Tracking with Digitized Hits

Cosmin Deaconu

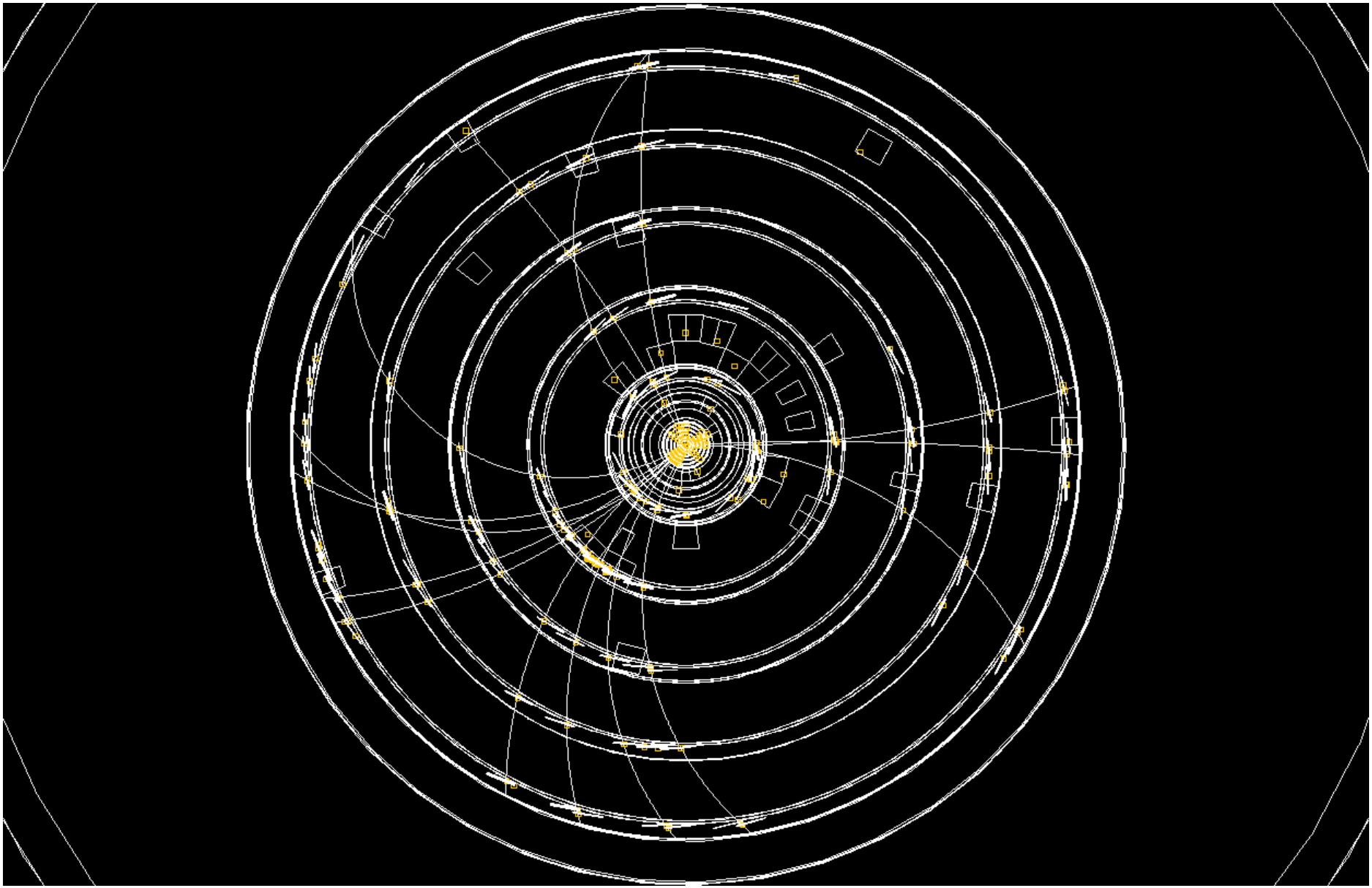
Stanford / SLAC

July 11, 2008

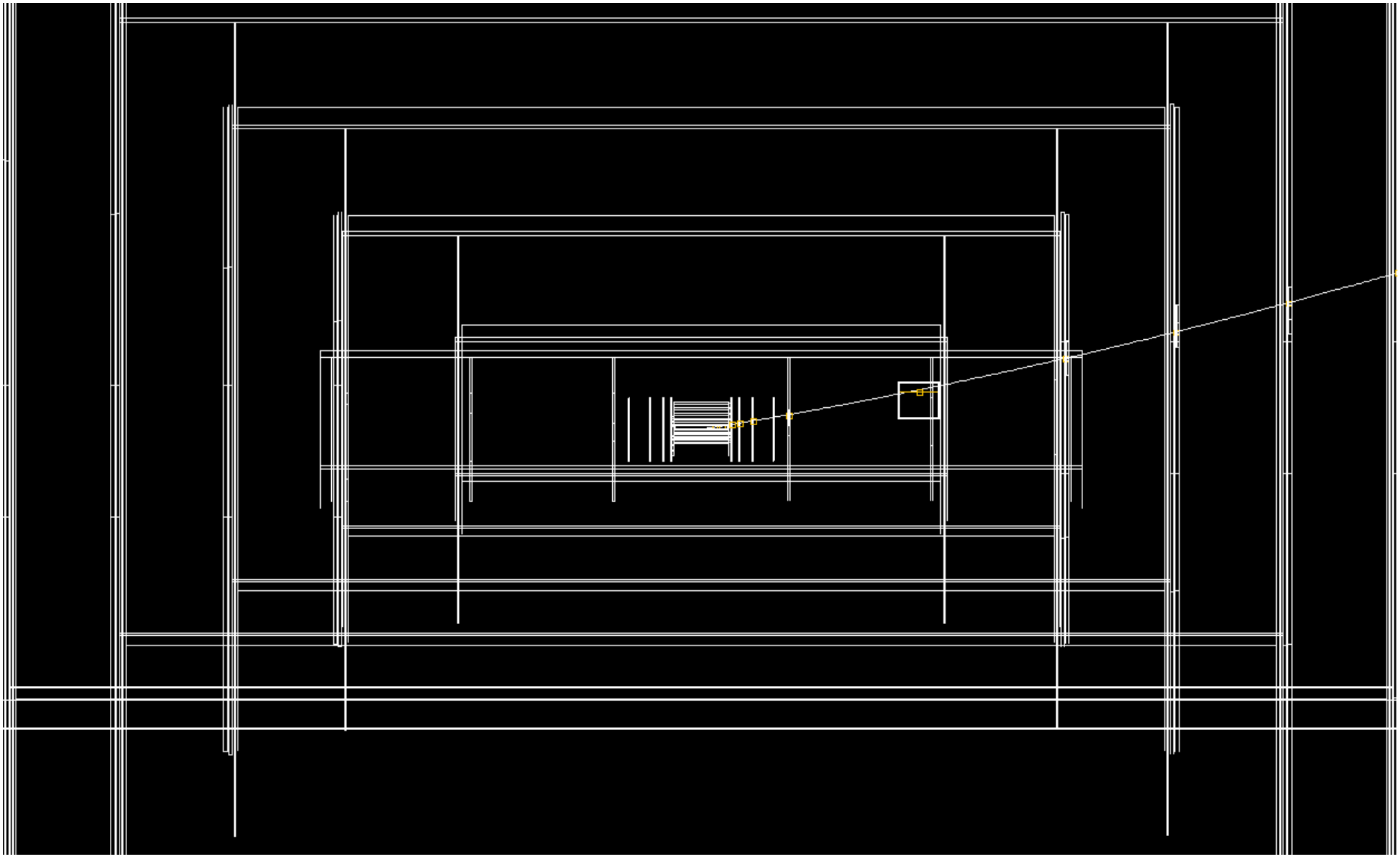
Introduction

- SeedTracker is designed so that it can accommodate different detectors and hit types. It does this by generating its own geometry model and by translating hits to its own HelicalTrackHit type.
- Previously, SeedTracker was focused only on hits generated by virtual segmentation in the sid01 detector model.
- I've made the necessary modifications such that SeedTracker may be used with fully digitized hits in the sid01_planar_tracker model.
- A working driver that finds tracks for the planar geometry may be found at org.lcsim.contrib.seedtracker.digiexample.DigiSeedTrackerDriver
- A test case was also written so that it will be more obvious when I break it =).

Example



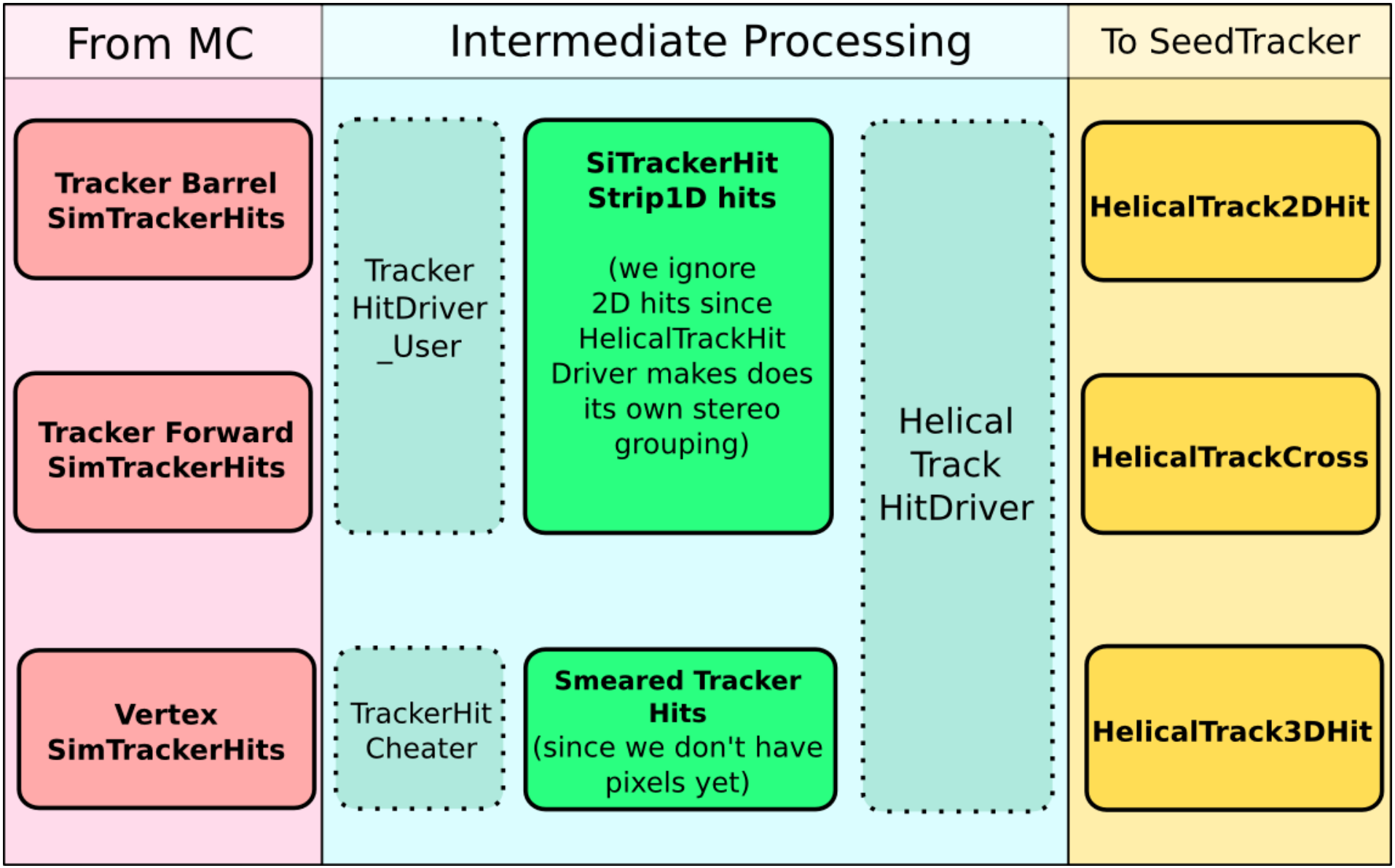
Example



Steps that were required for tracking with the planar tracker

1. Hit conversion from SiTrackerHitStrip1D and smeared hits to HelicalTrackHits.
2. Enhanced geometry modeling
3. Slight modifications to seed strategies

Hit Conversion: Overview



Hit Conversion: Strip treatment.

What HelicalTrackHitDriver does to SiTrackerHitStrip1D's

- Loop through all strips.
- Does this strip's module have multiple sensors?
  - Yes: create a HelicalTrackStrip from this strip and add it to a list of strips and continue.
  - No: make a HelicalTrack2DHit from the strip and add it to the event.
- At the end, run the StereoHitMaker on the list of HelicalTrackStrips to create HelicalTrackCrosses. Add the crosses to the event.

- We don't use the SiTrackerHitStrip2D's created by TrackerHitDriver_User because it does not attempt to make stereo hits across modules.

Geometry modeling

- SeedTracker creates its own geometry model for multiple scattering calculations. It models each second-level element (i.e. a barrel layer) as a cylinder or disk and calculates its radiation length.
- Initially, there was no support for dealing with boxes (especially rotated boxes). Because sid01_planar_tracker has a lot of boxes in it, this was sa problem. This has been remedied.
- There is still an unsettled issue regarding modeling polycones. Currently, only cylindrical sections of polycones are modeled. (This isn't something unique to the planar tracker.)
- In addition to adding new features, SeedTracker's MaterialManager was rewritten to be faster (and to only run when the detector is changed).

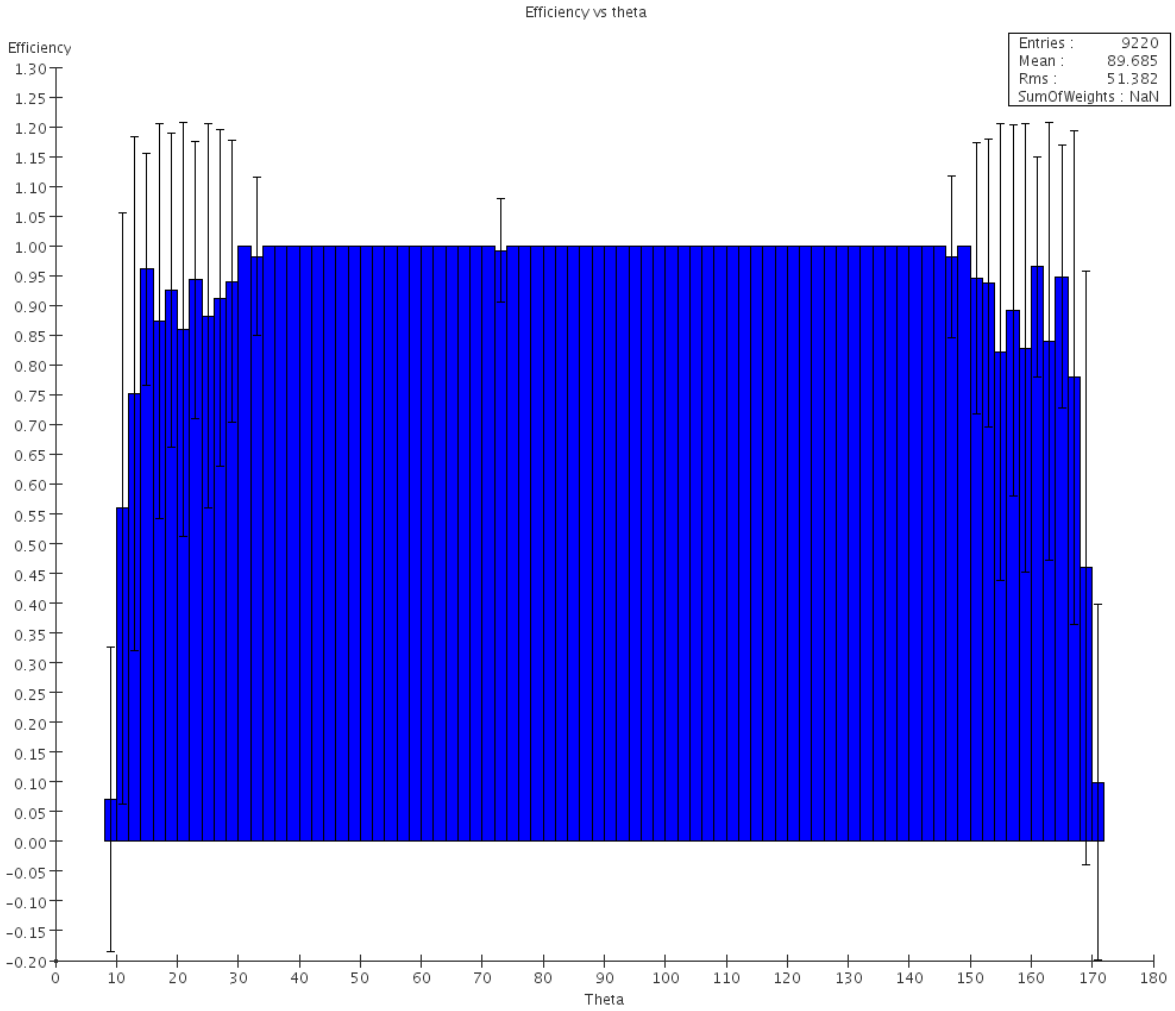
A word about SeedTracker strategies

- Reminder: A SeedStrategy defines the seed, confirmation and extension layers for the track finder. A layer is defined by a number, a detector name and a barrel/endcap flag (necessary to disambiguate the two forward regions).
- Differences between the two detectors:
  - In sid01, the layer numbering is 0-indexed but in sid01_planar_tracker the layer numbering is 1-indexed.
  - If a detector is named FooBar in sid01, it will be named SiFooBar in sid01_planar_tracker.

    Example: TrackerBarrel in sid01 becomes SiTrackerBarrel in sid01_planar_tracker.
- This means that currently strategies must be catered to each type of detector. It is possible to do a hard coded conversion, but that's not a good idea because these things can easily change for the planar tracker.

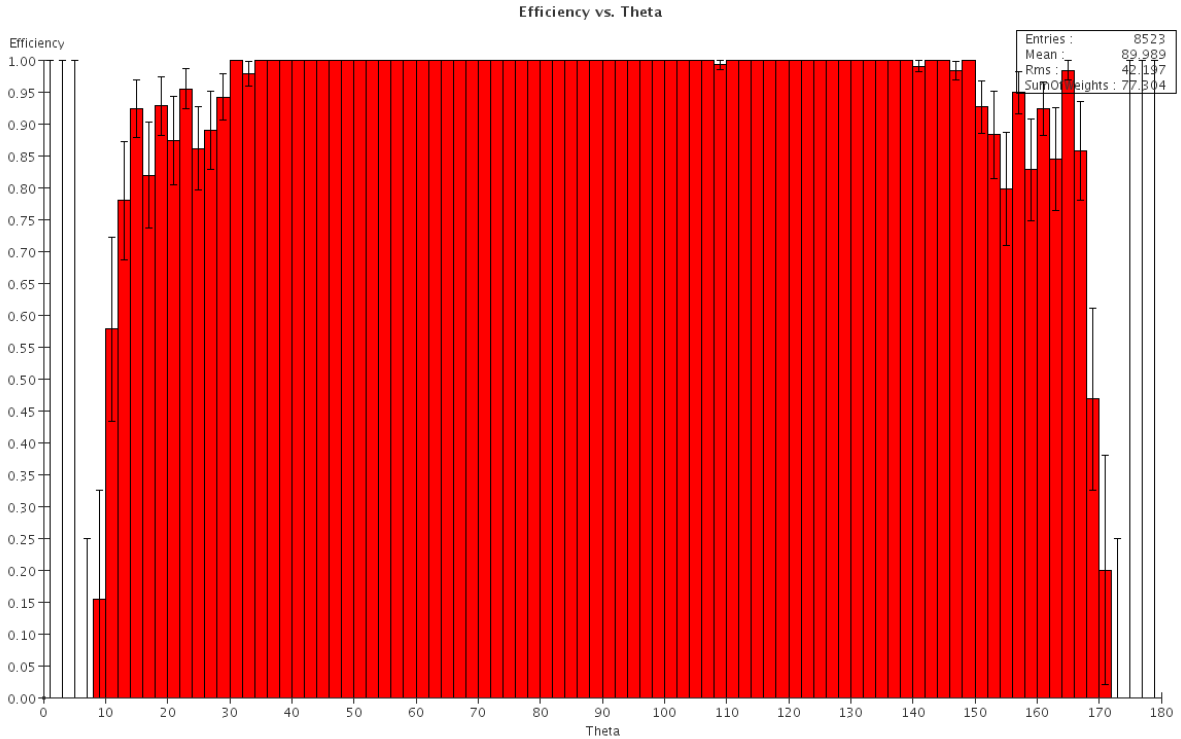
Preliminary efficiency results

Efficiency vs. θ using Digitized hits (100 GeV muons):



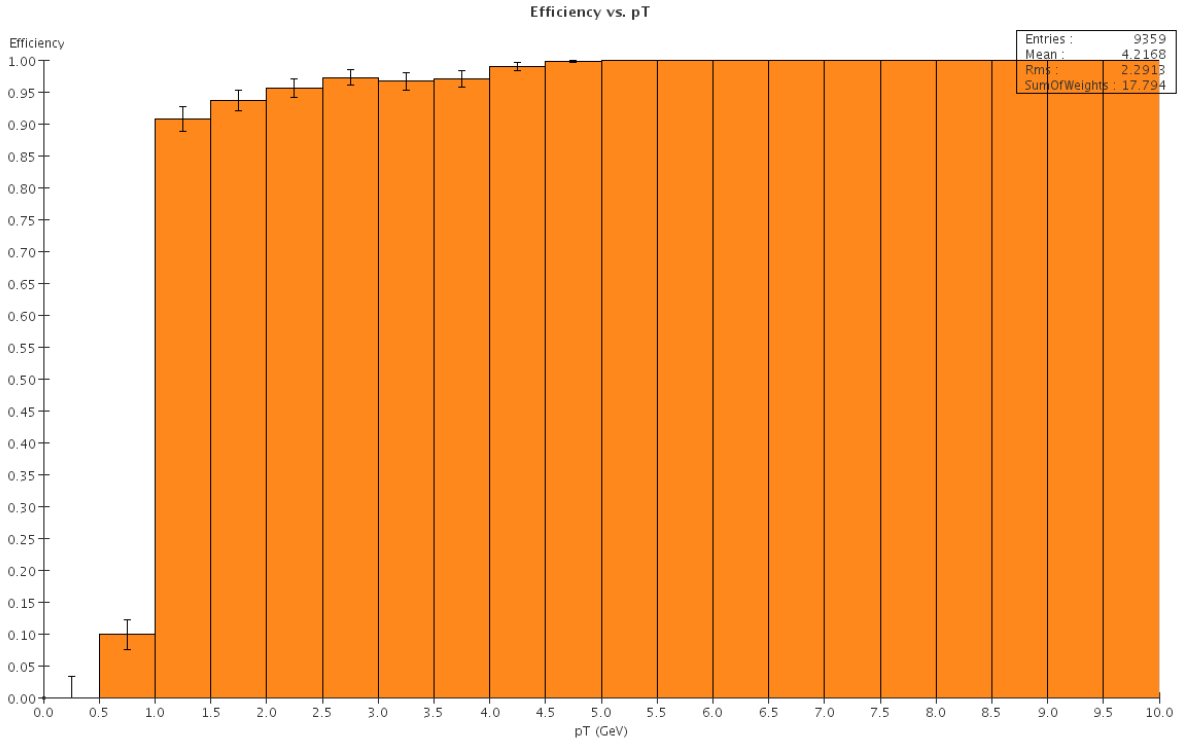
More preliminary efficiency results

Efficiency vs. θ using Digitized hits (1-10 GeV muons):



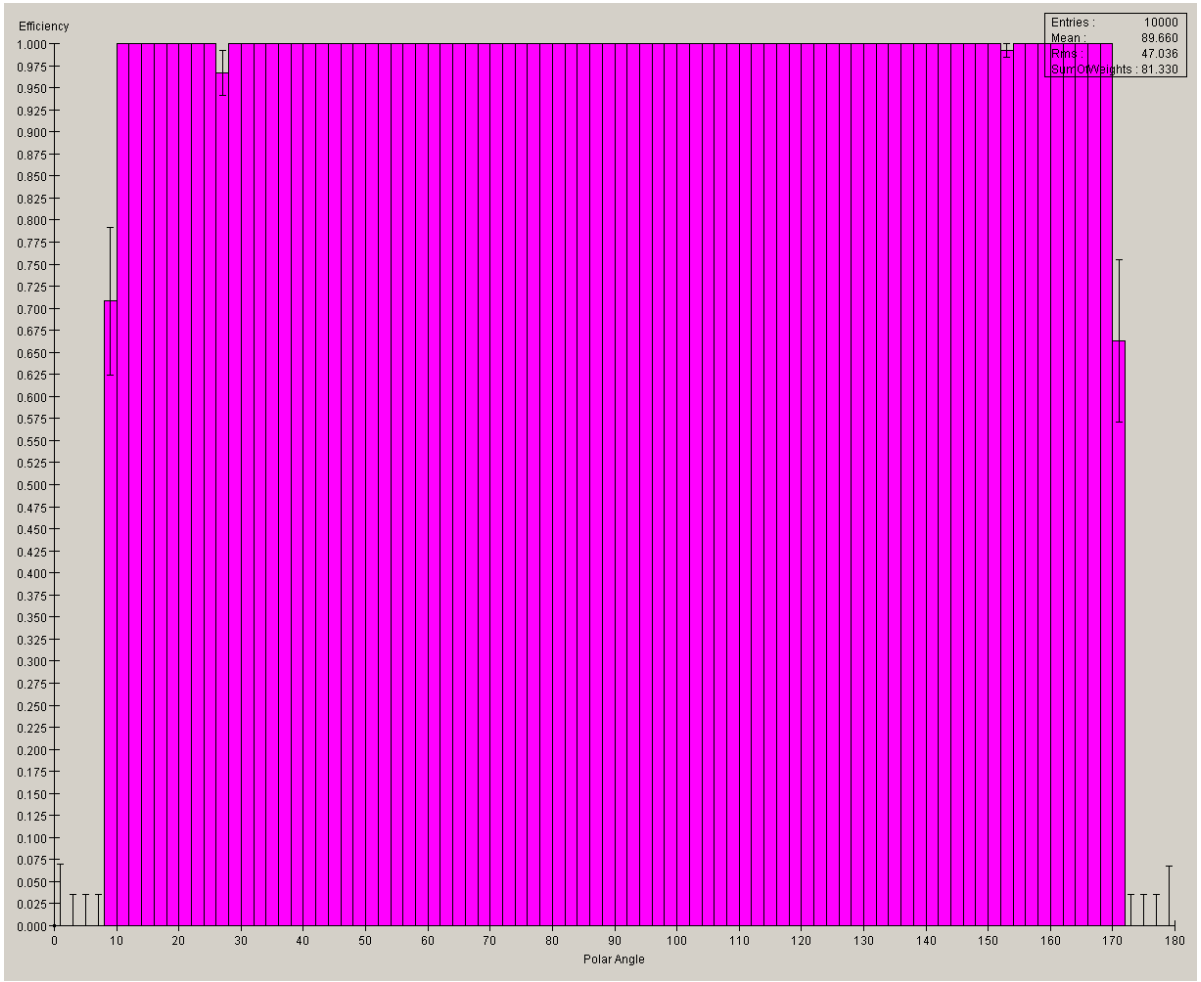
More preliminary efficiency results

Efficiency vs. p_T using Digitized hits (1-10 GeV muons):



Comparison with VS hit results

Efficiency vs. θ using VS hits (stolen from one of Rich's presentations):



Discussion of preliminary results

- Unclear at this point why forward tracking appears less efficient with digitized hits than with VS hits. Bad comparison? Possible bugs? Something real? Something to look at.
- SeedTracker's VS hit support is currently broken in the CVS version, which makes direct comparison difficult (which is why I had to use one of Rich's plots).
- SeedTracker is currently painfully slow on more complicated events. This also needs to be worked on. A lot.
  - In order to properly encapsulate the possible combinations of hits in endcap/barrel layers, about 20 strategies must be run for each event.
  - Profiling reveals that the most time-consuming task is calculating multiple scattering errors when extending hits.
  - Memory use is also a problem.

What's next:

- Obviously, further testing and hopefully some speed improvements.
- When SiTrackerHitPixel is ready, that will be used instead of smeared hits.
- Detector optimization. At this point, we should be able to tweak detector parameters in the compact.xml with impunity. (Maybe. We'll see how true that is soon).