PROCEEDINGS OF SCIENCE

Variable resolution Associative Memory optimization and simulation for the ATLAS FastTracker project

A. Annovi[a], A. Castegnaro[a], P. Giannetti[b], Z. Jiang[c], C. Luongo[*b], C. Pandini[d], C. Roda[eb], M. Shochet[c], L. Tompkins[c], G. Volpi[a]

[a] *INFN Frascati, Via E. Fermi 40, 00044 Frascati, Roma, Italy*
[b] *INFN Pisa, Largo Bruno Pontecorvo 3, 56127 Pisa, Italy*
[c] *University of Chicago, 5801 S Ellis Ave Chicago, IL 60637, United States*
[d] *Università di Milano, Via Conservatorio 7, 20122 Milano, Italy*
[e] *Università di Pisa, Largo B. Pontecorvo 3, 56127 Pisa, Italy*
E-mail: alberto.annovi@lnf.infn.it, andrea.castegnaro@lnf.infn.it, paola.giannetti@pi.infn.it, zihao.jiang@cern.ch, carmela.luongo@pi.infn.it, carloenrico.pandini@studenti.unimi.it, chiara.roda@cern.ch, shochet@hep.uchicago.edu, lauren.a.tompkins@gmail.com, guido.volpi@lnf.infn.it

ATLAS is planning to use a hardware processor, the Fast Tracker (FTK), to perform on-line track reconstruction at the level−1 event output rate (100 kHz). The processor can perform this task using a very large number of precalculated track patterns stored in a dedicated bank and extracting the ones compatible with a given event.
In order to obtain a better trade off between the number of patterns and the number of fits needed to complete tracking after the pattern matching, ATLAS FTK exploits variable resolution pattern matching, carried out by a dedicated device, called Associative Memory (AM) chip, that includes ternary logic, inspired from ternary commercial CAMs. This architecture is able to store the variable resolution feature that tunes the precision of the match for each pattern and for each detector layer, allowing the patterns to be of variable shape.
We have studied different methods of building the pattern bank exploiting different resolution pattern matchings. We show how this new feature achieves the goal of having few enough patterns to fit in the hardware, while maintaining good efficiency and the required rejection against random combinations of hits. We finally present a detailed preliminary study showing that the application of some degrees of variable resolution makes possible to build a bank that will allow the system to be fully functional at the luminosities and pileup conditions expected for the LHC after Phase-I upgrades.

11th International Conference on Large Scale Applications and Radiation Hardness of Semiconductor Detectors
3-5 July 2013
Auditorium Cassa di Risparmio di Firenze, via Folco Portinari 5, Florence, Italy

[*] Speaker.

1. Introduction

ATLAS is one of the two general purpose detectors designed to measure the product of the proton-proton collisions at the Large Hadron Collider (LHC) located at the CERN laboratory in Geneva. One of the characteristic feature of the hadron colliders is that the most interesting processes are rare and hidden under an extremely large background. Only a very limited fraction of the produced data can be transferred to offline storage, so it is necessary a multi-level trigger [1] to perform a large real-time data reduction.

On-line track reconstruction is a very important aspect in triggering at CERN Large Hadron Collider, especially during the high luminosity phases. The tracking, ~~indeed,~~ allows to distinguish the tracks produced from the main proton-proton collisions from the ones produced in the many overlaying collisions (pile-up). The Fast Tracker (FTK) system will perform this task for the ATLAS trigger [2].

The FTK hardware processor is an evolution of the CDF Silicon Vertex Tracker (SVT) [3, 4], where tracks inside very high multiplicity events must be found in a time span of a few microseconds and the addition of tracking to the hardware trigger has been shown to be an effective method of improving trigger. Trigger di Livello 1

The FTK system will operate at full level-1 output rates (100kHz) and will provide global track reconstruction over the whole inner silicon detector at the input of the level-2 trigger. The near-offline-quality tracks and increased available execution time will improve the downstream trigger algorithms, improving signal efficiency and background rejection.

Nuovo paragrafo con titolo

The FTK algorithm consists of two sequential steps. In the first step pattern recognition is carried out by a dedicated device called Associative Memory (AM) [5] which finds coarse-resolution track candidates named roads. The AM is a massively parallel system that simultaneously compares the incoming data (the event hits) with all the stored precalculated low resolution track patterns and returns the addresses of the matching patterns. In the second step, another processor (the Track Fitter) receives these matching patterns and fits the full-resolution hits inside the road to determine the track parameters. Only the tracks passing the χ^2 cut are kept.

A critical figure of merit for the AM-based track reconstruction system is the number of patterns that can be stored in the data bank. The FTK processor proposed for the ATLAS experiment is going to perform at a much higher luminosity ($> 10^{34} cm^{-2} s^{-1}$) than the Tevatron SVT, and this will increase the complexity of the events. As a consequence, a very large bank is necessary: candidate tracks have to be found with more than 95% efficiency over the whole inner detector, and the pattern recognition has to be extended to 12 silicon detector layers[1] with a small enough pattern width to obtain a very high rejection of fake tracks in order to drastically reduce the track fitting processing time. We include in the AM chip the variable resolution feature, that is the ability to employ fine pattern resolution only when necessary. In this way the shape of each pattern can be optimized to improve the acceptance for valid tracks with maximum fake rejection. Therefore this feature allows the use of a small AM bank, while profiting from the positive effects of high spatial resolution in pattern matching.

Spostare

The geometry of the ATLAS inner detector and the characteristics of the pattern bank used to perform the first step of the pattern recognition in the FTK algorithm are described in Section 2.

[1] 8 layers are used in the AM pattern matching. All 12 layers are used in track fitting.

The concept of variable resolution applied to the pattern matching is treated in Section 3, while the results of the simulations using this new pattern bank feature are discussed in Section 4.

2. The pattern bank for the ATLAS inner detector

silicon

The detector is segmented in layers placed at different radial distance from the beam axis. Each layer is in turn divided into bins of equal size, each one is a rectangle in $\phi - z$ space for barrel and $\phi - r$ space for endcaps.[2] For each event a number of particle tracks traverse the detector and each track crosses one bin per layer, generating hits. A pattern is a sequence of Super Strips (SS), one per layer, where the Super Strip is the logical OR of adjacent bins. Each track generates a hit pattern. The collection of all these patterns defines both the space of the tracks we are looking for and how they appear in the detector: this collection is the pattern bank.

As already mentioned, the pattern recognition step finds the roads which will be sent to the successive track fitting process. Here the critical parameter that must be optimized is the road width. If the roads are too wide, the load on the Track Fitter can become excessive due to the large number of uncorrelated hits within a road (fake roads). If the roads are too narrow, the needed size of the AM becomes too large and hence the cost becomes excessive. Therefore the trade off that we have to face is between the number of patterns that can be stored in the data bank and the number of fits that the Track Fitter has to perform.

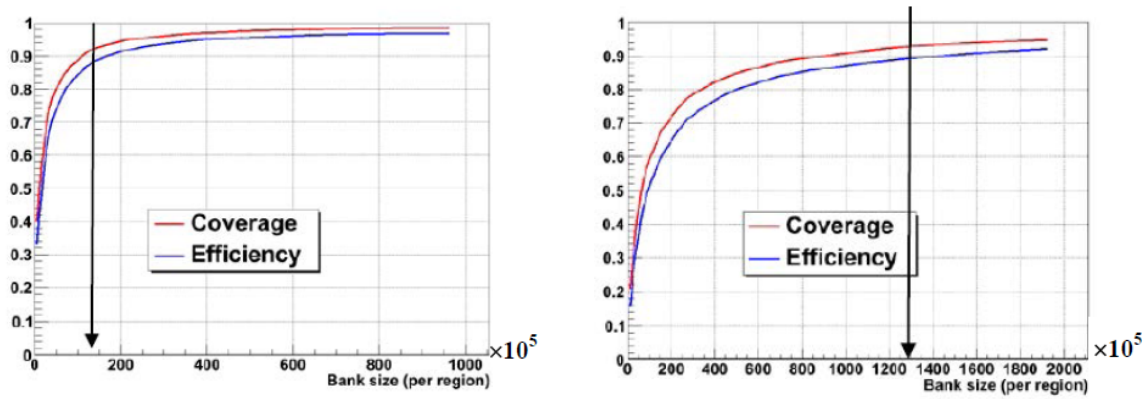
Therefore the quality of the AM bank can be characterized by two variables: the coverage and the road fake rate. The coverage is the fraction of reconstructable tracks based only on the detector geometry; it describes only the geometric efficiency of the bank. Track efficiency, instead, includes the contributions of all the algorithms used in FTK and the track fitting inside roads.

We generate tracks in the whole detector and we store new patterns corresponding to the generated tracks until the bank reaches the desidered coverage. In principle, the pattern bank may contain all possible tracks that go through the detector (a 100% efficient bank). Actually, when effects such as detector resolution smearing, multiple scattering, etc. are considered a huge number of extremely improbable patterns arises that largely increase the pattern bank size. For this reason, we decide to use a partially inefficient bank and we store new patterns until the bank reaches this coverage.

To maximize the efficiency of the pattern bank for a given size, we order the pattern list by the number of training tracks that match a pattern. This number defines the coverage of the single pattern. This procedure automatically ensure that "high-coverage" patterns are stored while "low-coverage" patterns are left out.

Figure 1 shows typical curves of track coverage and track efficiency versus bank size with different road resolution. The coverage is determined by the bank only, while the track efficiency includes the contributions of all the algorithms used in FTK and the track fitting. In both cases, there is an initial rapid rise as the bank is filled with high-coverage patterns, patterns that match tracks with a high probability, and then the curves rise slowly as low-coverage patterns, less probable

[2]The ATLAS reference system is a cartesian right-handed coordinate system, with the nominal collision point at the origin. The azimuthal angle (radians) is measured around the beam axis, and the polar angle θ is measured with respect to the z-axis. The pseudorapidity is defined as $|\eta| = -ln(tan\theta/2)$ and p_T is the track momentum transverse to the beam direction.



(a) Low resolution

(b) High resolution

Figure 1: The coverage and the efficiency of the AM bank versus bank size per region. (a) The SS sizes are: 22 pixels in $r-\phi$, 36 pixels in z and 16 strips in SCT. (b) The pattern width is reduced by a factor of 2 along the ϕ direction, so the SS sizes are: 11 pixels in $r-\phi$, 36 pixels in z and 8 strips in SCT. The arrows show the chosen operating point. A region corresponds to 45 degrees in ϕ of the detector.

ones, are added to the bank. Although the efficiency for real tracks grows slowly in this region, the number of fake matched roads increases nearly linearly with the bank size. Thus the number of patterns stored in the AM bank has to satisfies the need for a high efficiency and, at the same time, the need to limit the rate of fake roads. Figure 1a shows that in the case of low resolution roads, we need ~12 millions patterns to reach coverage and efficiency of 90%. Therefore the pattern bank is about ten times smaller than the high resolution pattern bank where ~120 millions patterns are needed to reach the same coverage and efficiency (Figure 1b). Thus reducing the pattern bank width by a factor of 2 just in the ϕ direction requires a bank a factor of 10 larger to provide a similar efficiency. Unfortunately, the price of the first solution is that the number of fake matched patterns is about ten times bigger, due to the large number of uncorrelated hits within a larger road. The challenge is therefore to obtain a solution that allows a small AM system and, at the same time, a low fake road rates to reduce the Track Fitter workload.

3. Variable Resolution Patterns

Section

As discussed in the previous section it is important to find a compromise between the number of patterns stored in the AM bank and the number of fits to be executed by the Track Fitter. The solution that is proposed is based on the use of "variable resolution patterns".

shows

Figure 2 shows a diagram that illustrates how this solution works. The drawing on the left shows an example of three typical tracks crossing seven detector layers. The number of bins in each layer indicates the SS resolution (1 bin for high resolution and 2 bins for low resolution). The tracks do not cross the white region inside the pattern, so this subregion of the pattern is not necessary, in fact it increases the combinatorial and the probability of uncorrelated hits due to noise. The fixed resolution approach provides two ways to store the patterns in the AM bank (shown in

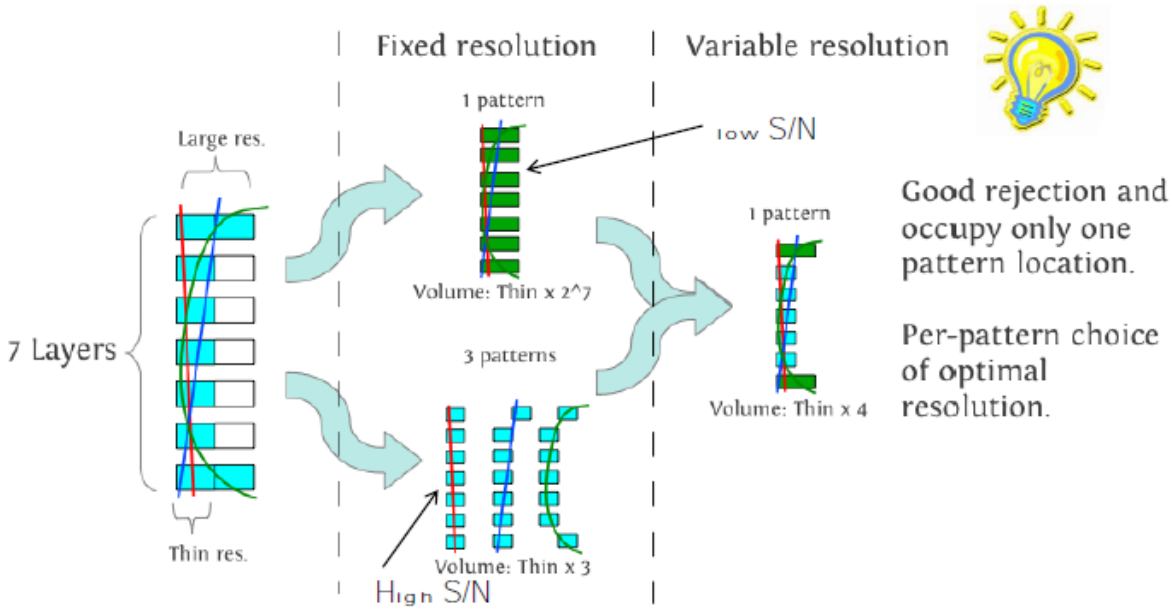


Figure 2: Diagram illustrating the variable resolution patterns.

the center of Figure 2). The first possibility (Figure 2, center top) is to use a single low resolution pattern stored in just one AM location. This fixed resolution pattern takes up little AM space, but its large Super Strips offer all their acceptance to the fakes (in the white bins not touched by any real track) The second possibility (Figure 2, center bottom) is to use three high resolution (thin) patterns. Of course the combinatorial is significantly reduced, in fact the three patterns fit perfectly the three tracks crossing the detector layers, but the price is the use of three AM locations. The best compromise is to use a single variable resolution pattern (Figure 2, right) minimizing the use of AM locations and the fake rate. The size of the patterns changes layer by layer and pattern by pattern. We divide the Super Strips in two parts: if both halves can be touched by real tracks (like the first and the latter layer of the example), the layer is used at low resolution applying the variable resolution feature; if, on the other hand, just one of the halves is compatible with a real track (the other layers in the figure), the layer is used at full resolution, reducing by a factor two the area that the SS offers to uncorrelated hits from other tracks. As a result we have, at the same time, low pattern bank size, low fake rate and high efficiency.

Figure 3 shows an advanced application of the variable resolution feature using more than one bit. On the left of Figure 3, no-variable resolution case, it is shown that three patterns are needed to accept the three tracks crossing the detector layers (the blue, the green and the pink patterns). In the middle it is shown the 1-bit variable resolution case: one pattern is enough to accept all the tracks, that is just one AM location is occupied. On the right there is an advanced application of this feature: 3-bit variable resolution. We need just one pattern, as the previous configuration, but less volume because the SS can be more thin thanks to the 3 bits. This solution increases the rejection of fakes roads and reduces the internal combinatorial produced by uncorrelated hits inside

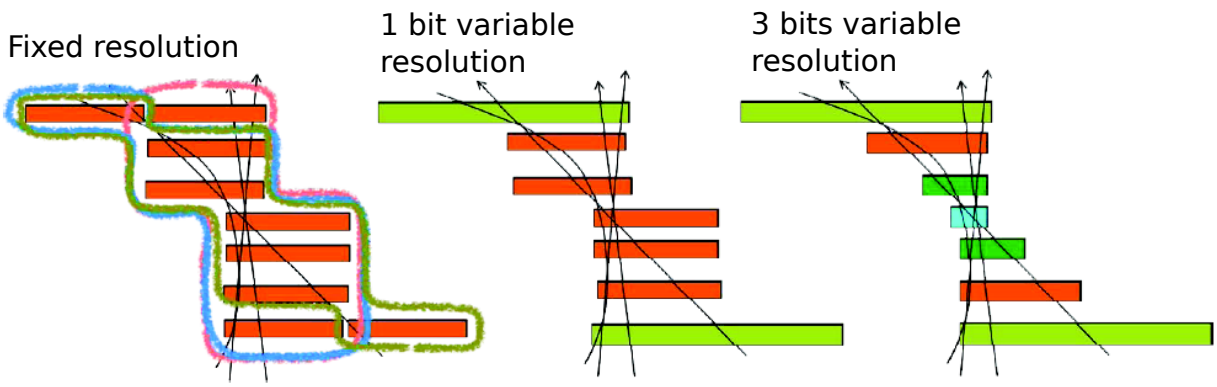


Figure 3: Diagram illustrating the use of many bits in the implementation of the variable resolution patterns: (left) fixed resolution patterns, (center) 1-bit variable resolution pattern, (right) 3-bit variable resolution pattern.

lager roads.

4. Simulation results

As discussed above, one of the main goals in the implementation of the AM pattern bank is to maximize the bank efficiency while reducing the AM bank size and the Track Fitter workload. This goal can be reached with the optimization of the use of the variable resolution patterns as discussed in this section.

The simulation program for FTK (FTKSim) processes complete ATLAS events and creates the same list of tracks that will be produced by the FTK hardware. The program, among other purposes, allows us to evaluate the crucial parameters needed for the hardware design, such as pattern bank size, number of roads, roads size and number of fits.

In order to sustain the level-1 trigger rate, it is necessary to organize FTK as a set of independent engines, each working on a different region of the silicon tracker. Therefore the detector is divided into 64 towers corresponding to 16 intervals in ϕ and 4 intervals in η. Each AM board has the following hardware limits [6]:

- $\#AMpatterns < 8M$[3]
- $< \#Roads/event > < 8 \times 10^3$
- $< \#Fits/event > < 40 \times 10^3$

We have concentrated the simulation studies on two scenarios that correspond to the expected LHC conditions in 2015 and 2019 respectively. In the 2015, assuming 46 pile-up events per trigger, FTK will be implemented using 16 boards handling 32 detector towers. In the 2019, assuming 69 pile-up

[3] $M = 1024 \times 1024$

events per trigger, FTK will be implemented using 128 boards managing 64 detector towers. The constraints that each tower must fulfill in the two scenarios are listed in Table 1.

	2015	2019
<#AM patterns/tower>	$4M$	$16M$
<#Roads/event>	4×10^3	16×10^3
<#Fits/event>	20×10^3	80×10^3

Table 1: Constraints for each tower in the two scenarios.

In the simulation we have studied different configurations of the variable resolution patterns. We start from one high resolution set of patterns, with a fixed SS size, and we try several different AM bank configurations. The first starting point (the narrowest road width) is a SS of size 15x36x16, where 15x36 is the number of pixels clustered in the SS ($\phi \times z$) and 16 is the number of strips clustered in the same SS (ϕ). Various configurations having different resolution have been studied and classified according to the track efficiency, number of matched roads and number of candidate tracks at the input of the Track Fitter. The track efficiency has been evaluated in a single muon dataset while the number of matched roads and the number of candidate tracks have been obtained using simulated events with 69 pile-up events. For example, if we apply the 1-bit variable resolution feature on both pixel and SCT layers, the maximum SS size is increased by a factor of two, from 15x36x16 to 30x72x32 AM bank configuration.

Option	Maximum Road Width	#AM $\cdot 10^6$	Efficiency(%) R=64	Roads/evt $\cdot 10^3$	Fits/evt $\cdot 10^3$
1.0	$(30x72)_{pix}x32_{sct}$	18	91.2	7.1	56
1.1	$(30x72)_{pix}x32_{sct}$	16.8	91.2	6.9	55
1.2	$(30x72)_{pix}x32_{sct}$	15	91.0	6.2	50
2.0	$(30x144)_{pix}x32_{sct}$	8	92.0	5	90
3.0	$(30x72)_{pix}x64_{sct}$	8	93.0	9	154

Table 2: Results in endcap towers for different road widths. The maximum road width is obtained applying the variable resolution feature to the 15x36x16 base AM bank configuration. #AM patterns, #Roads and #Fits are reported for one tower. #Roads and #Fits are evaluated with 2019 conditions (see text).

Table 2 and 3 summarize the most interesting results obtained from the FTK simulation for the endcap and the barrel regions, starting from the 15x36x16 AM bank configuration and applying the variable resolution feature with different configurations. The options are ordered by decreasing number of patterns. The maximum allowed is 16.8M, the number of memory locations on the two Associative Memory boards in a tower. Thus options 1.0 in the endcap and 1.0 and 1.1 in the barrel cannot be used. Option 1 uses 1-bit variable resolution on both pixel and SCT layers. Option 2 applies 1-bit variable resolution on the pixels in the ϕ direction and on the strips in SCT layers and 2-bit variable resolution on the pixels in the z direction. In the latter case the SS size can be as large as 30 pixels in the ϕ direction, 144 pixels in the z direction and 32 strips in SCT

Option	Maximum Road Width	#AM $\cdot 10^6$	Efficiency(%) R=64	Roads/evt $\cdot 10^3$	Fits/evt $\cdot 10^3$
1.0	$(30x72)_{pix}x32_{sct}$	21	94.8	3.9	33
1.1	$(30x72)_{pix}x32_{sct}$	18	94.1	3.4	28
1.2	$(30x72)_{pix}x32_{sct}$	16.8	93.3	3.2	26
2.0	$(30x144)_{pix}x32_{sct}$	8	95.0	4	60
3.0	$(30x72)_{pix}x64_{sct}$	8	96.0	6	98

Table 3: Results in barrel towers for different road widths. The maximum road width is obtained applying the variable resolution feature to the 15x36x16 base AM bank configuration. #AM patterns, #Roads and #Fits are reported for one tower. #Roads and #Fits are evaluated with 2019 conditions (see text).

layers. Option 3 uses 1-bit variable resolution feature on pixel layers and 2-bit variable resolution on SCT layers. The first three tables lines show that for a given variable resolution configuration, reducing the number of patterns reduces the number of roads and fits, while the efficiency is just minimally reduced. We can see in the fourth line of the table the power of the variable resolution pattern. Applying just one more variable resolution bit on the pixels, we get increased efficiecy and reduced number of roads with half size bank.

5. Conclusions

The variable resolution pattern matching allows the patterns to change in shape and matching volume improving the precision only where needed. It increases the rejection of fake roads, greatly reducing the number of roads out of the AM chip, and the compression factor in case of similar patterns, significantly reducing the number of patterns in the AM chip. Therefore the variable resolution pattern matching is an innovative idea that makes it possible to reduce the cost, size and complexity of FTK, in fact this feature allows setting the architecture parameters so that all hardware constraints are satisfied.

References

[1] W. Smith, *Triggering at LHC Experiments*, *Nucl. Instr. and Meth. A*, **vol.478**, pp.62-67, 2002

[2] A. Annovi, *Hadron Collider Triggers with High-Quality Tracking at Very High Event Rates*, *IEEE Trans. Nucl. Sci.*, **vol.51**, pp.391, 2004

[3] J. Adelman, *The Silicon Vertex Trigger upgrade at CDF*, *Nucl. Instr. and Meth. in Physics Research A*, **vol.572**, Issue 1, pp.361-364, 2007

[4] J. Adelman, *Real time secondary vertexing at CDF*, *Nucl. Instr. and Meth. in Physics Research A*, **vol.569**, pp.111-114, 2006

[5] M. Dell'Orso, L. Ristori, *VLSI Structures for Track Finding*, *Nucl. Instr. and Meth. A*, **vol.278**, pp.436, 1989

[6] `https://cds.cern.ch/record/1552352/files/ATL-COM-DAQ-2013-041.pdf`