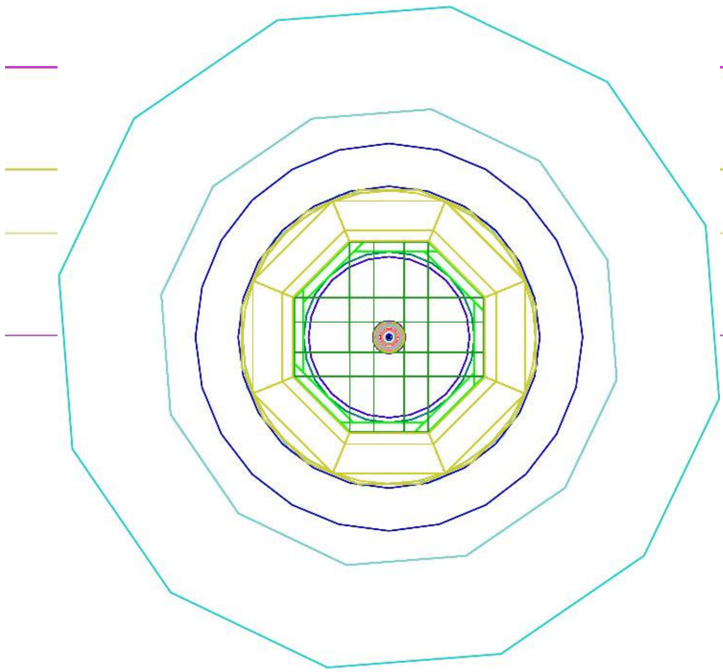
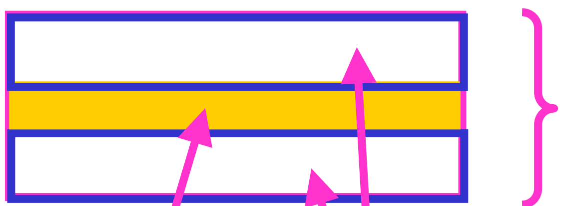
Geant4 Simulation of MAPS

- Geant4/Mokka application has flexible way to change Si thickness, pixel size ☺
- Thickness: default is 500μm, "sensitive" and "physical" equivalent
  - Need to separate these two, initially 20μm sensitive, 480μm substrate (easier comparison with standard simulation)
- … But only 1 x 32-bit int used for encoding "cell ID"
- OK for 1cm^2 pixels, ~4.10^7 in whole detector
- Number of MAPS sensors >2.10^9
  - Need 2 ints

- Want flexibility to study varying pixel size and digitisation efficiently
- Simulation of detector/interactions much slower than digitistion, so 2 stage process
  1. Simulate detector
  2. Implement digitisation as pre-processor to analysis/reconstruction
- Several possibilities…

Alveolus in LDC01/Ecal02





- Simple layer structure
- Sensitive and physical Si equivalent

SimCalorimeterHit

- Public Member Functions

virtual int getCellID0 () const=0

Returns the detector specific (geometrical) cell id.

virtual int getCellID1 () const=0

Returns the second detector specific (geometrical) cell id.

virtual float getEnergy () const=0

Returns the energy of the hit in [GeV].

virtual const float * getPosition () const=0

Returns the position of the hit in world coordinates.

virtual int getNMCParticles () const=0

Returns the number of MC contributions to the hit.

virtual int getNMCContributions () const=0

Returns the number of MC contributions to the hit.

virtual float getEnergyCont (int i) const=0

Returns the energy in [GeV] of the i-th contribution to the hit.

virtual float getTimeCont (int i) const=0

Returns the time of the i-th in [ns] contribution to the hit.

virtual int getPDGCont (int i) const=0

Returns the PDG code of the shower particle that caused this contribution.

virtual MCParticle * getParticleCont (int i) const=0

Returns the MCParticle that caused the shower responsible for this contribution to the hit.

SimTrackerHit

- Public Member Functions

virtual int getCellID () const=0

Returns the detector specific (geometrical) cell id.

virtual const double * getPosition () const=0

Returns the hit position in [mm].

virtual float getdEdx () const=0

Returns the dE/dx of the hit in [GeV].

virtual float getTime () const=0

Returns the time of the hit in [ns].

virtual MCParticle * getMCParticle () const=0

Returns the MC particle that caused the hit.

virtual const float * getMomentum () const=0

Returns the 3-momentum of the particle at the hits position in [GeV] - optional, only if bit LCIO::THBIT_MOMENTUM is set.

Option 0

- Reduce (sensitive detector) pixel size, treat each MAPS sensor ~50x50μm^2 pixel as SimCalorimeterHit
- Need to implement 2 x CellIDs
- Class provides hit position (world coordinate system) at cell centre
  - Problem: need position ~5x5μm^2 to use Giulio's efficiency mapping
  - Had originally planned to apply this in simulation (which *may* have been easier)

Option 1a

- Do not reduce (sensitive detector) pixel size, keep simulated segmentation as 1x1cm^2
- Use SimTrackerHit class for hits in epi-layer
  - Retain exact hit position in LCIO output file
- Apply Giulio's mapping in analysis
- Use the same, single CellID for all MAPSTrackerHits in same 1x1cm^2 pixel (can be used to determine cell centre via CGA)
- Use position as local coordinates in reference frame of 1x1cm^2 pixel
  - Very easy to apply efficiency mapping ☺
  - Need to provide modified methods for e.g. event display tools ☹
  - Need to either use CGA to convert from CellID to world coordinates, or generate associated SimCalorimeterHit in Si substrate
  - Easy to relate individual hits from same pixels (int comparisons)

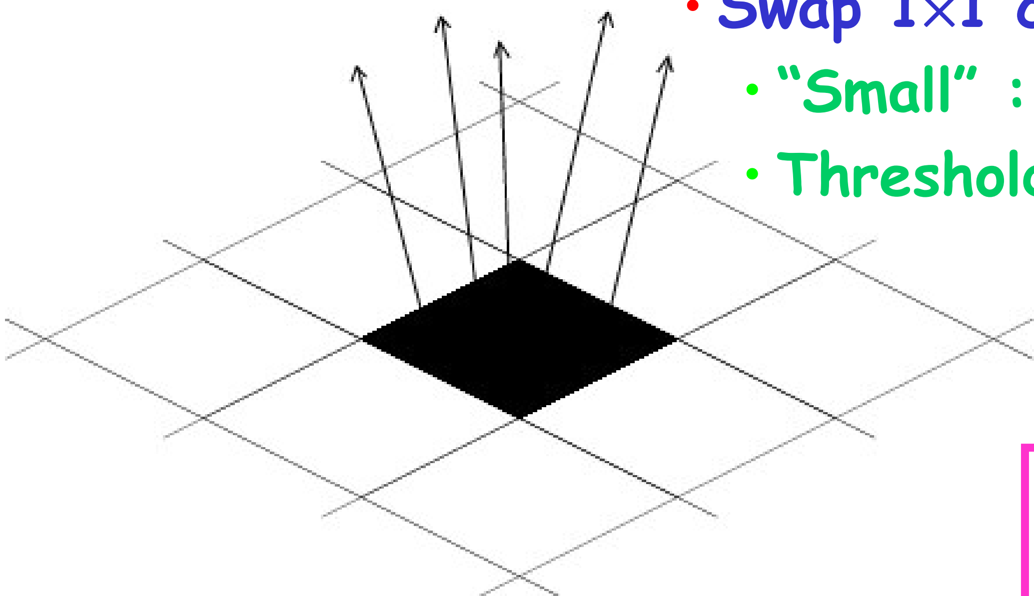
Option 1b

- Do not reduce (sensitive detector) pixel size, keep simulated segmentation as 1x1cm^2
- Use SimTrackerHit class for hits in epi-layer
  - Retain exact hit position in LCIO output file
- Apply Giulio's mapping in analysis
- Use single CellID to define which 5x5μm^2 area track hits
- Use position as world coordinate of hit
  - Very easy to apply efficiency mapping ☺
  - No need to provide modified methods for e.g. event display tools ☹
  - Difficult to relate hits from same MAPS pixel or 1cm^2 pixel – need to know about rotations, etc. of whole detector, many fp comparisons

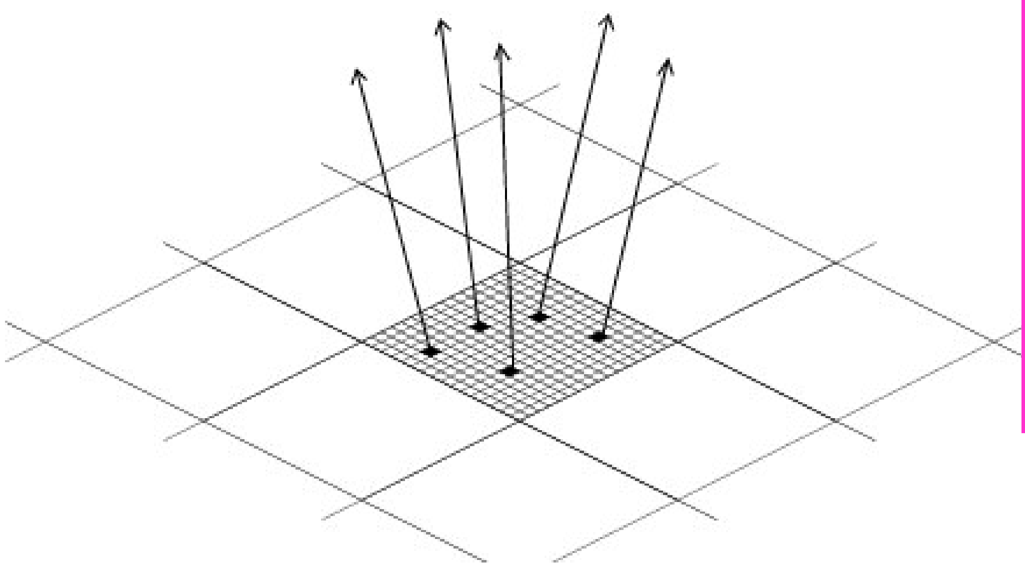
Option 1c

- Do not reduce (sensitive detector) pixel size, keep simulated segmentation as $1x1cm^2$
- Use SimTrackerHit class for hits in epi-layer
  - Retain exact hit position in LCIO output file
- Apply Giulio's mapping in analysis
- Use single CellID to define which $5x5\mu m^2$ area track hits
- Use position as world coordinate of CENTRE OF $1X1cm^2$ CELL
  - Very easy to apply efficiency mapping ☺
  - No need to provide modified methods for e.g. event display tools ☹
  - Less difficult to relate hits from same MAPS pixel, but need to get coordinates of $1cm^2$ cell for each MAPS hit

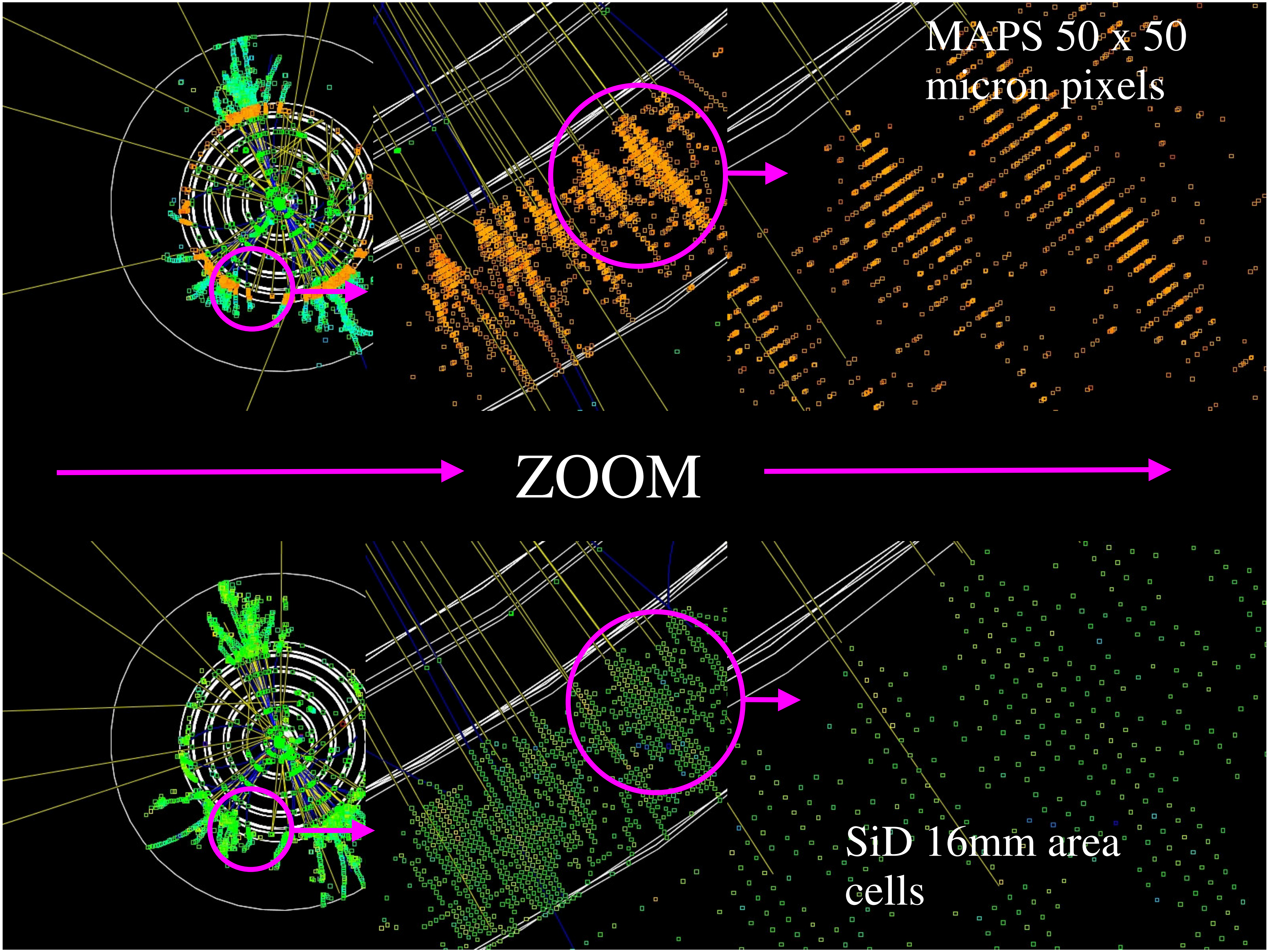
Basic concept for MAPS



- Swap 1×1 cm^2 Si pads with small pixels
  - "Small" := at most one particle/pixel
  - Threshold only/pixel, i.e. Digital ECAL



- How small is small?
  - EM shower core density at 500GeV is ~100/mm^2
  - Pixels must be < 100×100μm^2; working number is 50×50μm^2
  - Gives ~10^{12} pixels for ECAL!

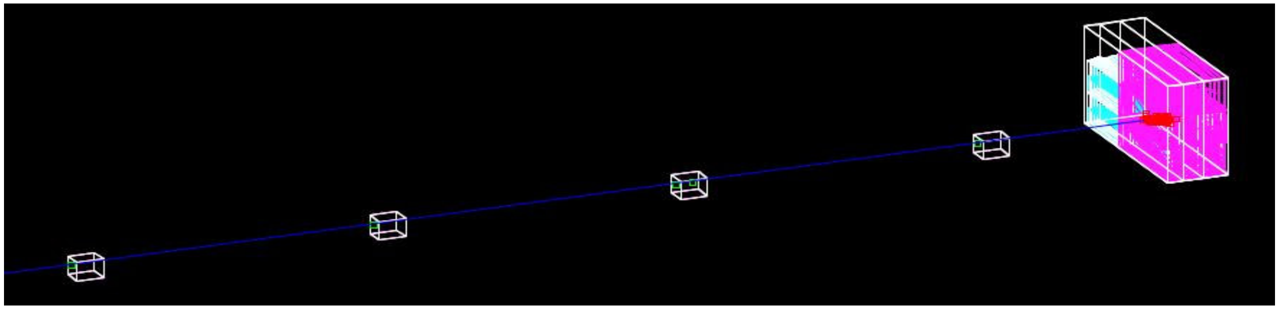

MAPS 50 x 50
micron pixels
ZOOM
SiD 16mm area
cells

Aims/Rationale

- Independent study of MAPS
- Try out evolving North American software suite
  - Event reconstruction framework
  - Easy to adapt geometry and implement MAPS
    - ⇒ SLIC
- Comparison of baseline SiD analogue Si to MAPS ECAL
- SLIC
  - Is well documented and supported
    http://www.lcsim.org/software/slic
- Gets geometry defintion from LCDD format, typically generated from "compact" XML format using GeomConverter, attractive for MAPS study.
- Setting up SLIC is OK
  - Dependences CLHEP, GEANT4, LCPhys, LCIO, Xerces-C++, GDML, LCDD, …

Software Framework

- This study using JAS3/org.lcsim
- Other prototype data analysis summer project (M.Stockton) using
  - George M.'s cleaned+calibrated LCIO files
  - Marlin
  - JAS3 + AIDA + Wired (for event display)



- Conclusion: very easy to use this lightweight framework, well adapted to getting started quickly with little overhead

Implementing MAPS in SiD

- Based on SiD geometry 'cdcaug05',
  - 20 layers @ 0.25cm W, 10 @ 0.5cm W
- Adapt Si thickness to an epitaxial layer thickness of 5µm + 295µm substrate for MAPS

```
<!-- Electromagnetic calorimeter -->

   <detector id="2" name="EMBarrel"
type="CylindricalBarrelCalorimeter" readout="EcalBarrHits">
       <dimensions inner_r = "127.0*cm" outer_z = "182.0*cm"
/>
       <layer repeat="20">
         <slice material = "Tungsten" thickness = "0.25*cm" />
         <slice material = "G10" thickness = "0.068*cm" />
         <slice material = "Silicon" thickness = "0.032*cm"
sensitive = "yes" />
         <slice material = "Air" thickness = "0.025*cm" />
       </layer>
       <layer repeat="10">
         <slice material = "Tungsten" thickness = "0.50*cm" />
         <slice material = "G10" thickness = "0.068*cm" />
         <slice material = "Silicon" thickness = "0.032*cm"
sensitive = "yes" />
         <slice material = "Air" thickness = "0.025*cm" />
       </layer>
   </detector>
```

```
<!-- Electromagnetic calorimeter -->

   <detector id="2" name="EMBarrel"
type="CylindricalBarrelCalorimeter" readout="EcalBarrHits">
       <dimensions inner_r = "127.0*cm" outer_z = "182.0*cm" />
       <layer repeat="20">
         <slice material = "Tungsten" thickness = "0.25*cm" />
         <slice material = "G10" thickness = "0.07*cm" />
         <slice material = "Silicon" thickness = "0.0295*cm" />
         <slice material = "Silicon" thickness = "0.0005*cm" sensitive
                "yes" />
         <slice material = "Air" thickness = "0.025*cm" />
       </layer>
       <layer repeat="10">
         <slice material = "Tungsten" thickness = "0.50*cm" />
         <slice material = "G10" thickness = "0.07*cm" />
         <slice material = "Silicon" thickness = "0.0295*cm" />
         <slice material = "Silicon" thickness = "0.0005*cm" sensitive
=               "yes" />
         <slice material = "Air" thickness = "0.025*cm" />
       </layer>
   </detector>
```