Table of Contents

Heavy Ion Photon reconstruction

Complete: 🟩

Goal of this page

The aim of this page is to document the photon reconstruction used in heavy ion events, where the code lives, how to run it, the input and output, and details of the implementation.

Code and tags

The code lives in the package RecoHI/HiEgammaAlgos.

The most recent version of the code will be tagged in CMSSW_3_4_0_pre4.

Before CMSSW_3_4_x release, if you want to check out the latest working version yourself, you can check out the release by the following command:

```
scramv1 project CMSSW CMSSW_3_3_0
cd CMSSW_3_3_0/src
cmsenv
cvs co -r V00-00-06 RecoHI/HiEgammaAlgos
```

and compile:

```
scramv1 b
```

This will prepare the working environment for you.

Configuration files

The reconstruction sequence is defined in RecoHI/HiEgammaAlgos/python/HiEgamma_cff.py.

How to run the photon reconstruction

Do you need to run the photon reconstruction?

By default the sequences `hiEcalClusters` and `hiEgammaSequence` are called in RecoHI/Configuration/python/Reconstruction_HI_cff.py during the standard heavy ion reconstruction process. If you also want to include the Egamma isolation variables, one can include the `hiEgammaIsolationSequence` in the sequence to produce heavy ion isolation maps. (This sequence is run by default when producing the heavy ion PAT photons)

Running the photon reconstruction only

In your configuration file, include the HiEgammaAlgos_cff.py by

```
# Egamma
process.Load("RecoHI.HiEgammaAlgos.HiEgamma_cff")
```

and put `hiEcalClusters` and `hiEgammaSequence` into your cms.Sequence. This will reconstruct photons with heavy-ion-specific algorithms up to AOD photon level.

Running the full heavy ion reconstruction

In your configuration file, include the Reconstruction_HI_cff.py by

```
process.Load("RecoHI.Configuration.Reconstruction_HI_cff")
```

and put `globalRecoPbPb` into your cms.Sequence.

PAT Photon reconstruction

After running the full heavy ion reconstruction, one can run the heavy ion PAT photon reconstruction as part of the heavy-ion PAT. To do this include CMS.PhysicsTools/PatAlgos/python/patHeavyIonSequences_cff.py in your configuration file:

```
process.Load("CMS.PhysicsTools.PatAlgos.patHeavyIonSequences_cff")
```

and put `patHeavyIonDefaultSequence` into your cms.Sequence.

This photon part of this sequence will process the reconstructed photon and embed the heavy ion photon isolation information into pat::Photon as user floats. At the same time, p+p default photon ID and isolation are also stored for comparisons / studies.

Input

The `hiEcalClusters` sequence requires as input the RecHits in the barrel and endcap of the electromagnetic calorimeter (ECAL).

Product type	Module label	Product instance label	Description
EcalRecHitCollection	ecalRecHit	EcalRecHitsEB	EcalRecHits in the barrel
EcalRecHitCollection	ecalRecHit	EcalRecHitsEE	EcalRecHits in the endcaps

The `hiEgammaIsolationSequence` requires as input the BasicClusters produced as an intermediate step of the `hiEcalClusters` sequence, as well as the RecHits in the HCAL and the Track Collection, to determine if a cluster in the ECAL is isolated.

Product type	Module label	Product instance label	Description
BasicClusterCollection	islandBasicClusters	islandBarrelBasicClusters	BasicClusters produced with the Island algorithm from EcalRecHits in the barrel
BasicClusterCollection	islandBasicClusters	islandEndcapBasicClusters	BasicClusters produced with the Island algorithm from EcalRecHits in the endcap
HFRecHitCollection	hfreco		RecHits in the forward Hadronic Calorimeter (HF)
HORecHitCollection	horeco		RecHits in outer Hadronic Calorimeter (HO)
HBHERecHitCollection	hbhereco		RecHits in the barrel and endcaps of the HCAL
TrackCollection	hiGlobalPrimTracks		Reconstructed tracks
reco::PhotonCollection	photons		Reconstructed reco::Photons

Output

The output of the `hiEgammaSequence` is

Product type	Module label	Description
reco::PhotonCollection	photons	Reconstructed photons

The output of the `hiEgammaIsolationSequence` is

Product type	Module label	Description
edm::ValueMap<float>	isoCCx	See Heavy ion specific isolation variables below for a description of these variables
edm::ValueMap<float>	isoCRx	
edm::ValueMap<float>	isoTxy	
edm::ValueMap<float>	isoRxy	

Note that *x* and *y* above are placeholders for the parameters used in calculating the isolation variables; the actual labels of the ValueMaps are isoCC1, isoCC2 etc. Also note that the values contained in these ValueMaps are embedded in the individual pat::Photons by the `patHeavyIonDefaultSequence`, so the ValueMaps do not need to be used directly. An example of how to access these variables is shown in the section Accessing the heavy ion photons isolation variables below.

The output of the photon part of the `patHeavyIonDefaultSequence` is

Product type	Module label	Description
std::vector<pat::Photon>	selectedLayer1Photons	PAT photons

Despite the name, no selection is performed on these photons. Since CMSSW_3_3_0 the selector in CMS.PhysicsTools/PatAlgos/python/selectionLayer1/photonSelector_cfi.py has been empty. So the collection `selectedLayer1Photons` is identical to `allLayer1Photons`.

Heavy ion specific isolation variables:

The heavyion specific isolation variables are categorized as four types:

- isoCCx: background subtracted isolation cone energy in ECAL with cone size x*0.1
- isoCRx: background subtracted isolation cone energy in HCAL with cone size x*0.1
- isoTxy: number of reconstructed tracks in a cone of x*0.1 with pt > y * 0.4 GeV/c
- isoRxy: in a cone size of isoRxy, the number of reconstructed tracks with pt > y * 0.4 GeV/c is smaller than x.

They are accessible in the PAT level as user floats within the pat::Photon. Before the PAT level, one can access them directly as value maps to AOD Photons.

Accessing the heavy ion photons isolation variables

A test module is prepared in RecoHI/HiEgammaAlgos/test/patTest/PATHIPhotonTestModule.cc, which works on reconstructed heavy-ion PAT Photons. The main features of this module are explained below.

One can access the PAT photon information with the following code:

```
edm::Handle<edm::View<pat::Photon> > photons;
iEvent.getByLabel("selectedLayer1Photons",photons);
std::auto_ptr<std::vector<pat::Photon> > output(new std::vector<pat::Photon>());
```

and then loop over the photon candidates by:

```
  for (edm::View<pat::Photon>::const_iterator photon = photons->begin(), end = photons->end(); ph
 ...
 ...
 }
```

Inside the loop, one can access the embedded isolation variables by

```
   Float_t isoCC1 = photon->userFloat("isoCC1");
   Float_t isoCR1 = photon->userFloat("isoCR1");
   Float_t isoT12 = photon->userFloat("isoT12");
   Float_t isoDR23 = photon->userFloat("isoDR23");
```

for your own analysis.

Implementation details

Compared to the p+p photon reconstruction algorithm, the photon reconstruction in heavy ion collisions uses the Island SuperCluster algorithm for both barrel and endcap regions while in p+p one uses the hybrid algorithm for the barrel and multi5x5 algorithm for the endcap. This choice is found to cache less background contribution compared to hybrid and multi5x5 algorithm. Also the primary vertex for photon reconstruction is that produced by the heavy ion vertex algorithm.

Further information

- SWGuideEgamma - pp photon information

Contact

- **Hypernews fora**:
    - Software issues: https://hypernews.cern.ch/HyperNews/CMS/get/hiswDevelopment.html, (<hn-cms-hiswDevelopment@cern.ch>)
    - General heavy ions issues https://hypernews.cern.ch/HyperNews/CMS/get/hi.html, (<hn-cms-hi@cern.ch>)
- **Contacts/Developers**: Yen Jie Lee

Review status

Reviewer/Editor and Date	Comments
YenJieLee - 17 Nov 2009	Completed documentation as part of documentation review
PhilipAllfrey - 24 Nov 2009	Expanded

Responsible: YenJieLee

This topic: CMSPublic > SWGuideHeavyIonPhotonReco
Topic revision: r8 - 2009-11-24 - PhilipAllfrey