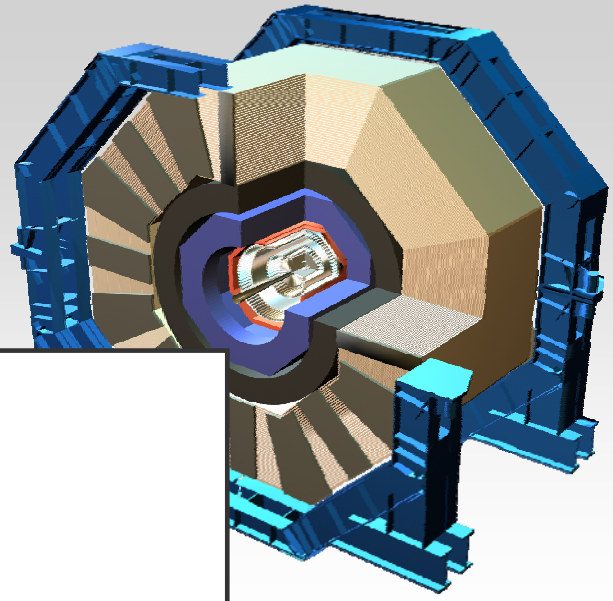


Tracking Toolkit for SiD

Dmitry Onoprienko
Kansas State University

ALCPG 2007 meeting @ Fermilab.
October 22 – 26, 2007.

Preamble

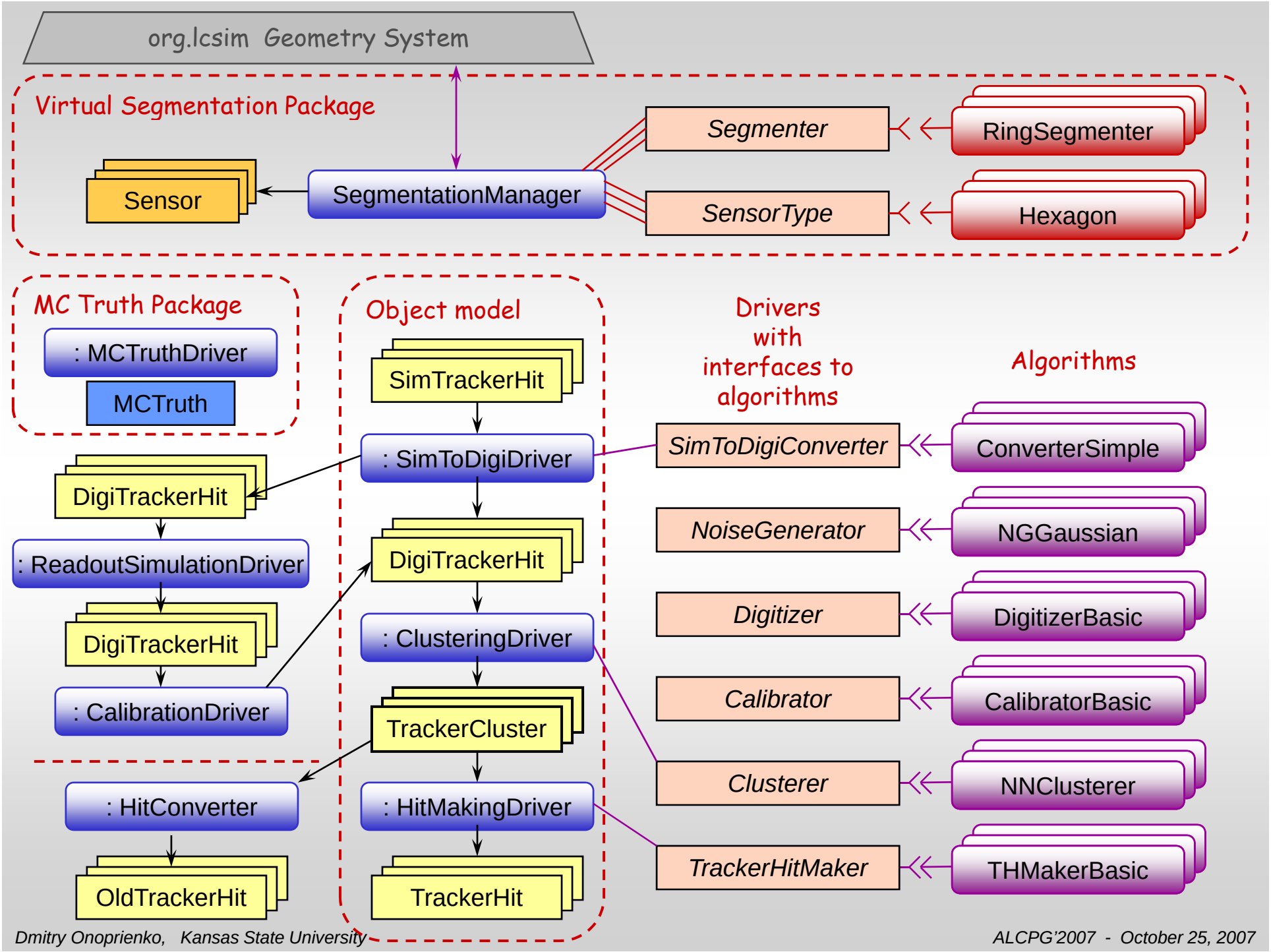
The toolkit was originally written to address the needs of Calorimeter Assisted Tracking algorithm development:

- virtual segmentation
- infrastructure to support non-3D hits in silicon strips
- convenient access to Monte Carlo truth and geometry information

Recently extended to address needs of other groups.

Not the only SiD Tracking Toolkit

- uses custom (LCIO compatible where possible) hit classes
- but can convert to standard org.lcsim TrackerHits
- hope to merge functionality with other packages


org.lcsim Geometry System
Virtual Segmentation Package
Sensor
SegmentationManager
Segmenter
RingSegmenter
SensorType
Hexagon
MC Truth Package
: MCTruthDriver
MCTruth
Object model
SimTrackerHit
: SimToDigiDriver
DigiTrackerHit
: ReadoutSimulationDriver
DigiTrackerHit
: CalibrationDriver
DigiTrackerHit
: ClusteringDriver
TrackerCluster
: HitConverter
OldTrackerHit
: HitMakingDriver
TrackerHit
Drivers with interfaces to algorithms
Algorithms
SimToDigiConverter
ConverterSimple
NoiseGenerator
NGGaussian
Digitizer
DigitizerBasic
Calibrator
CalibratorBasic
Clusterer
NNClusterer
TrackerHitMaker
THMakerBasic
Dmitry Onoprienko, Kansas State University
ALCPG'2007 - October 25, 2007

Virtual Segmentation - Sensor

One Sensor object per virtual segment.

All hit and cluster objects know what Sensor they belong to, and rely on it for geometry related services.

Sensor

{ From geom }

Attributes

private IDetectorElement _de
private int _id
private Hep3Vector _translation
private Transformation3D _rotation

Operations

public Sensor(IDetectorElement de, int id, SensorType type, Hep3Vector translation, Transformation3D rotation)
public IDetectorElement getDetectorElement( )
public SensorType getType( )
public int getID( )
public Hep3Vector getTranslation( )
public Transformation3D getRotation( )
public Hep3Vector localToGlobal(Hep3Vector point)
public Hep3Vector globalToLocal(Hep3Vector point)
public SymmetricMatrix localToGlobal(SymmetricMatrix covMatrix)
public SymmetricMatrix globalToLocal(SymmetricMatrix covMatrix)
public SymmetricMatrix localToGlobal(SymmetricMatrix covMatrix, Hep3Vector position)
public SymmetricMatrix globalToLocal(SymmetricMatrix covMatrix, Hep3Vector position)

Virtual Segmentation - SensorType

<<interface>>

SensorType

int **getChannelID**(Hep3Vector point)

Hep3Vector **getChannelPosition**(int channelID)

Hep3Vector **getChannelDimensions**(int channelID)

boolean **isValidChannelID**(int channelID)

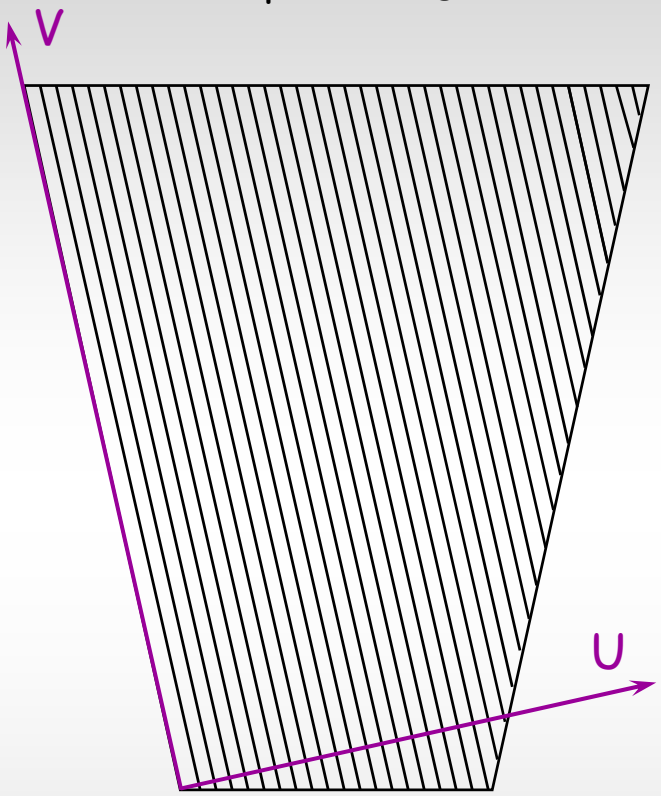
int **getMaxChannelID**()

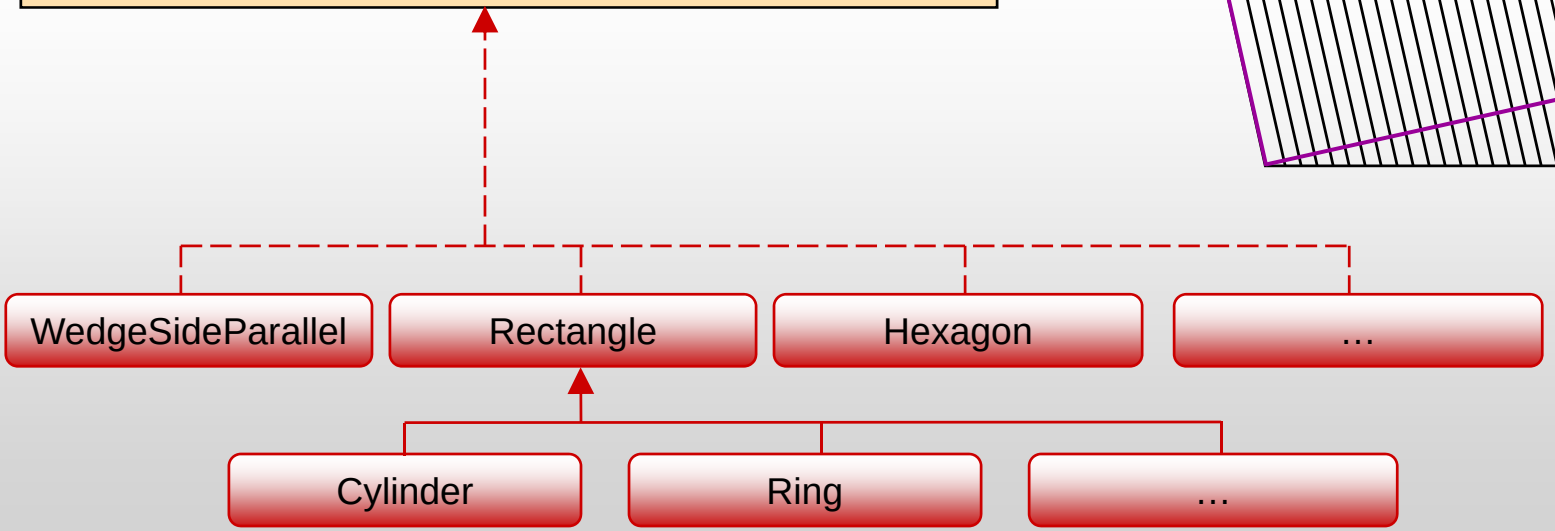
int **getHitDimension**()

int **getNeighbor**(int channelID, int shiftU, int shiftV)

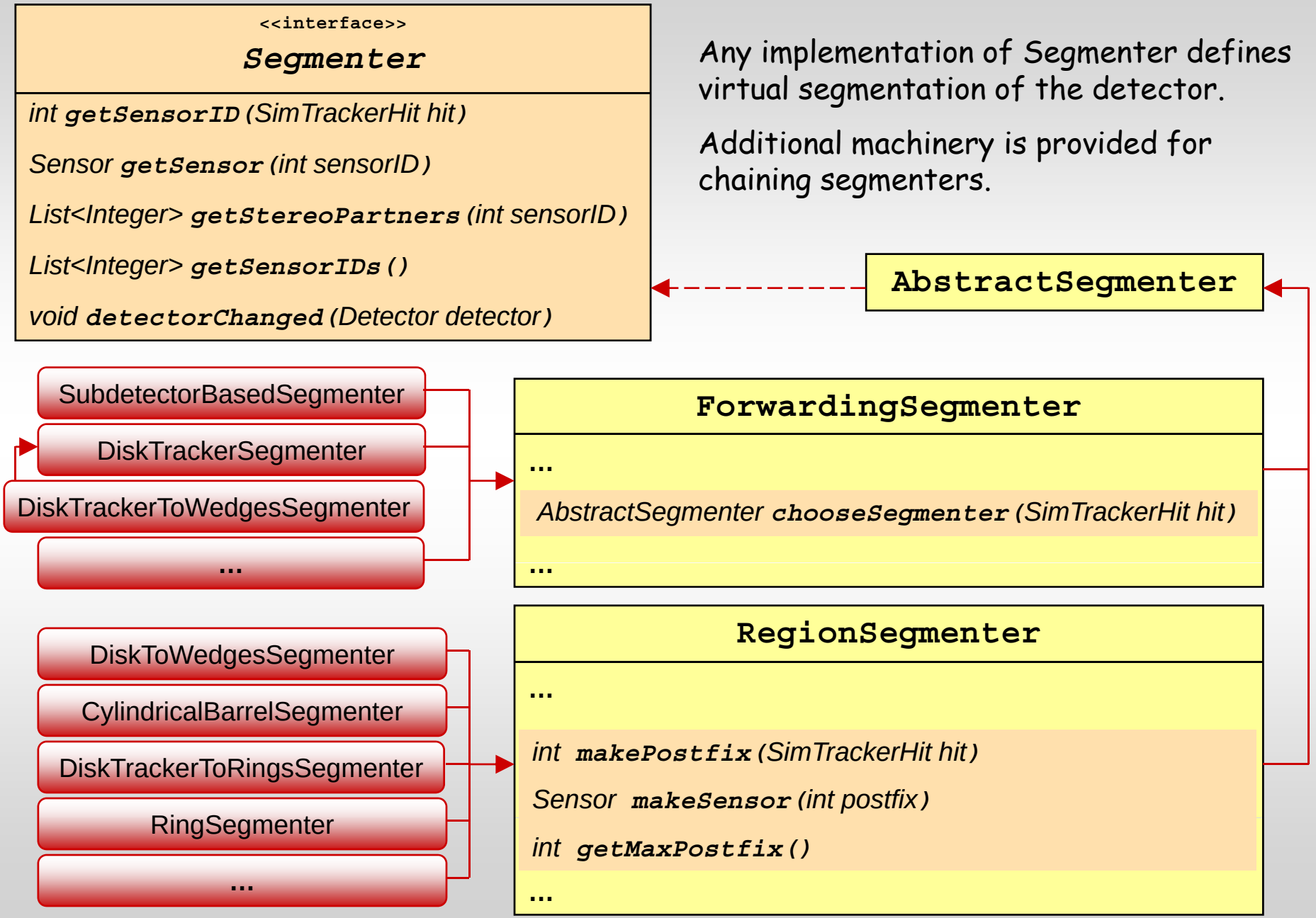
List<Integer> **getNeighbors**(int channelID)

Example: WedgeSideParallel





Virtual Segmentation - Segmenters



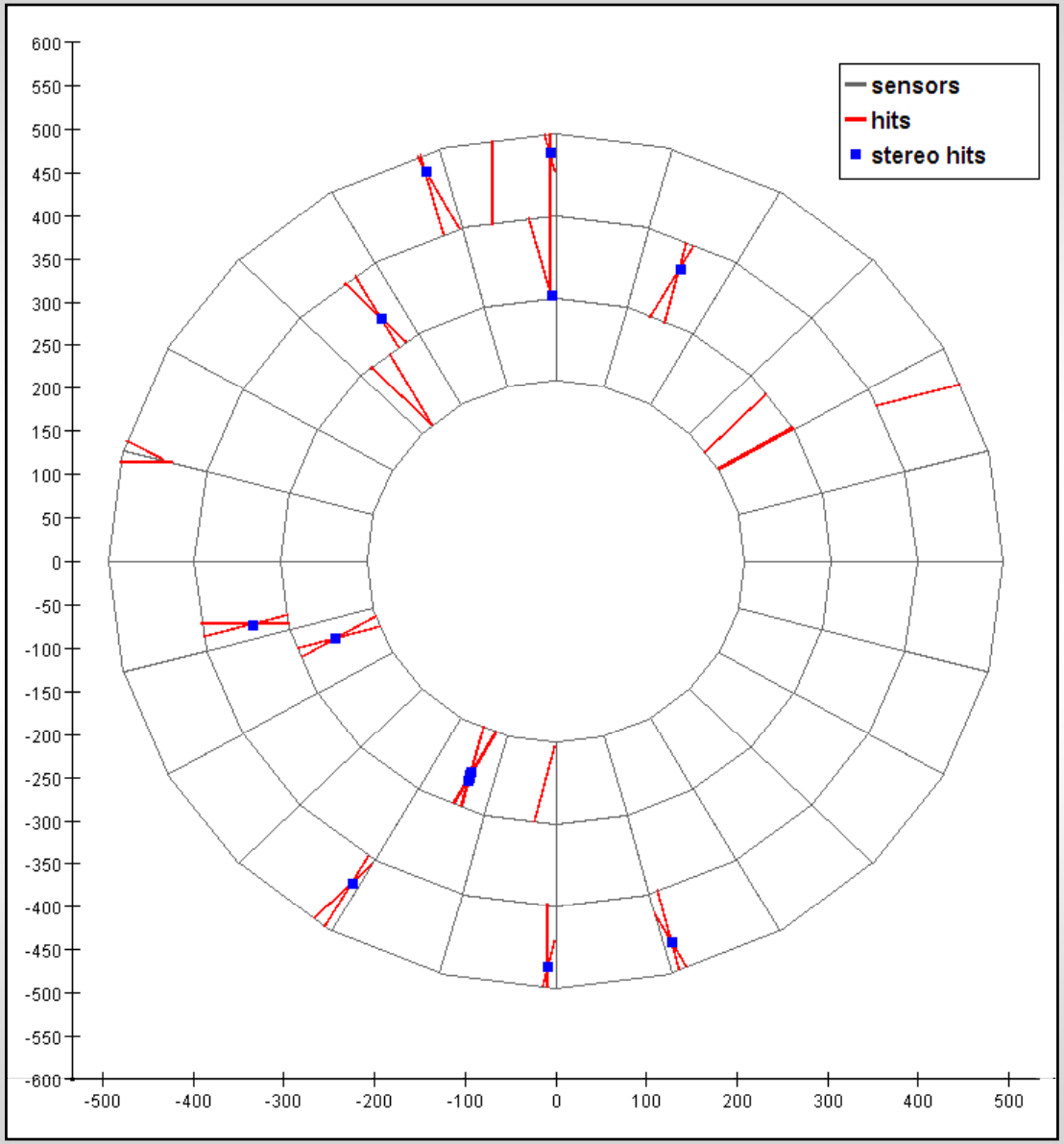
Virtual Segmentation - an Example Segmenter

```
public class ExampleSegmenter extends SubdetectorBasedSegmenter {

 public ExampleSegmenter() {

   // 25 um pixels in VTX Barrel

   CylindricalBarrelSegmenter vtxBarrel = new CylindricalBarrelSegmenter("VertexBarrel");
   vtxBarrel.setStripLength(25.*SystemOfUnits.micrometer);
   vtxBarrel.setStripWidth(25.*SystemOfUnits.micrometer);
   setSegmenter("VertexBarrel", vtxBarrel);

   // 10 cm x 25 um strips in outer Tracker Barrel

   CylindricalBarrelSegmenter trackerBarrel = new CylindricalBarrelSegmenter("TrackerBarrel");
   trackerBarrel.setStripLength(10.*SystemOfUnits.cm);
   trackerBarrel.setStripWidth(25.*SystemOfUnits.micrometer);
   setSegmenter("TrackerBarrel", trackerBarrel);

   // 25 um pixels in VTX Endcap

   DiskTrackerToRingsSegmenter vtxEndcap = new DiskTrackerToRingsSegmenter("VertexEndcap");
   vtxEndcap.setStripLength(25.*SystemOfUnits.micrometer);
   vtxEndcap.setStripWidth(25.*SystemOfUnits.micrometer);
   setSegmenter("VertexEndcap", vtxEndcap);

   // 15 degrees stereo wedges in outer Tracker Endcap

   DiskTrackerToWedgesSegmenter trackerEndcap = new DiskTrackerToWedgesSegmenter("TrackerEndcap");
   trackerEndcap.setNumberOfRadialSlices(new int[]{3,5,8,10, 10});
   trackerEndcap.setStripWidth(25.*SystemOfUnits.micrometer);
   trackerEndcap.setNumberOfPhiSlices(24);
   setSegmenter("TrackerEndcap", trackerEndcap);
 }
}
```

An Example Driver

```
public ExampleDriver() {

   add(new MCTruthDriver());

   // Segmentation description :

   Segmenter segmenter = new ExampleSegmenter();
   SegmentationManager segMan = new SegmentationManager(segmenter);
   SegmentationManager.setDefaultInstance(segMan);
   add(segMan);

   // Digitization :

   SimToDigiConverter converter = new ConverterSimple();
   SimToDigiDriver conversionDriver = new SimToDigiDriver(converter);
   conversionDriver.set("OUTPUT_MAP_NAME", "DigiTrackerHits");
   add(conversionDriver);

   // Clustering :

   ClusteringDriver clusteringDriver = new ClusteringDriver(new NearestNeighborClusterer());
   clusteringDriver.set("INPUT_MAP_NAME", "DigiTrackerHits");
   clusteringDriver.set("OUTPUT_MAP_NAME", "TrackerClusters");
   add(clusteringDriver);

   // Conversion to "standard" org.lcsim.event.TrackerHit :

   TrackerHitConverter hitConverter = new TrackerHitConverter();
   hitConverter.set("INPUT_CLUSTER_MAP_NAME", "TrackerClusters");
   hitConverter.set("OUTPUT_HIT_LIST_NAME", "StandardTrackerHits");
   add(hitConverter);
 }
```

Running the example - Hits



First layer of tracker endcap

ttbar @ 500 GeV

SiD01 segmented by ExampleSegmenter1

Status of the package

All code described is in CVS: org.lcsim.contrib.onoprien.tracking

Infrastructure & hit processing framework are fully functional

Example Drivers are available

Look at org.lcsim.contrib.onoprien.tracking.tests for additional examples.

Many of the algorithms are still simplistic – easy to substitute your own.

Functionality is added as need arises.