sPHENIX Track Reconstruction (and inner detector optimization)

sPHENIX in Asia Meeting

Christof Roland, MIT
for the tracking team

March 25, 2019

BNL
The sPHENIX tracking system

- **MVTX**
  - 3 layer vertex tracker based on Monolithic Active Pixel Sensors
  - Provides impact parameter resolution

- **INTT**
  - 4 layers of Silicon Strip detectors
  - See: Task Force report at Tallahassee meeting
    [https://indico.bnl.gov/event/5190/](https://indico.bnl.gov/event/5190/)
  - Fast response time for pileup disambiguation

- **TPC**
  - 48 layer continuous readout TPC
  - Main tracking device
  - Provides momentum resolution
TASK FORCE SUMMARY
Track Parameter Estimation vs INTT Mass (nLayers)

- The pion momentum resolution does not depend on the number of INTT layers
  - Verified by independent study based on the LIC detector model
- The electron reconstruction is severely affected by the material budget

Aug 24, 2018

sPHENIX General Meeting
Impact on Upsilon Mass Spectra...

- Simulation done with the latest INTT and TPC simulation
  – Need to reduce the material budget to the minimum necessary
• Requiring 2 MVTX hits and one INTT hit per reco track gives good rejection power against tracks originating from out of time pile up
Inner Tracker Task Force – Executive Summary

- Radiative tails in Upsilon in the e+ e- channel
  - Strongly suggest to limit the mass budget of the detector to the minimum necessary
- Momentum Resolution
  - Unaffected by the number of INTT layers
- Tracking efficiency
  - Low number of layers gives best performance, 0 layers probably not a viable option
- Pileup rejection
  - One INTT hit per track sufficient for oot vertex and track rejection
- Calibration precision for the TPC
  - No INTT layout can provide a single track pointing resolution that is smaller than the calibration accuracy requirement
  - Low number of INTT layers give best extrapolation performance
- Dead channel considerations and redundancy requirement
  - Install one more layer than the minimum number of hits required
- A 2 Layer (4 sub-layer) configuration probably gives the best performance while keeping sufficient redundancy with respect to dead channels and malfunctioning detector modules
- The radii of the layers should be chosen sufficiently small to provide full coverage of $|\eta| < 1$ for $|v_z| < 10$
  - Default positions of Layers 2 and 3 would provide full coverage for $|\eta| < 1$

For detailed information and plots see: https://indico.bnl.gov/event/5190/contributions/24445/
PATTERN RECOGNITION
Pattern recognition challenges

• Optimal parameter estimation
  – Momentum resolution
  – Impact parameter resolution

• Robustness against very high occupancy
  – High tracking efficiency
  – Little or no hit density dependence
    • Expect up to 30% detector occupancy in the TPC

• Efficient CPU usage
  – Goal is to reconstruct 100 Gevents per year
  – CPU usage dominated by track reconstruction
The algorithm

- Iterative Kalman Filter based track reconstruction package
  - Hough transformation based seeding algorithm
    - Provides redundancy against missing hits
    - Outside in approach
  - Track propagation and fitting based on the GenFit package
    - Open source software
    - Well tested through use in different experiments
      - E.g. PANDA, BELLE
    - Manpower efficient implementation
  - 2 iterations with hit removal and different seed constraints
    - 4 hits out of 7 layers
    - 6 hits out of 12 layers
Performance: Tracking Efficiency

- 100 pions embedded in HIJING
  - Various instantaneous luminosity scenarios
- Workable efficiency in worst case occupancy
  - Efficiency recoverable with
    - Better clustering algorithm
    - more iterations at the expense of CPU cycles
Performance: Parameter Estimation

- Unbiased momentum and impact parameter resolution in low occupancy events
- Slight deterioration of performance at high occupancy due to cluster centroid determination of overlapping clusters
  - To be fixed by more sophisticated clustering algorithm, e.g. neural networks
CPU resources for Tracking

• Prepare to reconstruct 96 billion events in Year-3
  • \(96\,000\,000\,000\,\text{ev} / 3600 \times 24 \times 365\,\text{sec} \sim 3000\,\text{ev/sec}\)

• Resources needed
  – **Assuming 15 sec per event:**
    • 45k equivalent-cores to reconstruct the data
    • within the year they were taken
      – Currently 34kCores allocated to STAR+PHENIX
    • 90k equivalent-cores for fixed latency reconstruction,
      – i.e reconstructing the data as they are taken modulo a calibration delay of 2-3 days
  – **Set target for tracking to 5 sec**
    • leave 10sec per event for calibration, Calo reconstruction, Particle Flow etc.

*Benchmark numbers discussed in the Computing Review, July 2018*
sPhenix Tracking Evolution vs time

Efficiency

- Central Hijing, Au+Au
- Central Hijing, Au+Au + Pile Up (200kHz)

December 2018

>90%

500 sec

2023

5 sec !!!

March 25, 2019
Need to speed up the track reconstruction by 2 orders of magnitude!!

March 25, 2019

sPHENIX in Asia Meeting
Is 5 sec per event a realistic goal?

Current sPhenix CPU Benchmark
- Min bias + pileup (200kHz): 480sec
  - Seeding: 420sec
    - scales with nhit^2
    - Out-of-time pileup hits a big concern
  - Pattern Recognition: 60sec

Examples from other experiments
- HLT tracking of ALICE
  - TPC Only -> sPhenix seeding step
    - ~1sec per ALICE event on 1 CPU, 24 Threads
      - 0.15 sec when scaled with sPhenix number of clusters on 24 threads
      - Min Bias + PU 420 sec -> 3.5 sec (one thread)
  - ATLAS tracking (ACTS Package)
    - Track propagation – 0.5ms/track (53 Layers)
    - Full RKF fit – 1-2 ms/track

5 sec per event should be feasible with state of the art technology

March 25, 2019
sPHENIX in Asia Meeting
What about Memory?

- Tracking uses up to ~10GB for a central event + pileup

- In order to process 100Billion events per year we need to be more efficient with respect to the memory consumption

- Target less than 2GB per job
  - LHC Grid standard requirement

⇒ Lower memory consumption allows to exploit efficient CPU usage, hyperthreading etc.
Ongoing work

• Redesign of the tracking storage objects
  – Move to more memory efficient storage model
    • New way of connecting truth and reco objects
    • Better sorting of hits
    • Faster access to subdetector data

• Reorganization of the simulation and tracking code
  – Split simulation from track reconstruction code into separate packages

• Modularization of the tracking code
  – Split monolithic tracking code into configurable modules
  – Crucial for collaborative software development

-> done!

-> mostly done!

-> work in progress
Ongoing work on algorithms...

Ideas for new Seeding scheme

• Nearest neighbor search in TPC
  – Use geometric indexing
    • E.g. R-trees from boost
    • Efficient access to hits near a given position (prediction)
  – Assemble track stubs to seed the KF

• Proof of principle study
  – 800ms to assemble track stubs in AuAu+100kHz pileup

• Optionally track the entire TPC
  – For the short extrapolation distances in the TPC we can probably find computationally cheaper algorithms than the KF
    • Neural networks
    • Cellular automata

• Need manpower to develop and implement new algorithms
Outlook

• We have ~4 years to bring our tracking code up to speed!
  – Current algorithm provides very high tracking efficiency and good robustness in view of the high occupancy conditions
    • Too slow and consumes too much memory 😞
• Target fixed latency reconstruction of recorded data -> 5sec/ev
  – We have a good idea of the code performance we need to achieve
  – There is technology available to get us there
  – Progress currently limited by the small size of the Tracking Team
    • Special thanks to Tony, Haiwang, Darren and Sookhyun for their heroic effort!
• Open to creative ideas for new algorithms
  – Particle tracking is a very active field of research
  – Many exciting new technologies emerging
    • Machine learning solutions
    • GPU/FPGA hardware acceleration
  – Many working “external” packages ready for testing
    • E.g ACTS (Open source ATLAS Tracking)
    • Manpower efficient implementation!
BACKUP
Radiative tails in Upsilon in the e+ e- channel
  - Strongly suggest to limit the mass budget of the detector to the minimum necessary

Momentum Resolution
  - Unaffected by the number of INTT layers

Tracking efficiency
  - Low number of layers gives best performance, 0 layers probably not a viable option

Pileup rejection
  - One INTT hit per track sufficient for oot vertex and track rejection

Calibration precision for the TPC
  - No INTT layout can provide a single track pointing resolution that is smaller than the calibration accuracy requirement
  - Low number of INTT layers give best extrapolation performance

Dead channel considerations and redundancy requirement
  - Install one more layer than the minimum number of hits required

A 2 Layer (4 sub-layer) configuration probably gives the best performance while keeping sufficient redundancy with respect to dead channels and malfunctioning detector modules

The radii of the layers should be chosen sufficiently small to provide full coverage of $|\eta| < 1$ (1.1?) for $|v_z| < 10$
  - Default positions of Layers 2 and 3 would provide full coverage for $|\eta| < 1$

For detailed information and plots see: https://indico.bnl.gov/event/5190/contributions/24445/
The pion momentum resolution does not depend on the number of INTT layers
  – Verified by independent study based on the LIC detector model
The electron reconstruction is severely affected by the material budget
• Simulation done with the latest INTT and TPC simulation
  – Need to reduce the material budget to the minimum necessary
Tracking Efficiency in AuAu + Pileup

- The 0 layer INTT option gives poor track matching efficiency to the MVTX at low momentum
  - Excluded early on and will not be shown in subsequent slides
- 4 Sub-Layer INTT configuration gives best performance for high occupancy data
pp efficiency and track selection

- $N_{TPC} > 20$
- $N_{TPC} > 20, N_{MVTX} \geq 2$
- $N_{TPC} > 20, N_{MVTX} \geq 2, N_{INTT} \geq 1$

Dec 5, 2018
sPHENIX MVTX Workfest
Requiring 2 MVTX hits and one INTT hit per reco track gives good rejection power against tracks originating from out of time pile up.
pp vertexing efficiency and purity

- Using ONE INTT layer to obtain timing information for a given vertex
- Vertexing efficiency and purity look good
Extrapolation to the TPC

• Need 150um position accuracy in the TPC to achieve our momentum resolution goal
  – Space charge distortions of $O(3\text{mm})$ expected
  – Need external calibration

• Test extrapolation precision from MVTX + INTT
  – Truth tracking to assemble the hits + Kalman Fit
  – Extrapolate to 30cm radius and compare extrapolation to truth hit position
Estimated Extrapolation dphi

- Estimated using geometric maximum-extent with momentum resolution from inner tracking.
  - no configuration meets the 150um limit needed for the TPC calibration
1, 2, 3, and 4 layer defaults...

- Fewer layers, i.e. less material budget gives better extrapolation performance
  - phi resolution similar with 1 and 2 layers.
Comparing 2 layer configurations

- Moving layers to larger R gives modest improvement
- Adding a layer with Z resolution gives only a small improvement
  - Two layers with phi resolution offers more redundancy in case of dead channels/chips
Placement of INTT layers

- Moving the INTT as close as possible to the TPC improves the extrapolation precision
  - But cuts into eta acceptance and puts Electronics in the path of high eta tracks
- Putting to outer INTT layer at ~11cm gives full eta coverage ($|\eta| < 1$) for $|z_{VTX}| < 10\text{cm}$
  - Poorer extrapolation accuracy to the TPC
- Decided to prefer eta coverage over extrapolation precision
Recommendation for the optimal INTT configuration

• Performance benchmarks suggest:
  – **1-2 points sufficient for pattern recognition**
    • matching TPC and Inner Tracking in AuAu+ pile-up
  – **Material budget of 2 layers (4 sub-layers) preferable to 4 layers**
    • Radiative tails in the electron reconstruction
  – **1 INTT hit sufficient for pile-up rejection**
    • vertex and track reconstruction
  – **2 layers provide best extrapolation precision to the TPC**
  – **2 layers with r-phi resolution provide sufficient redundancy**
    • Simulation with 8% dead channels in INTT give consistent performance with 0% scenario
• Layer position in $r$ constrained by $\eta$ acceptance, ~10cm radius optimal
  – Position of layer 2 and 3 in original 4 layer design, unless the $z$ coverage of the INTT ladders can be extended
Benchmarks from LHC experiments

- LHCb gets a reconstruction time of ~1ms per pp event + pileup (~500 tracks) from heavy use of neural networks, lookup tables and machine learning...

- CMS tracking software takes ~3ms per track with at nhit2 CPU dependence -> would translate to 1.5-5 sec for a sPhenix event

- ATLAS is about to release an open source tracking package – ACTS project
  - A. Salzburger has provided preliminary estimate for the performance for sPhenix

- Many proposals sent to NSF etc studying potential tracking acceleration using GPUs, FPGAs and ML in any linear combination thereof. Very promising approach.

- CONCLUSION
  - Assume a conservative kalman filter approach with a fully optimised code implementation
  - Estimated target CPU performance:
    - MB events + pileup:
      - 5 sec/event for tracking
      - 5 sec/event for TPC clustering, calibration, 3D vertexing and other services
      - 5 sec/event for calorimeter and particle flow reconstruction
    => 15 sec/event total projected event reconstruction time
Outlook on CPU performance

The current Kalman Track fit is very slow

- GenFit performance:
  - seed fit -> 40ms/track (~6 hits on track)
  - Internal Geometry handling not optimized
- CMS track fit performance
  - 1ms per track (14.5 hits on track)
- ATLAS tracking (ACTS Package)
  - Track propagation – 0.5ms/track (53 Layers)
  - Full RKF fit – 1-2 ms
- Projection:
  - Min Bias 49 sec -> ~1.0 sec
  - Min Bias + PU 65 sec -> ~1.5 sec
Data Volume Estimate

- Compressed Raw Data event size
  - Run 3,5 Au+Au: 2.3Mbyte

- Final Analysis Objects/Model not yet defined

- Candidate models under consideration:
  - ALICE HLT data format
    - Store tracks + hit information as residuals relative to tracks
    - Allows to reapply distortion corrections and refit tracks
    - Provides a compression factor of ~5, i.e. ~500kbyte/event
  - CMS miniAOD like data format
    - Limited precision storage of track/particle parameters including covariance matrices
    - ~25bytes per PF candidate after root compression
    - For 800 PF candidates (2xnTracks) -> 20kbyte/event
Event summary – 9BG event

- Total Memory usage of this event -> 8.95GB
  - Number of Hits:
    - 5481770 size: ~40bytes -> ~200MB
  - Number of Clusters:
    - 284729 size: 144 + n refs: 4249983 -> ~50MB
    - Stored 3 times (at least) Clusters, 3DHits (hough), Measurements (genfit) => 150MB
  - Number of Cells
    - TPC|INTT|MAPS:
      - 9212401 | 12481 | 83604 cell size: 136 -> 1210MB
    - Known sources of memory consumption account for ~2GB

- Need to be careful with the implementation of our storage objects
  - Likely culprit for “dark” memory: inheritance from TObject + heavy usage of templated data structures (STL)
  - Get rid of the cells (planned for next release) and optimize our simulation, reconstruction and storage strategies to optimize memory