nucl. Instrum. Methods A* **269** (1988) 93; E.J. Thomson *et al.*, *Online track processor for the CDF upgrade*, *IEEE Trans. Nucl. Sci.* **49** (2002) 1063.

[3] M. Dell'Orso and L. Ristori, *VLSI structures for track finding*, *Nucl. Instrum. Methods* **278** (1989) 436; F. Morsani *et al.*, *The AMchip: a VLSI associative memory for track finding*, *Nucl. Instrum. Methods* **315** (1992) 46; B. Ashmanskas *et al.*, *The CDF silicon vertex trigger*, *Nucl. Instrum. Methods A* 518 (2004) 532; G. Punzi and L. Ristori, *Triggering on Heavy Flavors at Hadron Colliders*, *Annu. Rev. Nucl. Part. Sci.* **60** (2010) 595.

[4] A. Baird *et al.*, *A Fast High-Resolution Track Trigger for the H1 Experiment*, *IEEE Trans. Nucl. Sci.* **48** (2001) 1276.

[5] L. Ristori, *An artificial retina for fast track finding*, *Nucl. Instrum. Methods A* **453** (2000) 425.

[6] D. Hubert and T. Wiesel, *Receptive felds of single neurones in the cat's striate cortex*, *J. Physiol.* **148** (1959), 574 and *Receptive fields, binocular interaction and functional architecture in the cat's visual cortex*, *J. Physiol.* **160** (1962), 106; N.J. Priebe and D. Ferster, *Mechanisms of neuronal computation in mammalian visual cortex*, *Neuron* **75** (2012) 194.

[7] M.M. Del Viva, G. Punzi, and D. Benedetti, *Information and perception of meaningful patterns*, *PLoS ONE* **8** 7 (2013) e69154.

[8] A. Andreani *et al.*, *The fasttracker real time processor and its impact on muon isolation, tau and b-jet online selections at ATLAS*, *Proceedings of the IEEE-NPSS Real Time conference*, (2010); M. Shocket *et al.*, *Fast tracker (FTK) technical design report*, *ATLAS-TDR-021* (2013).

[9] P.V.C. Hough, *Machine analysis of bubble chamber pictures*, *Conf. Proc.*, **C590914** (1959) and *Method and means for recognizing complex patterns*, US patent nr. 3069654 (1962).

[10] A. Abba *et al.*, *A specialized track processor for the LHCb upgrade*, *CERN-LHCb-PUB-2014-026* (2014).

[11] LHCb Collaboration *Framework TDR for the LHCb Upgrade*, *CERN-LHCC-2012-007* (2012).

[12] LHCb Collaboration, *LHCb VELO Upgrade Technical Design Report*, *LHCB-TDR-013* (2013).

[13] LHCb Collaboration, *LHCb Tracker Upgrade Technical Design Report*, *LHCB-TDR-015* (2014).