Survey of Online Tracking Algorithms

Sean Dobbs, A. Tomaradze

(Northwestern University)

M. Mertens

(Forschungszentrum Jülich GmbH)

Introduction

- Efficient online tracking algorithms are essential for triggering on physics events.
- Marius asked us to survey the algorithms used by other experiments as a starting point for STT online tracking.
- Our search concentrated on experiments
    - that ran in the past decade,
    - had cylindrically symmetric geometry (e.g. not LHCb)
    - had wire chamber-like main tracking system

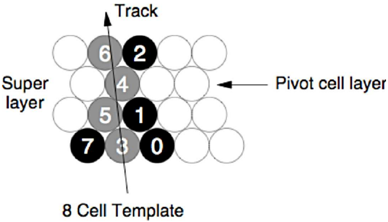
- Caveat: Details were not always easy to find or compare between different experiments, and often changed during the course of the experiment.

Tracking Algorithms

- Generally, there were two categories of track finding algorithms:
  - “local”’: track/road following, Kalman filter, etc.
  - “global”: Hough transform, Histogramming, etc.
- Three different levels of triggers are seen:
  - Level 1: fast track finding with specialized hardware (FPGAs)
  - Level 2: fast readout hardware for simplified reconstruction algorithms running on commodity hardware
  - Level 3: offline–quality reconstruction, with simplified calibrations and/or geometry
- Improvements in processor and network speed have lead to move away from L2 triggers and towards smarter L1 triggers with full event reconstruction in L3.
- N.B.: Algorithms are highly optimized for their specific detectors.

Template Matching

- Find track segments ("tracklets") in subset of detector ("superlayer") using large, fast associative memory banks in modern FPGAs
- Patterns based on realistic tracks, allowing for the possibility of missing hits. Can include patterns from tracks with displaced vertices.



- Example: BaBar matches eight–cell patterns that "pivot" around cell 4, with hits required in either four or three layers.

Track Following

- Start with initial track segment on inside or outside of detector
- Build track by extrapolating from initial segment, adding hits along predicted path
- Example: BaBar starts with track segments from inner superlayer, and builds outwards allowing one or two (in certain circumstances) superlayers to be missing.

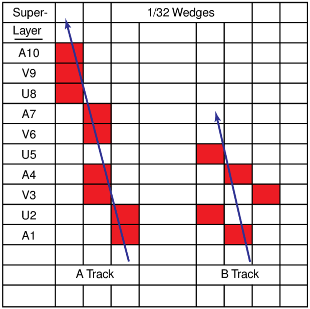


Figure 3: Track Linker Algorithm: two example tracks. The long track on the left side shows a segment hit pattern for an "A" track for which the segment hit corresponding to superlayer U5 is missing. The short track on the right side illustrates the stereo-wire rotation effect of a track with significant inclination, a track that is far from normal to the z-axis.

Experiment Parameters

	event rate	trigger rate (L1/(L2)/L3)	avg. track multi.	layers	cell size (mm)	trigger efficiency
e^+e^- Experiments						
CLEO III	250kHz	$<$ 1kHz/130Hz	~ 8 $(B\overline{B})$ 2 (e^+e^-)	47	7	$\sim$ 99%
BaBar	2kHz	970Hz/120Hz		40	$6-8$	$\sim$ 94%
Belle	5kHz	500Hz/500Hz		50	$8-10$	$>$ 90%
BES–III	$\sim$3kHz	$>$ 4kHz/1kHz	~ 4	43	$6-8$	$\sim$ 99%
ep Experiments						
ZEUS	$\sim$1MHz	600Hz/100Hz/20Hz	~ 10	72	~ 25	$\sim$ 70–90%
H1		1kHz/200Hz/50Hz/$\sim$5Hz		56	23–43	
$pp + p\bar{p}$ Experiments						
CDF	7.5MHz	30kHz/750Hz/75Hz	~ 35	96	8.8	96%
DØ		10kHz/1.5kHz/50Hz		32	0.4	$\sim$ 95%
CMS	$\leq$40MHz	100kHz/100Hz	> 100	~ 12	—	85–98%
ATLAS		100kHz/2kHz/200Hz		36	2	$>$ 90%
PANDA	$\sim$20MHz		$\sim 4-6$	24	5	

L1 Track Finding Algorithms

CLEO	templates for 16 axial layers, 8 stereo 4–layer superlayers stereo track “roads” matched, correlated to axial tracks
BaBar	$r - \phi$: tracklets found using templates for 8–cell groups in 4–layer superlayers, track following using 32 ϕ and 10 radial sectors z: Hough transform using 8 ϕ and 10 radial bins, followed by 2 χ^2 fits
Belle	$r - \phi$: tracklets found using templates for 5/6–layer superlayers track following using 64 wedges in ϕ and 6 radial sectors z: templates using 4 superlayers and 3 cathode layers in 8 ϕ sectors
BES-III	BaBar–style tracklet finding + track following in 4 superlayers (3 inner, 1 outer)

L1 Track Finding Algorithms

CDF	tracklet finding in 4 axial 12–layer superlayers, road finding in 288 $\phi-$ slices, both with templates
DØ	templates for 8 double layers in 80 $\phi-$ slices
ZEUS	templates for 3 axial 8–layer superlayers
H1	L1: tracklet finding in 4 3–layer superlayers, histogram track finder L2: finer histogram $+ \chi^2$ fit
CMS & ATLAS	No hardware–based track finding

L1 Processing Hardware

CLEO	axial: 32 Xilinx 5202, 16 Altera 7084
	stereo: 60 Altera 8820, 60 Altera 7128
BaBar	Xilinx Virtex 2: 72 axial, 48 stereo
Belle	1024 track segment finder, 64 track finder (Xilinx?)
BES III	Xilinx Virtex 2
CDF	Altera Flex 10k: 336 Track Finder, 288 Track Linker
DØ	160 Xilinx Virtex 2

Online Algorithms

	detector	algorithm
CLEO	DR	L1: lookup table (full inner + four–layer outer) + road following
		L3: 2D χ^2 circle fit
BaBar	DR	L1: four–layer tracklet finding + road following
		L3: lookup table + fast Kalman fit
Belle	DR	L1: 5/6–layer tracklet lookup table + combinatorial wedge finder
		L3: conformal transform χ^2 fit
BES-III	DR	L1: 4–layer tracklet lookup + road following
		L3: Kalman fit
ZEUS	DR	L1: tracklet finding/matching in $r-\phi$ and $z-r$
		L2: Road following + $r-\phi$ χ^2 circle fit + z info
		L3: Kalman fit
H1	DR	L1: tracklet finding/matching in 4×3 axial layers
		L2: 2D χ^2 circle fit in $r-\phi$ and $r-z$
		L3: none
		L4: Kalman fit?

DR: Drift Chamber

Online Algorithms

CDF	DR	L1: 4 layer tracklet lookup + road finding in axial superlayers
		L2: add in stereo hits near axial tracks, simple χ^2 fit
		L3: Histogram & Kalman
DØ	Fiber	L1: lookup table (8 axial double–layers)
		L2: simple χ^2 fit, classification
		L3: road following (Kalman–like), silicon+fiber
CMS	Silicon	L1: none
		L3: Kalman + DAF (tracks/vertex) + GSF (electrons)
ATLAS	Straw tubes	L1: none – “Regions of Interest” are passed on
		L2: Kalman filter with seeding from silicon
		L3: Inside–out (road following + DAF), followed by outside–in (Hough trans. + Kalman)

In each of these four cases, the L3 algorithms were the same as the offline reconstruction.

- DAF: Deterministic Annealing Filter, sort of probabilistic Kalman filter, said to be good for high occupancies
- GSF: Gaussian Sum Filter, said to be good for particles with non–Gaussian energy loss

Summary

- L1: Hardware–based combinatorial pattern matching, using powerful modern FPGAs.
- L3: Offline–quality track fitting (e.g. BaBar's fast Kalman or DAF)

- BaBar & ATLAS have similar geometries to PANDA STT, so they could be a good starting place.
  BaBar/BES–III's track finding algorithm said to handle curling tracks well, but requires z–information for them.
- Displaced vertices are generally handled well, though dealing with decays inside the STT take more planning.
- L1 trigger decision sensitive to beam–generated backgrounds.
- Kalman–type filters can handle track finding, fitting, vertexing, all in one algorithm.

- Lots of room to optimize algorithms!