ALCPG/SiD Tracking

Norman Graf

(SLAC)

ICFA-ILC Simulation & Reconstruction Workshop

May 3, 2007

SiD Workshop

- SiD held a workshop at Fermilab April 9-11.
  - http://ilc.fnal.gov/detector/rd/sid/sid07.htm
- Number of parallel sessions devoted to subdetector elements.
- Talks available at:
  - http://ilcagenda.linearcollider.org/conferenceTimeTable.py?confId=1390
- Repeating a number of slides from summary talk given by Dima Onoprienko.

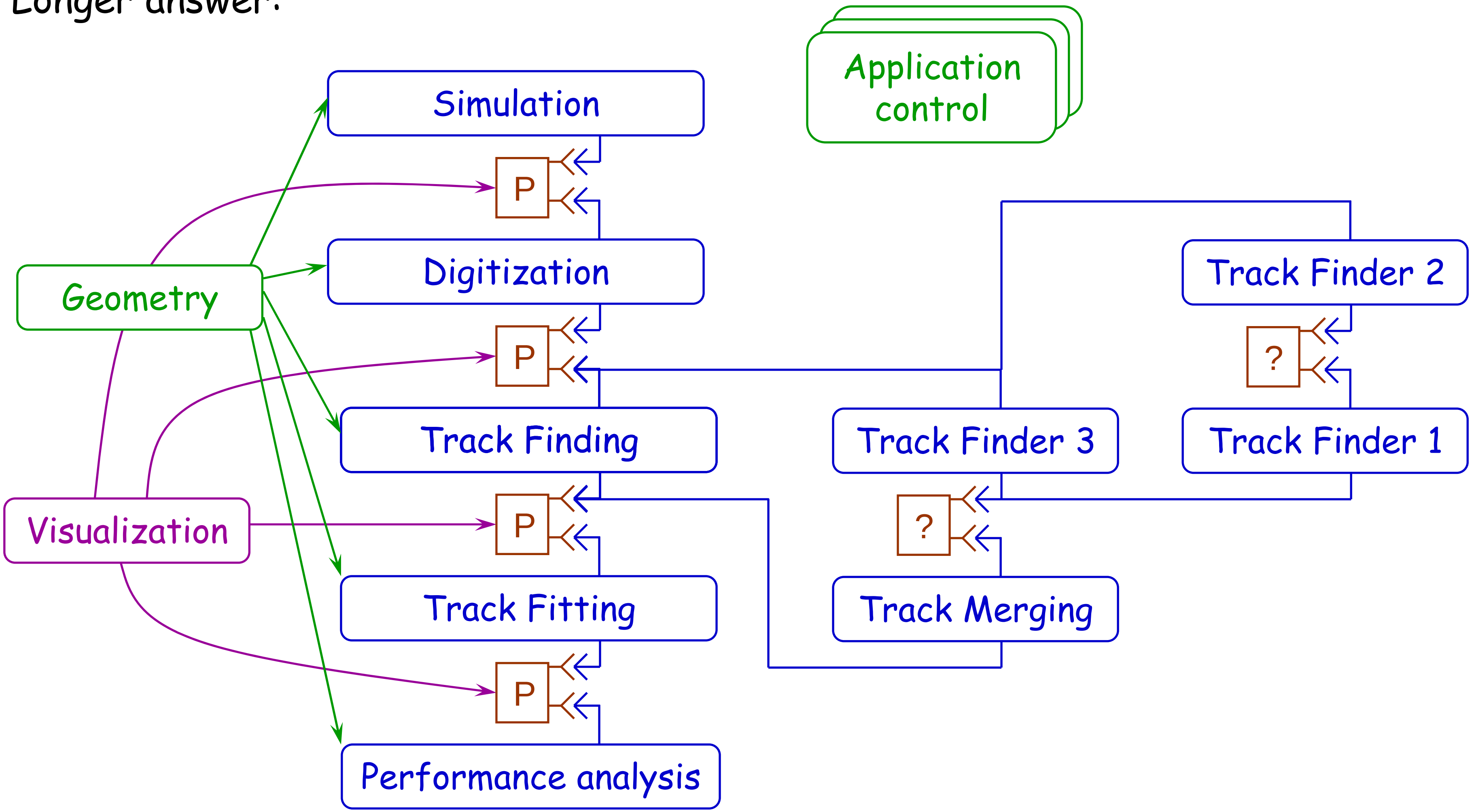
Tracking Software Infrastructure

- Main tracking efforts in ALCPG sim/reco group are aimed at supporting more realistic tracking detectors.
- Improved geometries
- Better hit digitization
- More complex track finding and fitting

Tracking software - what is needed ?

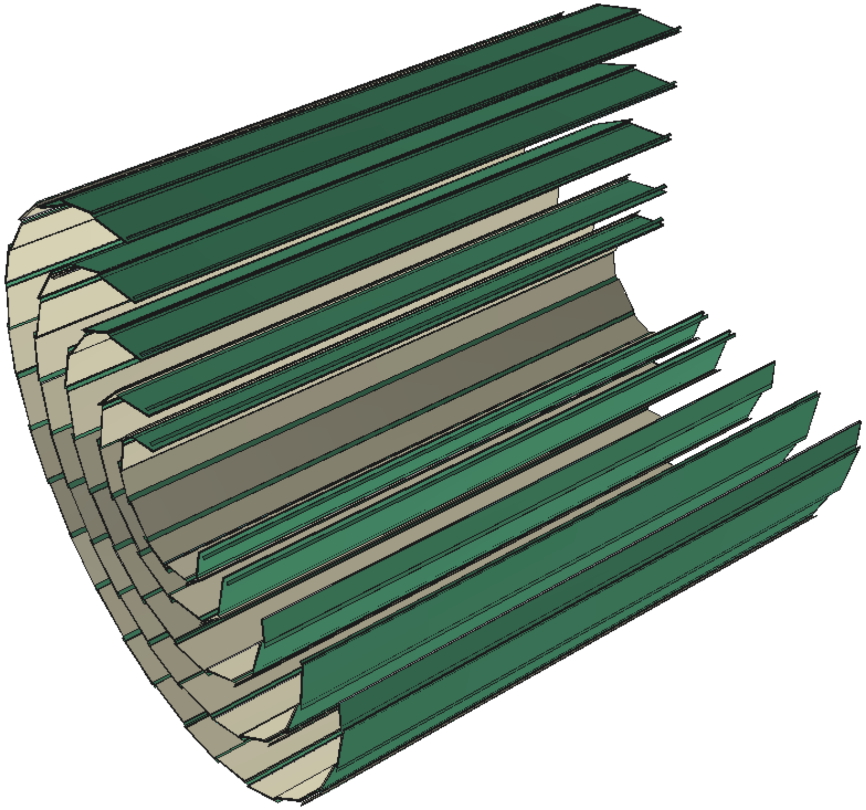
Short answer: algorithms and infrastructure to optimize the detector and characterize its performance.

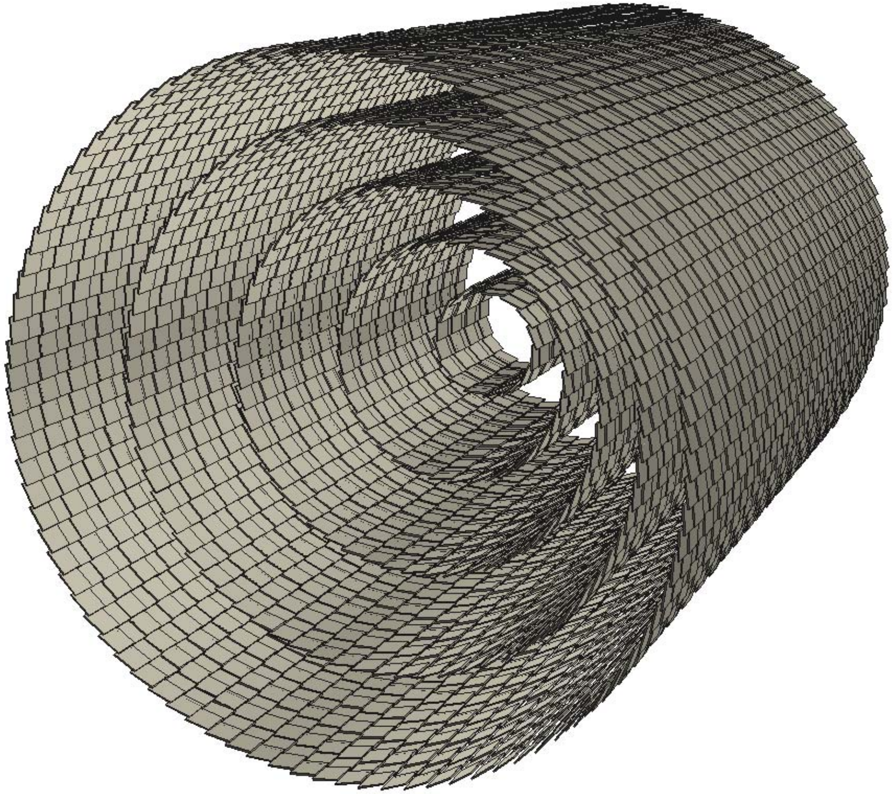
Longer answer:



Improved Geometries

Vertex Barrel with Staves





Outer Tracker Barrel with modules

Tracker Hit Digitization (Pixel)

- FullCCDSimulation package to simulate charge diffusion and all effects of pixel signals digitization exists. Developed by Nick Sinev (Oregon) and released at Paris LCWS (2004). It was verified by comparison with VXD3 data from SLD. Being used for a number of tracking studies.
- Recent extensions to other technologies.
- Analysis of new split-column CCD readout presented at SiD workshop.
- Talk available here.

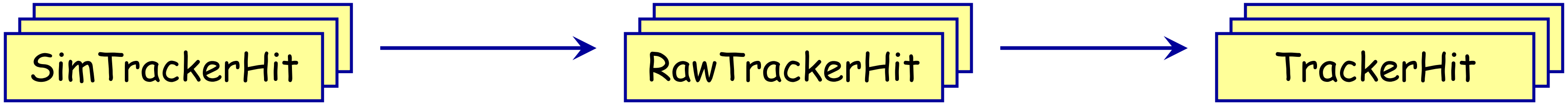
Tracker Hit Digitization (Strips)

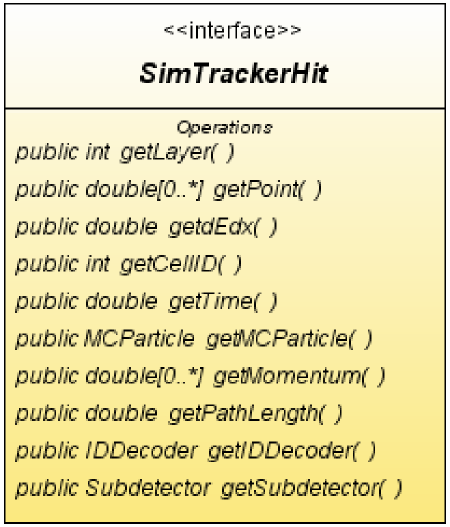
- Conversion of SimTrackerHits to TrackerHit for silicon microstrip detectors under active development.
- Require tight coupling to geometry to be able to extract information about orientation wrt magnetic field, strip direction, pitch, etc.
- Geometry infrastructure (local ↔ global) ~complete.
- Focus now on simulations of drift, diffusion, readout electronics, etc.
- Hope to have package released by LCWS07.
- Would like to interact with people doing similar tasks here.

TrackerHit for 1D & 2D measurements

- One problem we have had is how to persist TrackerHit objects which do not represent a 3D point measurement.
- Strip detectors are intrinsically 1D
    - measure coordinate normal to strip direction in sensor plane
- Pixel detectors are intrinsically 2D
    - measure coordinates in the plane of the sensor
- Not correct to treat these as 3D measurements with large uncertainties in non-measured domain.
- Would like to address these issues at this workshop
- How have others handled/solved this?

Tracking infrastructure - currently used hit objects



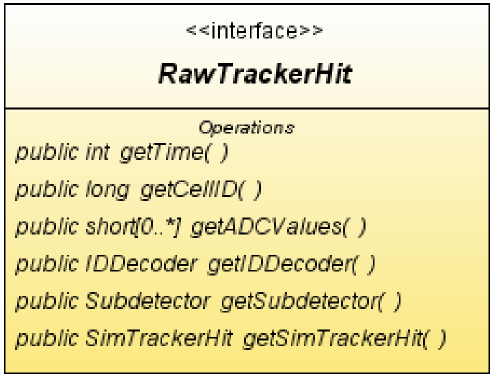


<<interface>>

SimTrackerHit

Operations

public int getLayer( )
public double[0..] getPoint( )*
public double getdEdx( )
public int getCellID( )
public double getTime( )
public MCParticle getMCParticle( )
public double[0..] getMomentum( )*
public double getPathLength( )
public IDDecoder getIDDecoder( )
public Subdetector getSubdetector( )

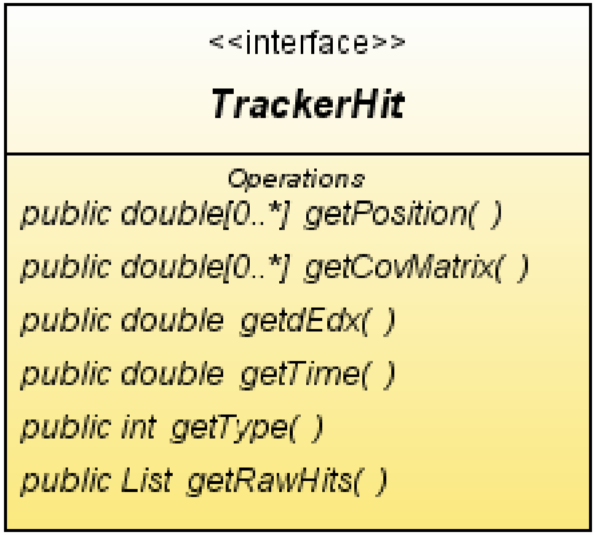


<<interface>>

RawTrackerHit

Operations

public int getTime( )
public long getCellID( )
public short[0..] getADCValues( )*
public IDDecoder getIDDecoder( )
public Subdetector getSubdetector( )
public SimTrackerHit getSimTrackerHit( )



<<interface>>

TrackerHit

Operations

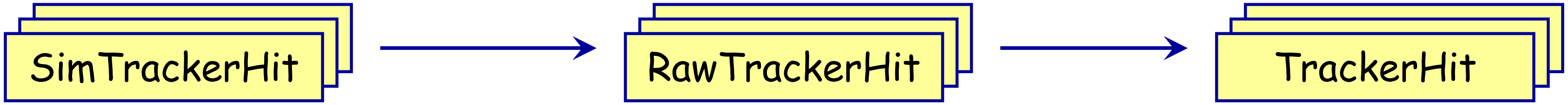
public double[0..] getPosition( )*
public double[0..] getCovMatrix( )*
public double getdEdx( )
public double getTime( )
public int getType( )
public List getRawHits( )

TrackerHit issues :

- No clear way to access MC truth
- No access to geometry
- No way to describe a non-point-like object like a silicon strip.
- Unsafe and ambiguous return types.

To work around these issues, various packages introduced their own custom extensions, resulting in incompatibilities between different tools.

Tracking infrastructure - currently used hit objects



Other missing ideas :

Clusters of hit strips/pixels.

Given a cluster or hit, how to navigate to associated geometry information ? Given a geometry object, how to find associated hits and clusters ?

Existing position info is in global coordinates but the native information of strips/pixels is in local coordinates.

Concept of "Hit on a Track":

> Some corrections can only be made after a cluster is associated with a track.
> How to handle cluster splitting ?

Classes that organize hit objects for fast access by pattern recognition and fitting code.

Clear definition of the point after which processing of real data and processing of MC events proceed along the same path.

> We want to use the same framework for MC, test beam data and SiD data.

Defined mechanism for overlapping events, or adding noise to the event.

Access to MC truth is sometimes incomplete and awkward to use.

Reconstruction - Track Finders

- **Vertex Seeded Tracking** (Fred Blanc, Steve Wagner)

  Package: **org.lcsim.contrib.SODTracker**

  Find track seed in vertex detector and pick up hits in outer tracker

  Uses MC truth for vertex seeds and runs on SimTrackerHits at the moment

- **Conformal Mapping** (Norman Graf)

  Packages: **org.lcsim.recon.tracking.trf*****

  Conformal mapping of circular trajectory to a straight line

  Uses alternative tracking infrastructure

- **Calorimeter Seeded Tracking** (Dima Onoprienko, Eckhard von Toerne)

  Package: **org.lcsim.recon.cat**

  Find MIP stubs in the calorimeter and extrapolate them into tracker, picking up hits

  Can be used to reconstruct long-lived particles (K_S^0, Λ, etc.)

- **Standalone Outer Tracking** (Tim Nelson, Bruce Schumm)

  Package: **org.lcsim.contrib.tracking**

  Find track candidates using outer barrel tracker

- **Standalone Outer Tracking** (Rich Partridge) [Under development]

  Package: **org.lcsim.contrib.seedtracker**

  Flexible selection of seed layers and track selection criteria.

Reconstruction - Fitters

Available algorithms :

- **Weight matrix** (Nick Sinev)

  Package: **org.lcsim.contrib.NickSinev.tracking.wmfitter**

  Complete, functional

- **Kalman filter** (Fred Blanc, Steve Wagner)

  Packages: **org.lcsim.contrib.KFFitter**

  Work in progress

- **Kalman filter** (Norman Graf)

  Package: **org.lcsim.recon.tracking.trffit**

  Designed to work with TRF toolkit

- **… and a few packages have their own private fitters.**

Global Track vs Trajectory

- Track in the trf context contains the ordered list of hits and (optionally) optimal fits (including MCS and energy loss) at each measurement/interaction surface.
- Track is able to return the best fit at any interior point.
- Propagators are defined which allow the track (parameters plus covariance matrix) to be extrapolated beyond the measurements.
  - Precision of the extrapolation depends on how well the material in the intervening region is handled.
  - Requires tight coupling with the geometry system.
  - Current trf package requires that the detector be well modeled as an assembly of surfaces.

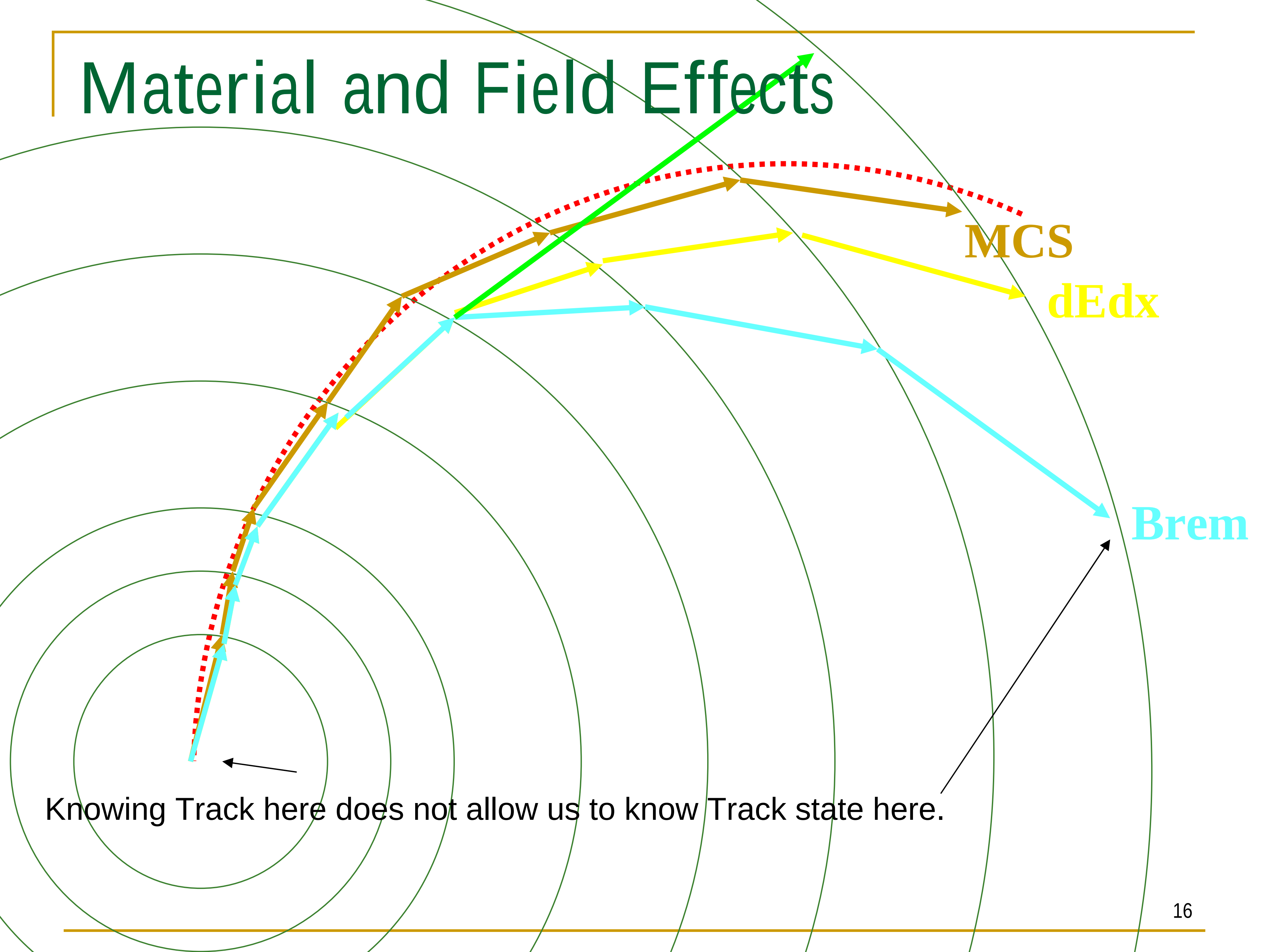
What is a track?

- Ordered association of digits, clusters or hits (finder)
  - Digit = data read from a detector channel
  - Cluster = collection of digits
  - Hit = Cluster (or digit) + calibration + geometry
    - Provides a measurement suitable to fit a track
    - E.g. a 1D or 2D spatial measurement on a plane
- Trajectory through space (fitter)
  - Space = 6D track parameter space
    - 3 position + 2 direction + 1 curvature
  - 5 parameters and error matrix at any surface
- Track is therefore only piecewise helical.
  - default is to break track down by measurement layers.
  - could increase granularity for inhomogeneous fields

Track Definition

- Six parameters are required to determine a charged particle's ideal path in a magnetic field.
- However, knowing these parameters at a single point (*e.g.* the distance of closest approach to the beam, **dca**) is insufficient for precision fits due to material effects (dE/dx, MCS, bremsstrahlung) and field inhomogeneities.
  - No global functional form for the fit.
- Current LCIO Track interface definition is too simplistic by not allowing for these effects.

Material and Field Effects



Knowing Track here does not allow us to know Track state here.

Surfaces

- Surfaces generally correspond to geometric shapes representing detector devices.
- They provide a basis for tracks, and constrain one of the track parameters.
- The track vector at a surface is expressed in parameters which are “natural” for that surface.
- Abstract interface is defined and a number of concrete surfaces are currently implemented.
- Question for TPC community is whether hits on surfaces is a good model.

1.) Cylinder

- Surface defined coaxial with z, therefore specified by a single parameter r.
- Track Parameters: $(\phi, z, \alpha, \tan\lambda, q/p_T)$
- Bounded surface adds z_{min} and z_{max}.
- Supports 1D and 2D hits:
  - 1D Axial: ϕ
  - 1D Stereo: $\phi+\kappa z$
  - 2D Combined: (ϕ, z)
- Is a (ϕ, z) measurement at a fixed radius (center of annular pad row) a good model for TPC?

2.) XY Plane

- Surface defined parallel with z, therefore specified by distance u from the z axis and an angle ϕ of the normal with respect to x axis.
- Track Parameters: (v, z, dv/du, dz/du, q/p)
- Bounded surface adds polygonal boundaries.
- Supports 1D and 2D hits:
  - 1D Stereo: $w_v*v + w_z*z$
  - 2D Combined: (v, z)

3.) Z Plane

- Surface defined perpendicular to z, therefore specified by single parameter z.
- Track Parameters: (x, y, dx/dz, dy/dz, q/p)
- Bounded surface adds polygonal boundaries.
- Supports 1D and 2D hits:
  - 1D Stereo: $w_x*x + w_y*y$
  - 2D Combined: (x,y)

4.) Distance of Closest Approach

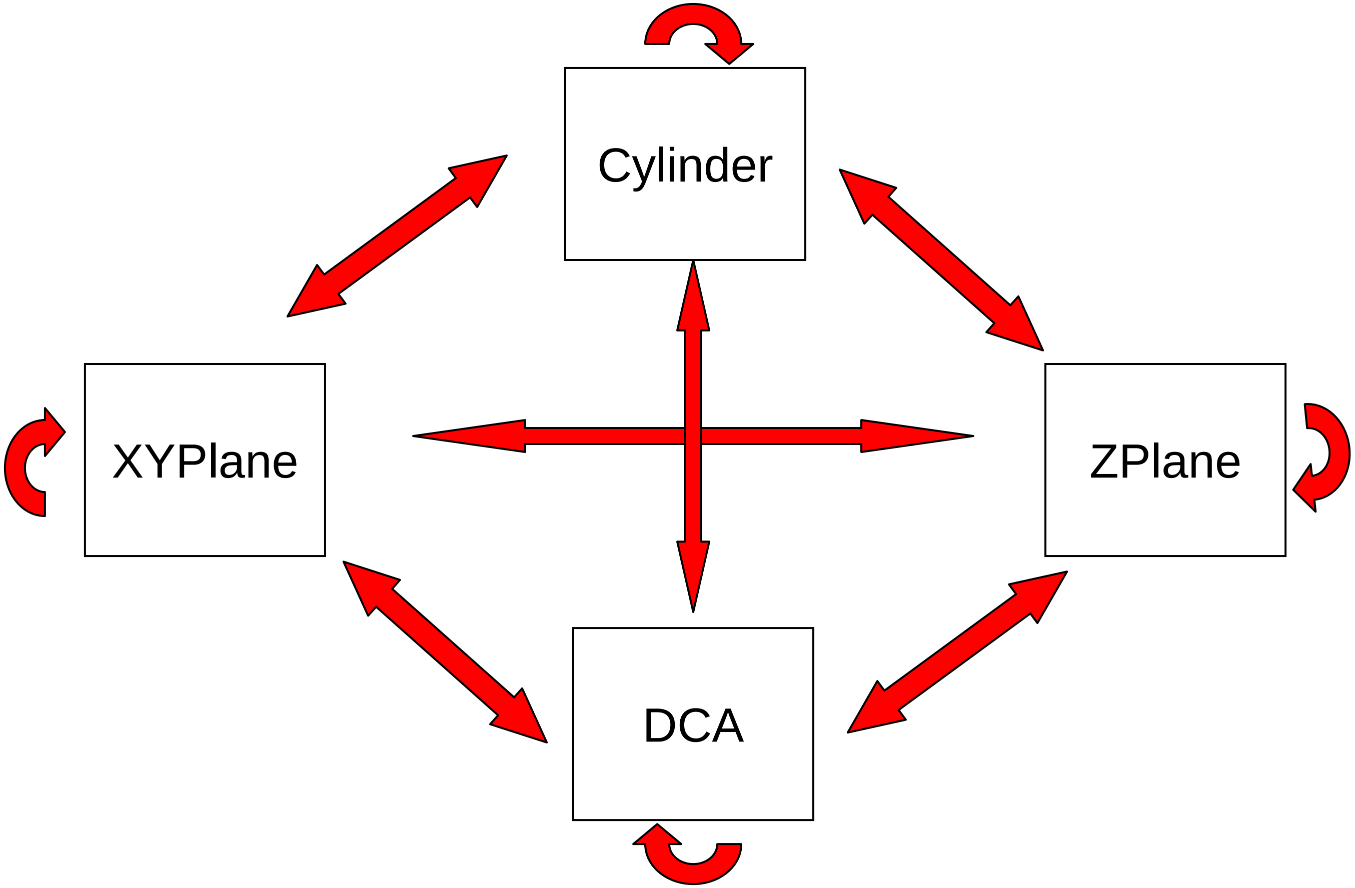
- DCA is also a 5D **Surface** in the 6 parameter space of points along a track.
- It is **not** a 2D surface in 3D space.
- Characterized by the track direction and position in the (x,y) plane being normal; $\alpha=\pi/2$.
- Track Parameters: $(r, z, \phi_{dir}, \tan\lambda, q/p_T)$

Propagator

- Propagators propagate a track (and optionally its covariance matrix) to a new surface.
- A propagator returns an object of type PropStat which describes the status of the attempted propagation:
  - *i.e.* whether it was successful and, if so, in which direction the track was propagated (forward or backward).
- Interacting Propagators modify the track and its covariance matrix (in case of energy loss), or just the covariance matrix (thin multiple scattering.)

Propagators

- Propagators are defined for all combinations of surfaces:



Interactors

- Describes the interface for a class which modifies a track. Examples are:
- Multiple Scattering
  - ThickCylMS
  - ThinXYPlaneMS
  - ThinZPlaneMS
- Energy Loss
  - CylELoss

Detector

- Use compact.xml to create a tracking Detector composed of surfaces, along with interacting propagators to handle track vector and covariance matrix propagation, as well as energy loss and multiple scattering.
    - Silicon pixel and microstrip wafers modeled as either xyplane or zplane.
    - TPC modeled as cyindrical layers (corresponding to pad rows).
    - Currently using thin multiple scattering approximations.
    - Using pure solenoidal field propagators
        - Runge-Kutta propagators available when needed.

Summary

- Tracking infrastructure issues that have been a bottleneck for quite a while are being actively addressed, primarily geometry & digitization.
- Would like to collaborate with similar efforts in other regions.
  - Common Interfaces < Common Algorithms < Common Code
- Tracking software tools for detector optimization are still far from perfect, but are perfectly capable of addressing at least some of the design questions – we need more people to use those tools.
- We're in the middle of an active development period on many fronts – stay in touch with developers and other users. Forum is a good place to ask questions:

users developers SiD

Backup Slides

Infrastructure components

- Hit
  - Defined at a surface.
  - Provides a measurement and associated error
  - Provides a mechanism to predict the measurement from a track fit
  - Provides access to underlying cluster and/or digits

TrackerHit

- Current TrackerHit interface only accommodates three dimensional hits.
- Many tracking subdetectors only provide one dimensional measurements (silicon microstrips) or two dimensional hits (such as silicon pixels).
- Furthermore, using Cartesian coordinates is not always the most natural for individual subdetectors.
- Cylinder:
  - 1D Axial: ϕ
  - 1D Stereo: $\phi+\kappa z$
  - 2D Combined: (ϕ, z)
- XYPlane:
  - 1D Stereo: $w_v*v + w_z*z$
  - 2D Combined: (v, z)
- ZPlane:
  - 1D Stereo: $w_x*x + w_y*y$
  - 2D Combined: (x,y)

trf Hits

- trfcyl:
  - HitCylPhi : a phi measurement on a cylinder.
  - HitCylPhiZ : stereo measurement on a cylinder.
    - phiz = phi + stereo*z.
  - HitCylPhiZ2D : measurement of both phi and z on a cylinder.
- trfxyp:
  - HitXYPlane1 : one dimensional v-z measurement on a XYPlane.
    - avz = wv*v + wz*z
  - HitXYPlane2 : two dimensional (v,z) measurement on an XYPlane
- trfzp:
  - HitZPlane1 : one dimensional xy measurement on a ZPlane.
    - axy = wx*x + wy*y
  - HitZPlane2 : two dimensional (x,y) measurement on a ZPlane