Status of Silicon Simulation and Hit Reconstruction

Tim Nelson, Jeremy McCormick

1/18/08

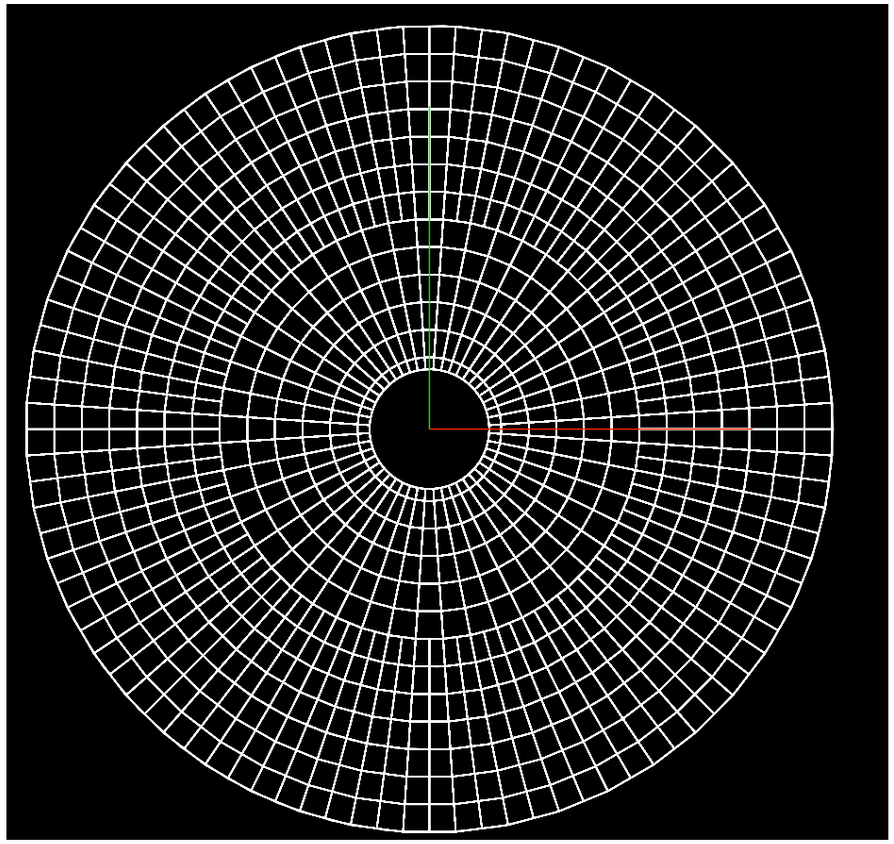
Overview

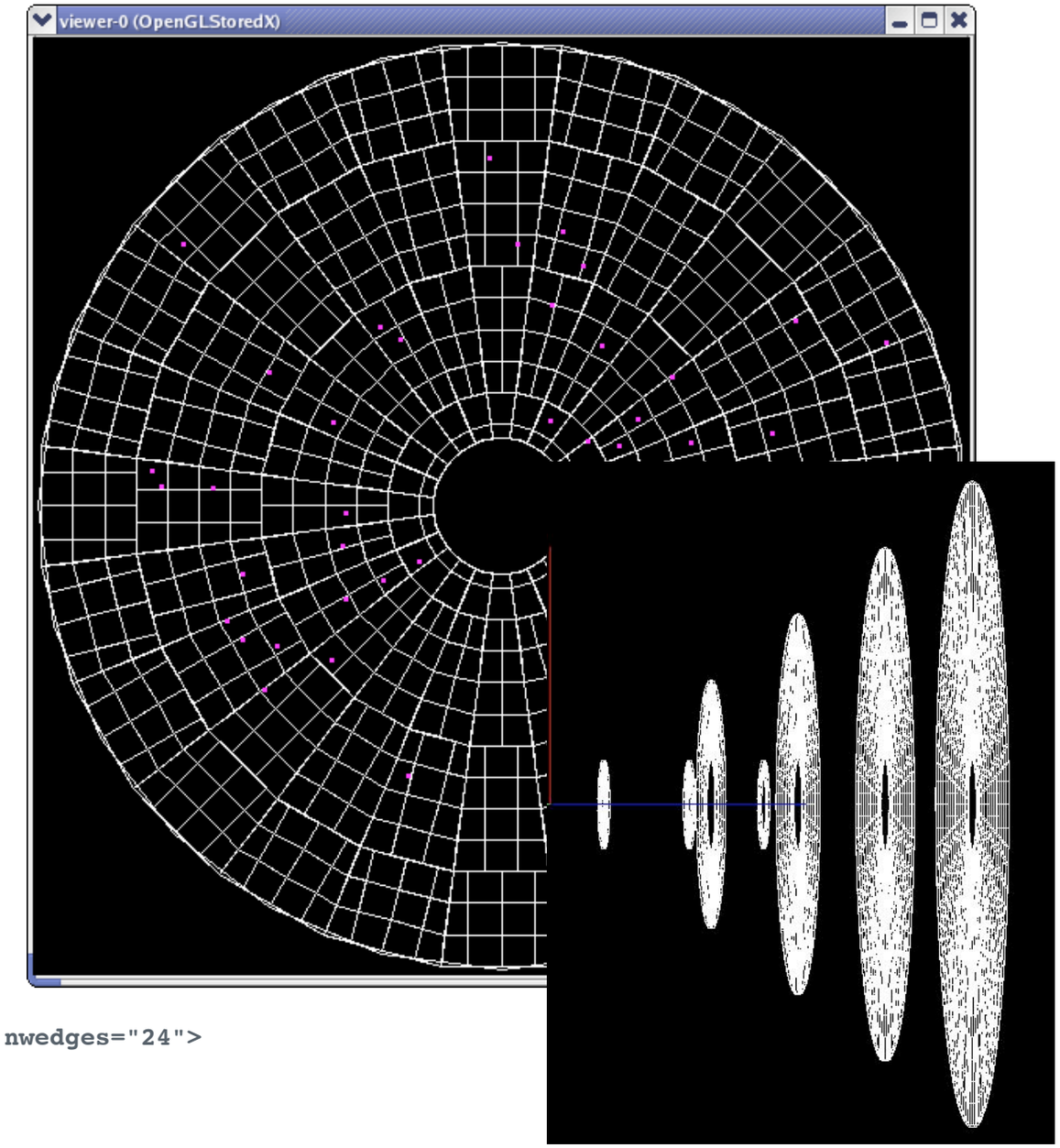
- Detector modeling / simulation ➡ SimTrackerHits (energy depositions as given by GEANT)
- Silicon simulation / digitization ➡ RawTrackerHits (ADC values from readout electronics)
- Hit clustering / reconstruction ➡ TrackerHits (clustered hits w/ positions)
- Pattern recognition / fitting ➡ Tracks

Detector Modeling

- *Compact description for planar barrel trackers*
  - Accommodates all layouts that have been discussed (including double-sided)
  - Specific models for outer tracker and vertex detector of baseline SiD
  - Need to review parameters and check dead material before put into production
- *Compact description for planar endcap trackers*
  - Can simulate any wedge-type geometry with any stereo angle (e.g. 90-degree)
  - Can be easily modified to to include tilts/overlaps for more realistic model
  - Specific model for outer tracker w/ 6 sensor shapes: need model for pixels
  - Need to review parameters and check dead material before put into production

Endcap Examples






```
<module name="SiTrackerEndcapModule">
    <module_component thickness="2.0" material="Silicon" sensitive="true" />
    <module_component thickness="8.0" material="Air" />
</module>

<!-- Endcap Tracker Layers -->
<layer id="4" inner_r="185.0" outer_r="478.0" inner_z="626.0" thickness="10.0" nwedges="24">
    <module_parameters r_size="85.333" phi_size_max="80" />
</layer>
```

Detector Description Code

One representation for GEANT4 simulation,

- org.lcsim.geometry.compact.converter.lcdd - SiTrackerBarrel and SiTrackerEndcap convert to lcdd for SLIC (GEANT4)

and a second, consistent one for reconstruction, along with tools for managing and manipulating the detector description:

- org.lcsim.detector.converter.compact - SiTrackerBarrelConverter and SiTrackerEndcapConverter convert to org.lcsim detector/geometry description
- org.lcsim.detector - 54 classes comprising the kernel of detector description and geometry navigation system
- org.lcsim.detector.solids - 21 classes describing geometry objects and operations
- org.lcsim.detector.identifier - 16 classes for handling identification of elements

Silicon Simulation/Digitization

Description of the attributes of silicon sensors (in GeomConverter):

- org.lcsim.detector.tracker.silicon (11 classes)
- will add SiPixels (extends SiSensorElectrodes just as SiStrips)

Description of operation of silicon sensors and readout chips (in lcsim):

- org.lcsim.contrib.SiStripSim (7 classes)
  - CDFSiSensorSim extends SiSensorSim: improved CDF deposition model
  - KPiX extends ReadoutChip: register-level simulation of KPiX including encoding and decoding of ADC values (range bit, etc...)

Code produces org.lcsim.event.base.BaseRawTrackerHit objects

An example driver to produce RawTrackerHits:

- org.lcsim.contrib.RobKutschke.TKNHits.TKNRawHitsDriverV1

Silicon Simulation/Digitization

Clustering code for strip sensors:

- org.lcsim.contrib.SiStripSim.StripClusterMaker (extends ClusterMaker)
- Will add PixelClusterMaker

StripClusterMaker produces org.lcsim.contrib.SiStripSim.SiTrackerHitStrip1D

An example driver to produce TrackerHits:

- org.lcsim.contrib.RobKutschke.TKNHits.TKNClusterMakerDriverV1

TrackerHit Class Heirarchy

- BaseTrackerHit
  - TransformableTrackerHit
    - SiTrackerHit
      - SiTrackerHitStrip1D
      - SiTrackerHitStrip2D
      - SiPixelHit

➡ *Nothing is lost when these types are persisted*

BaseTrackerHit

```
public double[] getPosition()
public double[] getCovMatrix()
public double getdEdx()
public double getTime()
public List<RawTrackerHit> getRawHits()
public int getType()
public Hep3Vector getPositionAsVector()
public SymmetricMatrix getCovarianceAsMatrix()
public Set<SimTrackerHit> getSimHits()
public Set<MCParticle> getMCParticles()
public IDetectorElement getSensor()
public IIdentifierHelper getIdentifierHelper()
```

Derived information is cached on first request to optimize speed

TransformableTrackerHit

Supports transformation of hits into different coordinate systems while protecting user from doing dangerous things with this capability

```
public TrackerHit getTransformedHit(TrackerHitType.CoordinateSystem coordinate_system)

public TrackerHit getTransformedHit(ITransform3D global_to_local)

public boolean isPersistable()

public ITransform3D getLocalToGlobal()

public TrackerHitType.CoordinateSystem getCoordinateSystem()

public TrackerHitType.MeasurementType getMeasurementType()

private ITransform3D getGlobalToHit(TrackerHitType.CoordinateSystem coordinate_system)
```

Allows information about specific type of hits to be persisted

TrackerHitType

Encodes information about TrackerHit into an integer type that is persistable

```
public class TrackerHitType
{
    public enum CoordinateSystem
    {
        GLOBAL,
        SENSOR,
        UNKNOWN;
    }

    public enum MeasurementType
    {
        STRIP_1D,
        STRIP_2D,
        PIXEL;
    }

    private static class Decoder
    {
        private static TrackerHitType decoded(int raw_type)
        private static int encoded(TrackerHitType type)
    }
}
```

SiTrackerHit

Adds some function specific to silicon detectors

```
public SiSensor getSensor()

public SiTrackerIdentifierHelper getIdentifierHelper()

public SiSensorElectrodes getReadoutElectrodes()
```

These last two override BaseTrackerHit functions.

SiTrackerIdentifierHelper

```
public int getModuleValue(IIdentifier id)

public int getSensorValue(IIdentifier id)

public int getSideValue(IIdentifier id)

public int getStripValue(IIdentifier id)

public int getModuleValue(IExpandedIdentifier id)

public int getSensorValue(IExpandedIdentifier id)

public int getSideValue(IExpandedIdentifier id)

public int getStripValue(IExpandedIdentifier id)

public int getModuleIndex()

public int getSensorIndex()

public int getSideIndex()

public int getStripIndex()
```

SiTrackerHitStrip1D

Adds some function specific to strip hits

```
public double getHitLength()

public LineSegment3D getHitSegment()

public Hep3Vector getMeasuredCoordinate()

public Hep3Vector getUnmeasuredCoordinate()
```

Allows access to detailed information for strip hits

SiTrackerHitStrip2D

Adds some function specific to 2D strip hits

```
public List<SiTrackerHitStrip1D> getHits1D()

public boolean isGhost()

public Hep3Vector getPositionWithLineFrom(Point3D origin)

public Hep3Vector getPositionWithDirection(Hep3Vector direction)
```

Connection to 1D hits using LCRelations:
some additional code is needed to automate this process.

More To-do Items

- Standard drivers:
  - Rob's drivers have been helpful use-cases
  - Working now on replacing them with a set of standard drivers for all processing
  - Included will be a "meta-driver" that controls the entire workflow: goal is to make life easy for users and standardize the output
- ***Visualization: see Cosmin's talk next...***
- Maintenance and improvements in response to users

The list is getting very short, time to move to the next steps:

➡ *Track finding and fitting*