31-1-2019

Group Meeting

Towards an event reconstruction software for SFGD

C. Jesús-Valls
cjesus@ifae.es

INTRODUCTION

Last week:

Unpacking structure:

Tree of Events that contain a collection of 2D hits.

Full access to all data from electronics

No interface to convert →
Unpacking to Reco

Reco structure:

RecoEvents that contain a collection of 3D voxels.

It provided:

- 2D to 3D reconstruction,
- visualization tools
- basic clustering methods

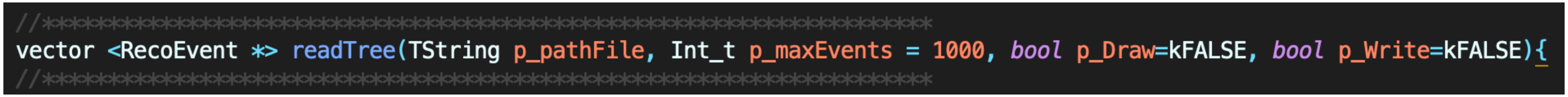
This week goals:

Merge both working platforms.

- Keep simplicity of the reconstruction package.
- Give access to all the information in hits.
- Make the package suitable to 'copy and paste' already developed studies.

It directly reads newDataStructure.root:

A new function has been created, called readFiles():



```
//*********************************************************************
vector <RecoEvent *> readTree(TString p_pathFile, Int_t p_maxEvents = 1000, bool p_Draw=kFALSE, bool p_Write=kFALSE){
//*********************************************************************
```

/folder/to/file_newDataStructure.root

Show each reconstructed event?

num of events to be reconstructed

Print some extra info on the command shell

It returns a vector of pointers to the reconstructed events. Each one contains:

- A vector of all 3D <Voxel*>.
- A vector of all continuous clusters <ProtoTracks*>.

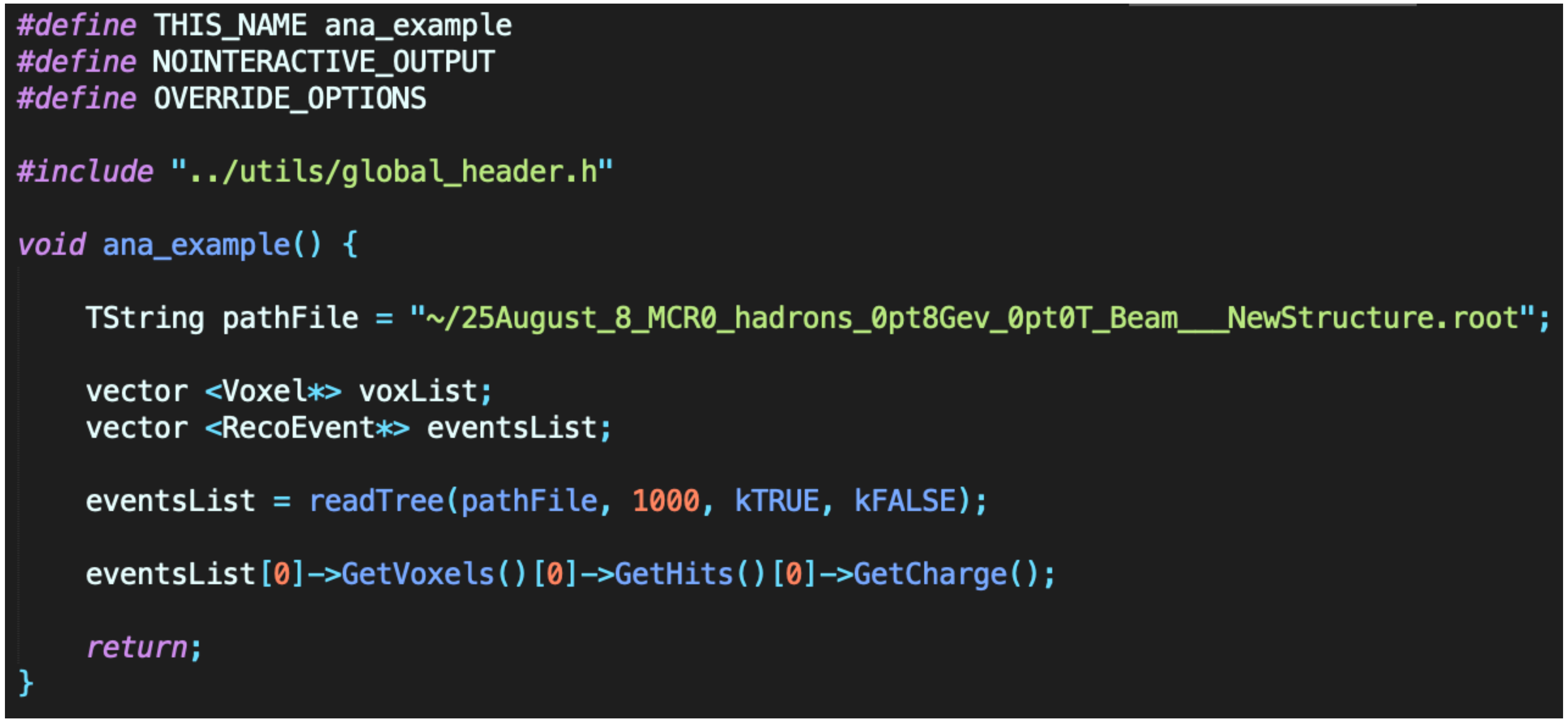
Each Voxel contains:

- A 3D vector <Hit*> that contains pointers to each one of the different hits used to reconstruct it.

vector[0] is the viewXY, vector[1] is the viewXZ, vector[2] is the viewYZ

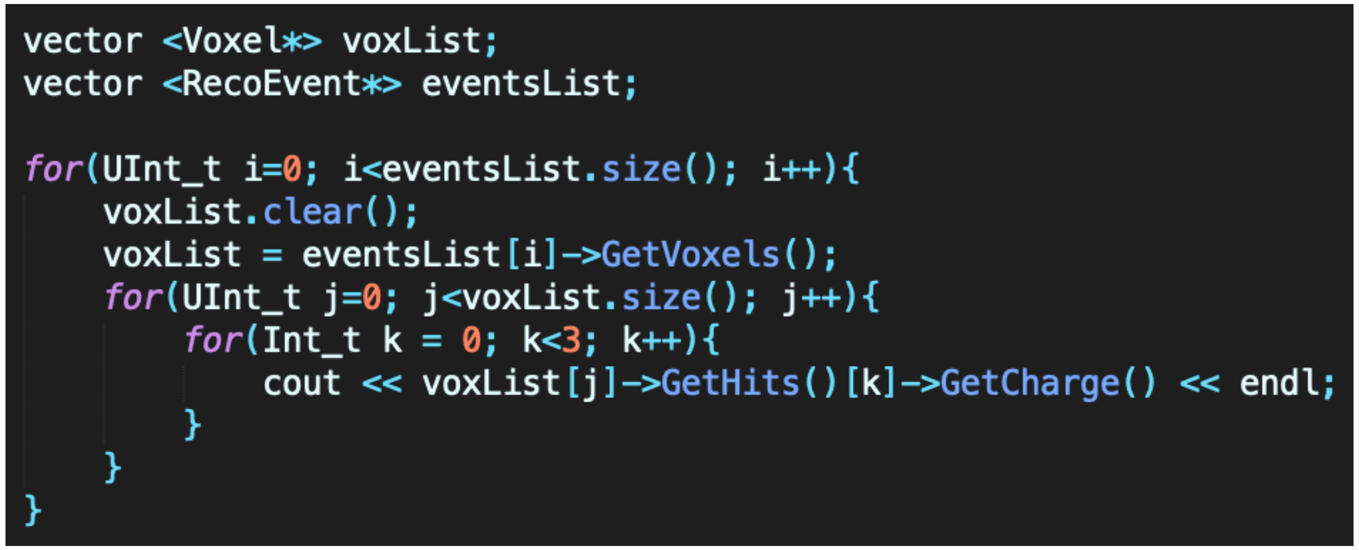
NEW RECONSTRUCTION FEATURES

How to access the information?



```
#define THIS_NAME ana_example
#define NOINTERACTIVE_OUTPUT
#define OVERRIDE_OPTIONS

#include "../utils/global_header.h"

void ana_example() {

    TString pathFile = "~/25August_8_MCR0_hadrons_0pt8Gev_0pt0T_Beam___NewStructure.root";

    vector <Voxel*> voxList;
    vector <RecoEvent*> eventsList;

    eventsList = readTree(pathFile, 1000, kTRUE, kFALSE);

    eventsList[0]->GetVoxels()[0]->GetHits()[0]->GetCharge();

    return;
}
```

In a loop:

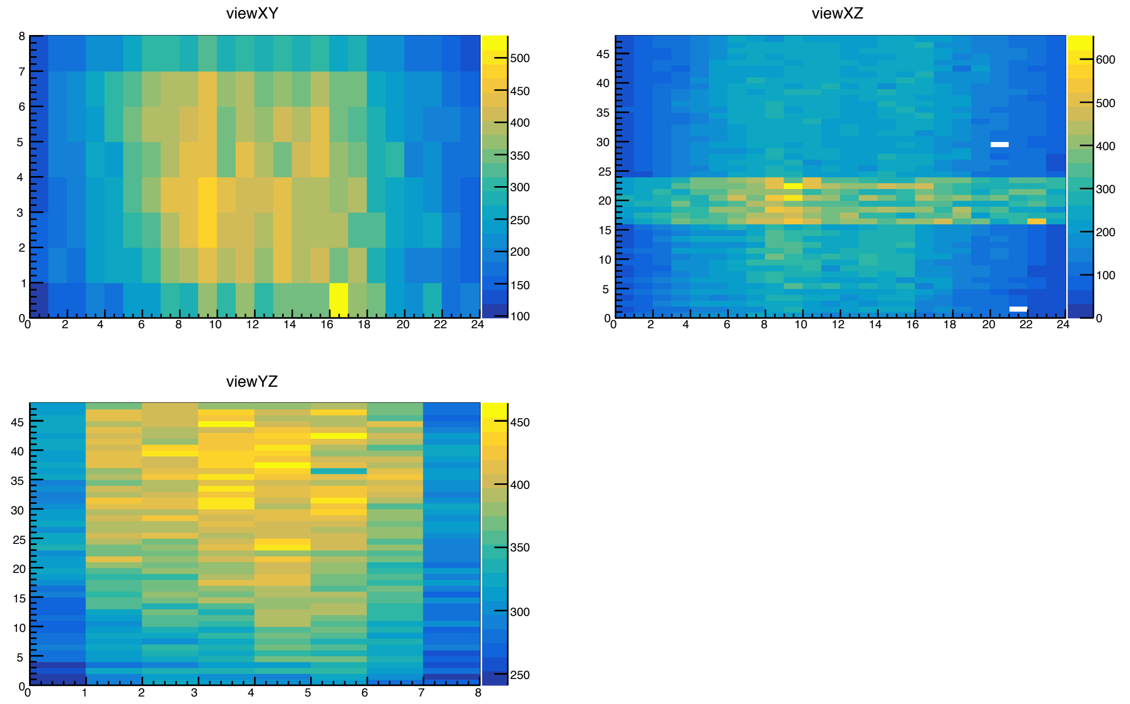


```
vector <Voxel*> voxList;
vector <RecoEvent*> eventsList;

for(UInt_t i=0; i<eventsList.size(); i++){
    voxList.clear();
    voxList = eventsList[i]->GetVoxels();
    for(UInt_t j=0; j<voxList.size(); j++){
        for(Int_t k = 0; k<3; k++){
            cout << voxList[j]->GetHits()[k]->GetCharge() << endl;
        }
    }
}
```

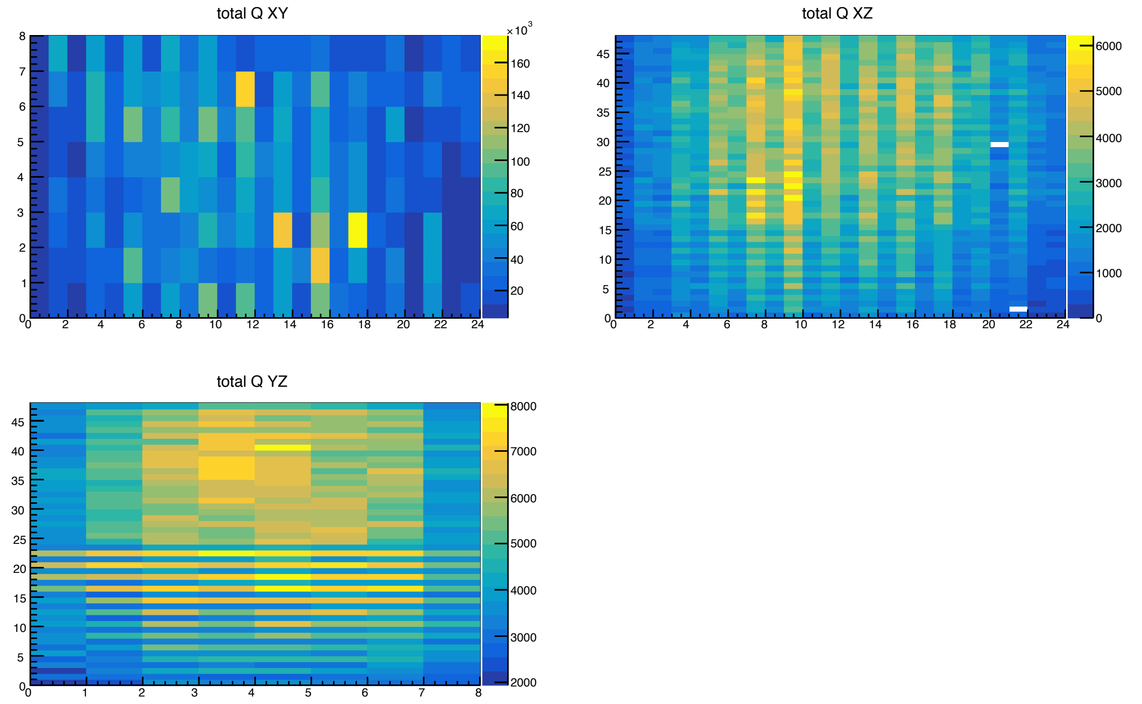
However… we have many not real voxels (duplicated) hits. Davide suggest to use MonteCarlo to better develop 2D to 3D reconstruction.

THE RECONSTRUCTION PACKAGE: MC EXAMPLES

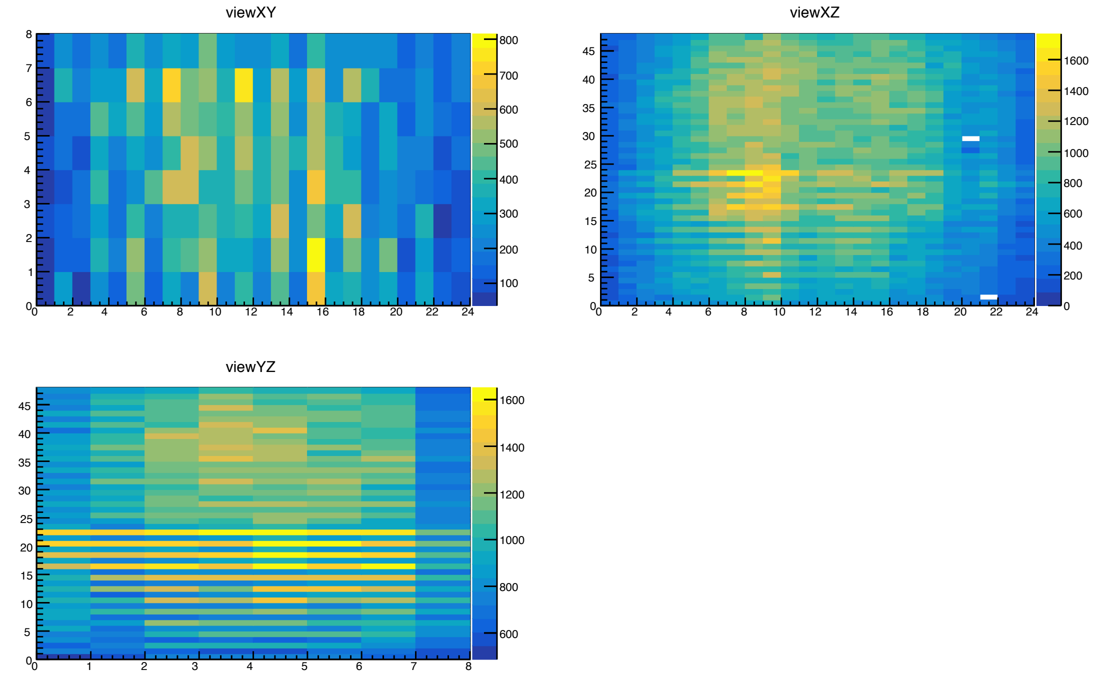
original



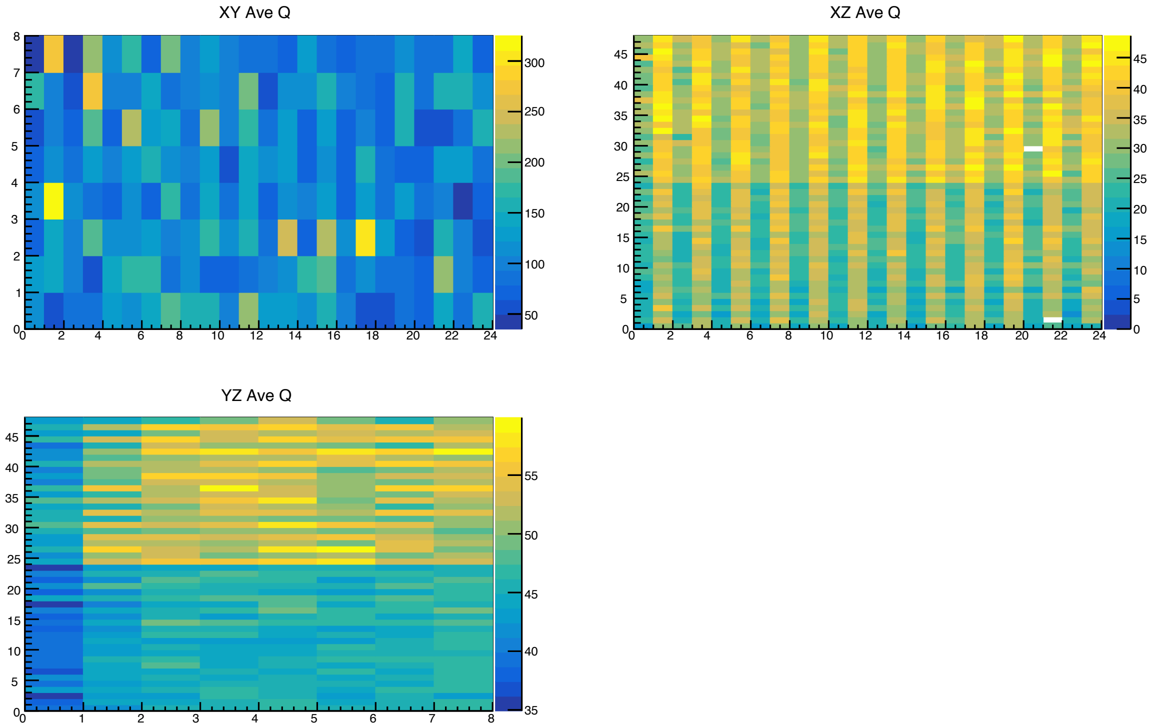
Q



selected



Ave Q


XY Ave Q
XZ Ave Q
YZ Ave Q