Online Event Reconstruction
in HEP Experiments

Ivan Kisel
GSI

FIAS Colloquium
Frankfurt, December 3, 2009
<table>
<thead>
<tr>
<th>Research Center</th>
<th>Accelerator (GeV)</th>
<th>Experiment</th>
<th>Physics</th>
</tr>
</thead>
<tbody>
<tr>
<td>SLAC, USA</td>
<td>PEP-II, $e^- \times e^+$ (9 x 3.1)</td>
<td>BaBar</td>
<td>B-Physics</td>
</tr>
<tr>
<td>Fermilab, USA</td>
<td>Tevatron, $p \times p$ (1000 x 1000)</td>
<td>D0</td>
<td>Universal</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CDF</td>
<td>Universal</td>
</tr>
<tr>
<td>BNL, USA</td>
<td>RHIC, Heavy Ions</td>
<td>PHENIX</td>
<td>Quark-Gluon-Plasma</td>
</tr>
<tr>
<td></td>
<td></td>
<td>STAR</td>
<td>Quark-Gluon-Plasma</td>
</tr>
<tr>
<td>KEK, Japan</td>
<td>KEK-B, $e^- \times e^+$ (8 x 3.5)</td>
<td>BELLE</td>
<td>B-Physics</td>
</tr>
<tr>
<td>CERN, Switzerland</td>
<td>LHC, $p \times p$ (7000 x 7000)</td>
<td>ATLAS</td>
<td>Universal</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CMS</td>
<td>Universal</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ALICE</td>
<td>Quark-Gluon-Plasma</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LHCb</td>
<td>B-Physics</td>
</tr>
<tr>
<td>DESY, Germany</td>
<td>HERA, $e^{+/-} \times p$ (27.5 x 920)</td>
<td>ZEUS</td>
<td>Proton-Physics</td>
</tr>
<tr>
<td></td>
<td></td>
<td>H1</td>
<td>Proton-Physics</td>
</tr>
<tr>
<td></td>
<td></td>
<td>HERMES</td>
<td>Spin-Physics</td>
</tr>
<tr>
<td></td>
<td></td>
<td>HERA-B</td>
<td>B-Physics</td>
</tr>
<tr>
<td>FAIR/GSI, Germany</td>
<td>SIS 100/300, $p$, Heavy Ions</td>
<td>PANDA</td>
<td>Quark-Physics</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CBM</td>
<td>Quark-Gluon-Plasma</td>
</tr>
</tbody>
</table>
From Raw Data to Physics

1. Particle Accelerator
2. Particle Detectors
3. Data Acquisition
4. Data Reconstruction
5. Physics Analysis
HEP Experiments: Collider and Fixed-Target

Inelastic collisions: $10^7 - 10^9$

Signal events: $10^2 - 10^{-2}$

High energy = high density + high rate

Different experimental setups have similar event reconstruction problems!
Schematic View of a Detector Setup
Methods for Event Reconstruction

Track finding

- Global Methods
  - all hits are treated equivalently
  - typical methods:
    - Conformal Mapping
    - Histogramming
    - Hough Transformation

- Local Methods
  - sequential selection of candidates
  - typical methods:
    - Track following
    - Kalman Filter

- Neural Networks
  - combine local and global relations
  - typical methods:
    - Perceptron
    - Hopfield network
    - Cellular Automaton
    - Elastic Net

Track fitting

- Vertex finding/fitting

- Ring finding

Kalman Filter

Combinatorics

Time consuming!!!
Global Methods: Hough Transformation

**Measurement Space**

\[ y = a \cdot x + b \]

**Parameter Space**

\[ b = -x \cdot a + y \]

HT -> S. Lebedev
Local Methods: Kalman Filter for Track Finding

KF Fit

KF Find

Seeding Planes

Initial estimates for $R_0$ and $\Gamma_0$

Prediction step

Filtering step

State estimate $\hat{R}$

Error covariance $\hat{\Gamma}$

Track parameters and errors
Each **cell** has 8 neighboring cells, 4 adjacent orthogonally, 4 adjacent diagonally. The **rules** are:

**Survivals.** Every cell with 2 or 3 neighboring cells survives for the next generation.

**Deaths.** Each cell with 4 or more neighbors dies from overpopulation. Every cell with 1 neighbor or none dies from isolation.

**Births.** Each empty cell adjacent to exactly 3 neighbors is a birth cell.

It is important to understand that all births and deaths occur *simultaneously*.

---

M. Gardner, *Scientific American*, **223** (October 1970), 120-123
Neural Networks: Cellular Automaton – Animation

1. Hits

2. Segments

3. Counters

4. Track-candidates

5. Tracks
Track Finding in the Pattern Tracker of HERA-B (DESY)

Extremely low resolution and efficiency of the pattern tracker of HERA-B

<table>
<thead>
<tr>
<th>Parameter</th>
<th>OTR</th>
<th>ITR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hit resolution, μm</td>
<td>500</td>
<td>200</td>
</tr>
<tr>
<td>Hit efficiency, %</td>
<td>90</td>
<td>86</td>
</tr>
</tbody>
</table>

Hough Transformation

Kalman Filter

Cellular Automaton
Competition CATS(CA)/RANGER(KF)/TEMA(HT) (HERA-B, DESY)

The reconstruction package **CATS** based on the Cellular Automaton for track finding and the Kalman Filter for track fitting outperforms alternative packages (**SUSi**, **HOLMES**, **L2Sili**, **OSCAR**, **RANGER**, **TEMA**) based on traditional methods in efficiency, accuracy and speed.

### Tracking efficiency

![Tracking efficiency graph]

### Tracking quality

<table>
<thead>
<tr>
<th>Resolutions</th>
<th>CATS</th>
<th>RANGER</th>
<th>TEMA</th>
</tr>
</thead>
<tbody>
<tr>
<td>x, μm</td>
<td>246</td>
<td>93</td>
<td>322</td>
</tr>
<tr>
<td>y, mm</td>
<td>3.7</td>
<td>1.4</td>
<td>5.0</td>
</tr>
<tr>
<td>t_x, mrad</td>
<td>0.62</td>
<td>0.24</td>
<td>0.71</td>
</tr>
<tr>
<td>t_y, mrad</td>
<td>4.73</td>
<td>1.79</td>
<td>6.96</td>
</tr>
</tbody>
</table>

### Time consumption

![Time consumption graph]

### Parameters

- **P(x)**
  - CATS: 1.59
  - RANGER: 1.11
  - TEMA: 1.37
- **P(y)**
  - CATS: 1.52
  - RANGER: 0.98
  - TEMA: 1.25
- **P(t_x)**
  - CATS: 1.16
  - RANGER: 0.93
  - TEMA: 1.25
- **P(t_y)**
  - CATS: 1.53
  - RANGER: 0.99
  - TEMA: 1.30

<table>
<thead>
<tr>
<th>Hits/track</th>
<th>CATS</th>
<th>RANGER</th>
<th>TEMA</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>31</td>
<td>23</td>
<td>26</td>
</tr>
</tbody>
</table>

P(Ranger, TEMA) pulls the mean length of reconstructed primary tracks.
Tracking Challenge in CBM (FAIR/GSI, Germany)
Tracking Challenge in CBM (FAIR/GSI, Germany)

- Fixed-target heavy-ion experiment
- $10^7$ Au+Au collisions/sec
- ~ 1000 charged particles/collision
- Non-homogeneous magnetic field
- Double-sided strip detectors (85% fake space points)

Track reconstruction in STS/MVD and displaced vertex search is required already in the first level trigger.

First Level Event Selection (FLES) is done in a processor farm fed with data from the event building network.
Many-Core HPC: Cores, Threads and SIMD

Cores and Threads = task level parallelism

Process

Thread1  Thread2
...  ...
exe  r/w
r/w  exe
exe  r/w
...  ...

2010

CPU

Thread  Thread

2000

19XX

CPU

Core

Scalar  Vector

D  S

Vectors (SIMD) = data stream parallelism
CPU/GPU Programming Frameworks

- **Cg, OpenGL Shading Language, Direct X**
  - Designed to write shaders
  - Require problem to be expressed graphically

- **AMD Brook**
  - Pure stream computing
  - No hardware specific

- **AMD CAL (Compute Abstraction Layer)**
  - Generic usage of hardware on assembler level

- **NVIDIA CUDA (Compute Unified Device Architecture)**
  - Defines hardware platform
  - Generic programming
  - Extension to the C language
  - Explicit memory management
  - Programming on thread level

- **Headers and Vector classes (Vc)**
  - Overload of C operators with SIMD/SIMT instructions

- **Intel Ct (C for throughput)**
  - Extension to the C language
  - Intel CPU/GPU specific
  - SIMD exploitation for automatic parallelism

- **OpenCL (Open Computing Language)**
  - Open standard for generic programming
  - Extension to the C language
  - Supposed to work on any hardware
  - Usage of specific hardware capabilities by extensions
Core of Reconstruction: Kalman Filter based Track Fit

- **State vector**
  \[ r = \{ x, y, z, p_x, p_y, p_z \} \]
  - position
  - momentum

- **Covariance matrix**
  \[ C = \begin{pmatrix} \sigma_x^2 & 0 & \cdots & 0 \\ 0 & \sigma_y^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_z^2 \end{pmatrix} \]
  - error of \( x \)

- **Initial estimates**
  \( r_0 \) and \( C_0 \)

- **Prediction step**
  \( \tilde{r}_k, \tilde{C}_k \)

- **Filtering step**
  \( r_k, C_k \)

- **State estimate**
  \( r_n \)

- **Error covariance**
  \( C_n \)
## Kalman Filter Track Fit on Intel Xeon, AMD Opteron and Cell

<table>
<thead>
<tr>
<th>Stage</th>
<th>Description</th>
<th>Time/track</th>
<th>Speedup</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Initial scalar version</td>
<td>12 ms</td>
<td>–</td>
</tr>
<tr>
<td>1</td>
<td>Approximation of the magnetic field</td>
<td>240 µs</td>
<td>50</td>
</tr>
<tr>
<td>2</td>
<td>Optimization of the algorithm</td>
<td>7.2 µs</td>
<td>35</td>
</tr>
<tr>
<td>3</td>
<td>Vectorization</td>
<td>1.6 µs</td>
<td>4.5</td>
</tr>
<tr>
<td>4</td>
<td>Porting to SPE</td>
<td>1.1 µs</td>
<td>1.5</td>
</tr>
<tr>
<td>5</td>
<td>Parallelization on 16 SPEs</td>
<td>0.1 µs</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>Final simdized version</td>
<td>0.1 µs</td>
<td>120000</td>
</tr>
</tbody>
</table>

Motivated, but not restricted by Cell!

Table showing the stages of the Kalman Filter Track Fit on different platforms and their corresponding times and speedups.

- 2 Intel Xeon Processors with Hyper-Threading enabled and 512 kB cache at 2.66 GHz;
- 2 Dual Core AMD Opteron Processors 265 with 1024 kB cache at 1.8 GHz;
- 2 Cell Broadband Engines with 256 kB local store at 2.4 GHz.

**Graph:**
- Comparison of fit time per track for Intel, AMD, and Cell platforms.
- The Cell platform shows the most significant speedup, achieving a 10000 times faster performance compared to the initial scalar version.

**Vector classes**

**Figure 1:**
- Authors: lxg1411@GSI, eh102@KIP, blade11bc4 @IBM

---
Performance of the KF Track Fit on CPU/GPU Systems

<table>
<thead>
<tr>
<th>Type</th>
<th>Time/track, μs</th>
<th>Speed-up</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scalar double</td>
<td>2.6</td>
<td>–</td>
</tr>
<tr>
<td>Vector double</td>
<td>1.6</td>
<td>1.6</td>
</tr>
<tr>
<td>Vector single</td>
<td>0.7</td>
<td>2.3</td>
</tr>
</tbody>
</table>

Speed-up 3.7 on the Xeon 5140 (Woodcrest) at 2.4 GHz using icc 9.1

Scalability (μs) on different CPU architectures – speed-up 100 with 32 threads

Real-time performance on different Intel CPU platforms

<table>
<thead>
<tr>
<th>GPU</th>
<th>Type</th>
<th>Clock, GHz</th>
<th>Throughput, 10^6 tr/s</th>
</tr>
</thead>
<tbody>
<tr>
<td>NVIDIA Unit</td>
<td>8800 GTS 512</td>
<td>1.6</td>
<td>13.0</td>
</tr>
<tr>
<td></td>
<td>GTX 280</td>
<td>1.3</td>
<td>21.7</td>
</tr>
</tbody>
</table>

CBM Progress Report, 2008
Avoid random access to data - appropriate sorting of the measurements is necessary
First Level Event Selection (FLES) Complexity

World-wide LHC Computing Grid

- Largest Grid service in the world!
  - Around 140 sites in 35 countries
  - Tens of thousands of Linux PC servers (over 100,000 cores)
  - Tens of petabytes of storage

CBM DAQ/FLES

- 50 kB/ev
- 10¹ ev/s
- 100 ev/slice

Parallelization within and between the layers of the FLES architecture to provide a high CPU/GPU load

Big Bang

Farm
PC
CPU
Sub-Farm
PC
CPU/GPU
Socket
Core
Thread
Vector

-∞
2009
∞
Tracking Challenge in ALICE (CERN, Switzerland)
ALICE HLT Cellular Automaton Track Finder

Problems:
• ~ 10000 charged particles/collision
• High track density
• Huge number of measurements (TPC)

Solution:
• Parallel processing:
  • vectorization,
  • multi-threading,
  • many-core systems
Performance for p-p and Pb-Pb Events

100 p-p events

5.7 ms

1490 ms

161 ms

S. Gorbunov, ALICE HLT Group
HEP Experiments

ARES (JINR)

MMbar (PSI)

NEMO (Modane)

DISTO (Saclay)

COMPASS (CERN)

HERA-B (DESY)

LHCb (CERN)

ALICE (CERN)

CBM (FAIR/GSI)
**Publications**

- **98 total**
- **30 physics**
- **8 quoted by PDG**

---

**Selected publications**


---

**PhD Thesis**


---

**Lectures**

- I. Kisel, Pattern recognition in high-energy physics experiments (in German), SS 2005, WS 2006/07. PDF (225 MB).

---

**Full list of publications**

3. P. Szefer, I. Kisel et al. (CBM Project), 2002.
7. I. Kisel, V. Kovalenko, E. Laplache et al. (NEMO Collaboration), 2006.
Summary

• Reconstruction is an interdisciplinary branch of the experimental HEP (physics, mathematics, computers, detectors, electronics)
• Efficient event reconstruction is very expensive with farms of thousands of CPUs!
• Inefficient event reconstruction is even more expensive $\varepsilon_{\text{tot}} = (\varepsilon_{\text{phys}} \cdot \varepsilon_{\text{det}} \cdot \varepsilon_{\text{elctr}}) \cdot \varepsilon_{\text{reco}}$ !
• There are no general libraries, but we share experience between experiments
• Future HEP experiments will require more and more complicated reconstruction!