

# Software News

Angela Lucaci-Timoce



## Overview

- 1 New reconstruction
- 2 *calice\_run*
- 3 Other News



# New AHCAL Reconstruction Software

## Default reco

*HcalSROMapper*  
*HcalMappingI*  
*HcalCalibration*  
*HcalMappingII*  
*HcalBadHitFilter*

- Note that *HcalCalibration* processor does not only calibrate, but also obtains: temperature, pedestal calculation, calibrations (cleanly separated in new reco)
- Wrong position of the tiles for rotated detector

## New reco

*HcalSROMapper*  
*HcalMappingProcessor*  
*HcalCellDescriptionProcessor*  
*HcalSiPMTemperatureProcessor*  
*HcalPedestalProcessor*  
*HcalSiPMCalibrationsProcessor*  
*HcalSiPMCalibrateProcessor*  
*HcalPositionProcessor*

- Correct position of tiles
- Based entirely on code developed by **Beni** for his PhD (many thanks)
- Documented (doxygen)

# New AHCAL Reconstruction Software

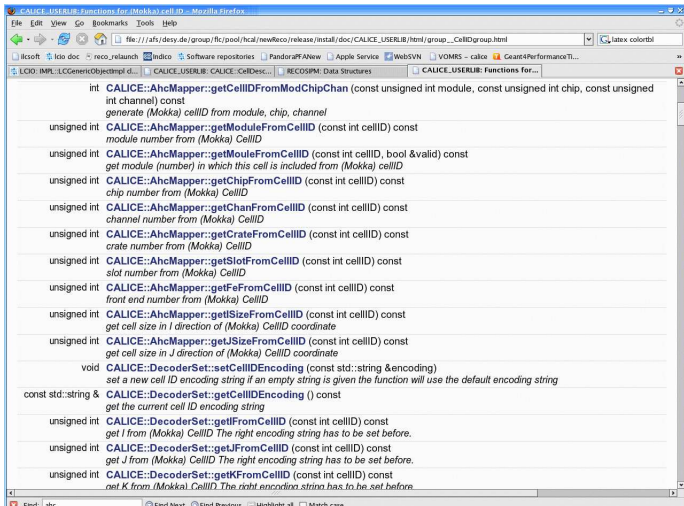
- New directories/packages:  
*calice\_userlib/Mapping*  
*calice\_reco/recoSiPM*  
*calice\_reco/cellGeometry*

## Advantages

- clean code, separated tasks
- documented (doxygen)
- easy to do systematics
- correct position of AHCAL hits: checked by Beni, to be soon shown with event display

# Mapping

- Main working horse: **AhcMapper** (framework allows inclusion of mappers for other detectors)
- Example for Mokka cell ID:



```
int CALICE::AhcMapper::getCellIDFromModChipChan (const unsigned int module, const unsigned int chip, const unsigned
int channel) const
    generate (Mokka) cellID from module, chip, channel

unsigned int CALICE::AhcMapper::getModuleFromCellID (const int cellID) const
    module number from (Mokka) CellID

unsigned int CALICE::AhcMapper::getModuleFromCellID (const int cellID, bool &valid) const
    get module (number) in which this cell is included from (Mokka) cellID

unsigned int CALICE::AhcMapper::getChipFromCellID (const int cellID) const
    chip number from (Mokka) CellID

unsigned int CALICE::AhcMapper::getChanFromCellID (const int cellID) const
    channel number from (Mokka) CellID

unsigned int CALICE::AhcMapper::getCrateFromCellID (const int cellID) const
    crate number from (Mokka) CellID

unsigned int CALICE::AhcMapper::getSlotFromCellID (const int cellID) const
    slot number from (Mokka) CellID

unsigned int CALICE::AhcMapper::getFeFromCellID (const int cellID) const
    front end number from (Mokka) CellID

unsigned int CALICE::AhcMapper::getSizeFromCellID (const int cellID) const
    get cell size in I direction of (Mokka) CellID coordinate

unsigned int CALICE::AhcMapper::getJSizeFromCellID (const int cellID) const
    get cell size in J direction of (Mokka) CellID coordinate

void CALICE::DecoderSet::setCellIDEncoding (const std::string &encoding)
    set a new cell ID encoding string if an empty string is given the function will use the default encoding string

const std::string & CALICE::DecoderSet::getCellIDEncoding () const
    get the current cell ID encoding string

unsigned int CALICE::DecoderSet::getIFromCellID (const int cellID) const
    get I from (Mokka) CellID The right encoding string has to be set before.

unsigned int CALICE::DecoderSet::getJFromCellID (const int cellID) const
    get J from (Mokka) CellID The right encoding string has to be set before.

unsigned int CALICE::DecoderSet::getKFromCellID (const int cellID) const
    get K from (Mokka) CellID The right encoding string has to be set before.
```

# Software - continued

- *HcalCellDescriptionProcessor*:  
a cell is described by position, size and angle
- *HcalSiPMTemperatureProcessor*: provides the SiPM temperature
- *HcalSiPMCalibrationsProcessor*: a SiPM has MIP/gain calibrations, pedestal, inter-calibration, saturation (the same as the one used in the default reco), quality flag

- *HcalSiPMCalibrateProcessor*: one formula

$\text{calibratedEnergy} = \text{saturationContainer} \rightarrow \text{deSaturate}(\text{rawEnergy} - \text{pedestal}) *$

$* \text{interCalibrationValue} / \text{gainValue}) * \text{gainValue} / \text{interCalibrationValue} / \text{mipValue}$

- *HcalPositionProcessor*: correct position of AHCAL hits

# AHCAL Cell Position

- Basic assumptions: save values in the data base for a  $0^\circ$  configuration, and calculate the rest for a rotation angle  $\theta$
- Cell position is calculated with:

$$\vec{r}_{detector} + \vec{r}_{module} \cdot \mathcal{M}_{zshift} + \vec{r}_{cell} \cdot \mathcal{M}_{rotation}$$

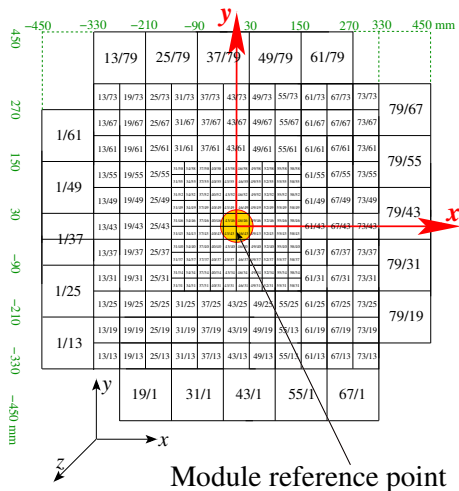
where

$$\mathcal{M}_{zshift} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1/\cos\theta \end{pmatrix}$$

and

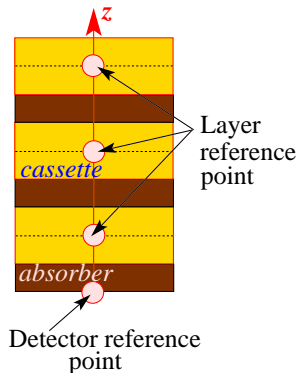
$$\mathcal{M}_{rotation} = \begin{pmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{pmatrix}$$

# AHCAL Cell Position - continued



- $\vec{r}_{detector}$ : based on *DetectorTransformation* (*Hcal/HcalDetectorPosition*), which contains the rotation angle and the global position of the detector front face center
- $\vec{r}_{module}$ : based on *ModuleLocation* (*Hcal/HcalModuleLocation*), which contains the positions at which the modules center will be placed for a detector at  $0^\circ$ , with respect to the center of the detector's front face

# AHCAL Cell Position - continued

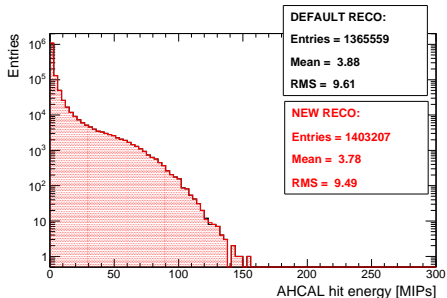
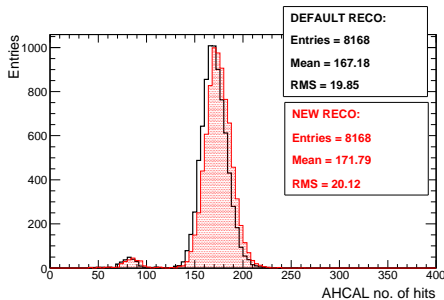
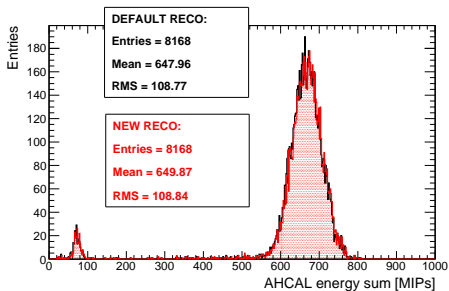


- $\vec{r}_{cell}$ : based on *ModuleDescription* (*Ahc/ModuleDescription*), which contains the cell position with respect to the center of the cassette

- Many thanks to Nils, who updated the database folders



# Comparison: Default vs New Reco



Small effect due to cut of 0.4 MIPs:

- in default reco: BEFORE saturation correction
- in new reco: AFTER saturation correction  
⇒ a few more hits at low energy

## Default Digitisation

*HcalSRMapper*  
*HcalMCGanging*  
*HcalMappingI*  
*HcalTcmtAppendNoise*  
*HcalMCDigitisation*  
*HcalMCCalibration*  
*HcalMappingII*  
*HcalBadFilter*

## New Digi

- To be developed
- In discussion with Katja (Munich)

- Switch from old ASCII steering files to recommended XML type steering files
- Reconstruction: `../generateRecoXml < run number >`
- **New** reconstruction: `../generateRecoSiPMXml < run number >`
- Noise: `../generateNoiseXml < run number >`
- Digitisation: `../generateDigiXml < run number > < mc.list > < noise file >]`

## Other News

- New processor: [AppendMultiAmplitude](#), from Nils and Beni, to append the amplitude of the multiplicity counter to the event (once can later cut on it to reject double particles)

## Last but not least

- Latest software changes available in the moment only in the [DESY GIT repositories](#)
- To be soon released in the SVN