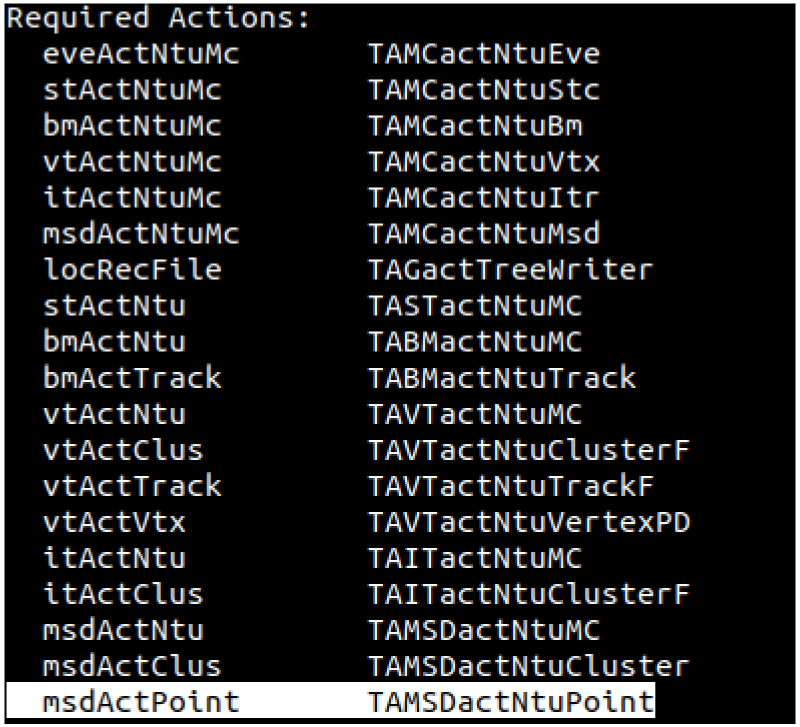
Update on MSD in SHOE and reconstruction

Riccardo Ridolfi
riccardo.ridolfi@bo.infn.it

Vidyo Meeting – 27 January 2020

New action TAMSDactNtuPoint

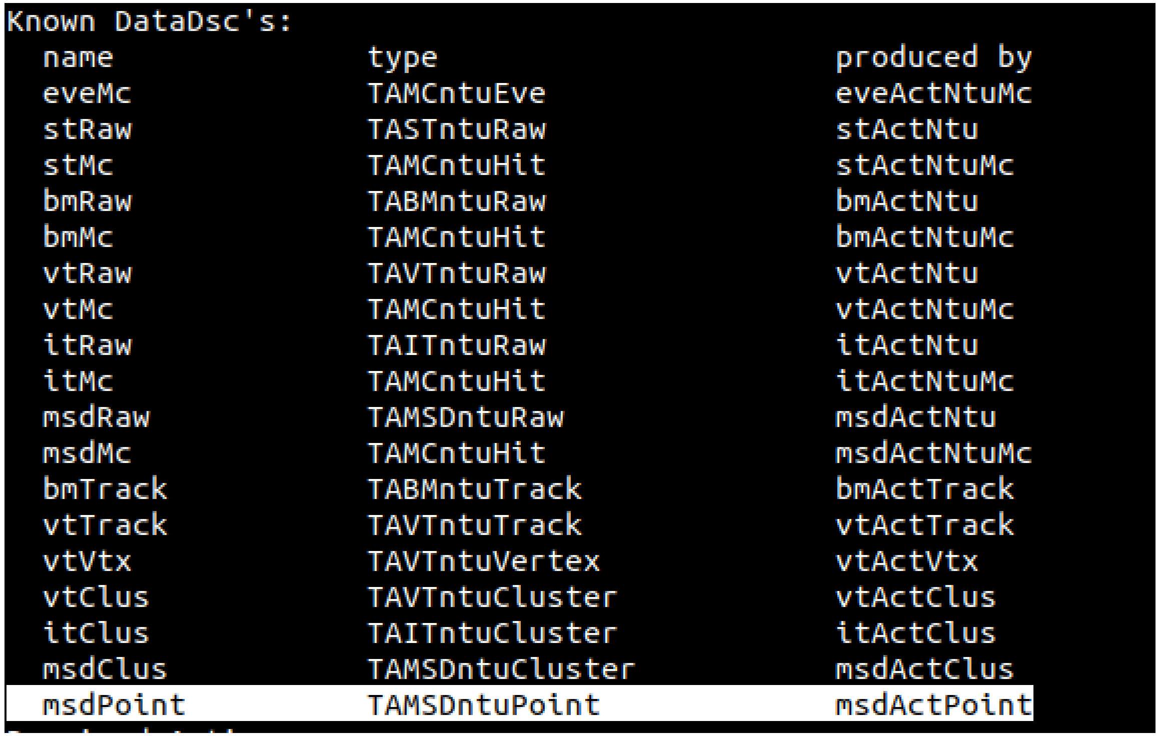


```
Required Actions:
  eveActNtuMc        TAMCactNtuEve
  stActNtuMc         TAMCactNtuStc
  bmActNtuMc         TAMCactNtuBm
  vtActNtuMc         TAMCactNtuVtx
  itActNtuMc         TAMCactNtuItr
  msdActNtuMc        TAMCactNtuMsd
  locRecFile         TAGactTreeWriter
  stActNtu           TASTactNtuMC
  bmActNtu           TABMactNtuMC
  bmActTrack         TABMactNtuTrack
  vtActNtu           TAVTactNtuMC
  vtActClus          TAVTactNtuClusterF
  vtActTrack         TAVTactNtuTrackF
  vtActVtx           TAVTactNtuVertexPD
  itActNtu           TAITactNtuMC
  itActClus          TAITactNtuClusterF
  msdActNtu          TAMSDactNtuMC
  msdActClus         TAMSDactNtuCluster
  msdActPoint        TAMSDactNtuPoint
```

It takes hits from two layer of the same tracking station and combine them (more in the following)

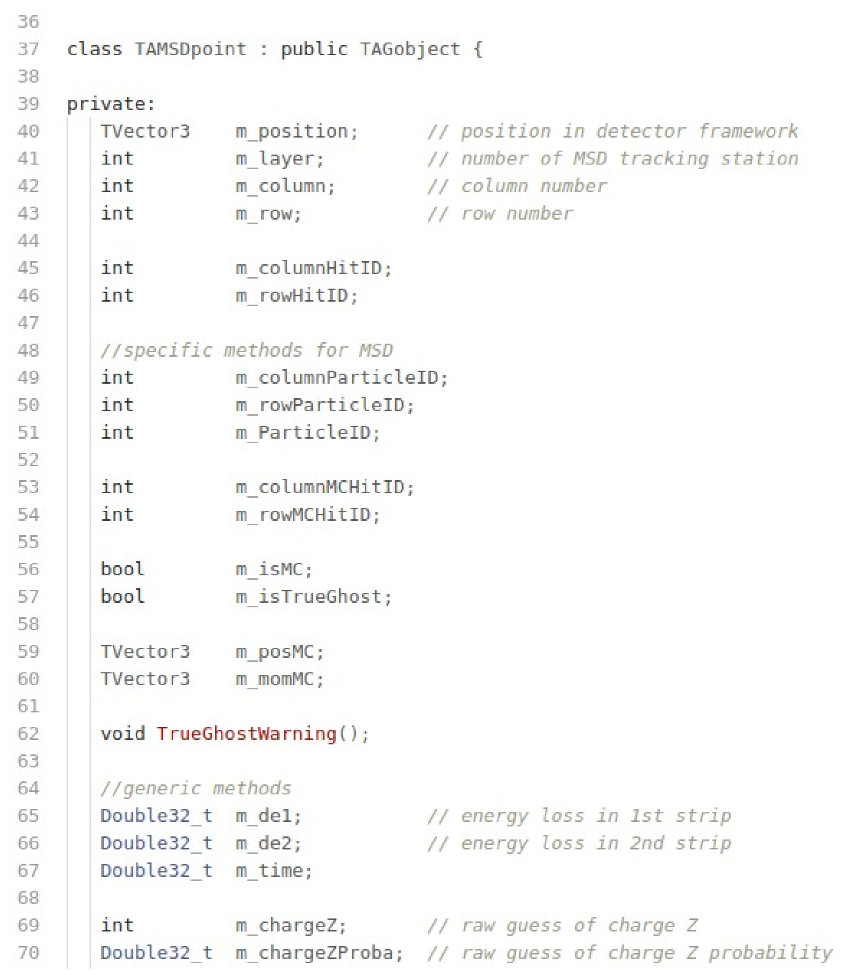
Clusters will be added as soon as we have a first hint of the digitization step

New classes TAMSD(ntu)Point

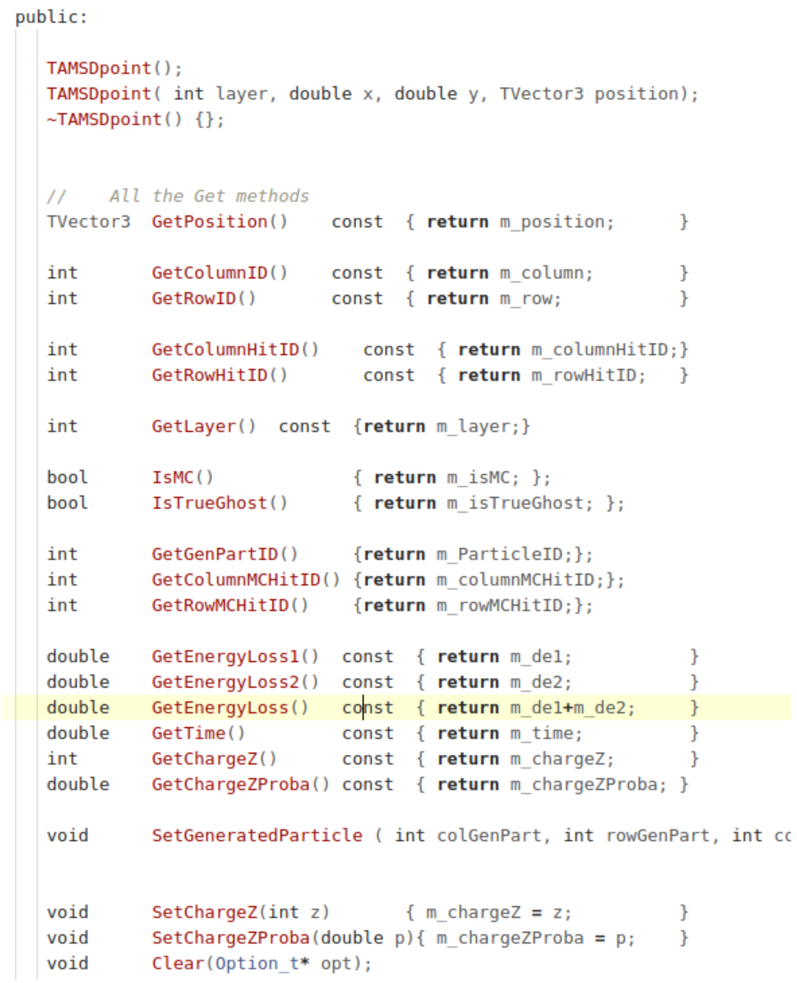


```
Known DataDsc's:
  name                type                      produced by
  eveMc               TAMCntuEve                eveActNtuMc
  stRaw               TASTntuRaw                stActNtu
  stMc                TAMCntuHit                stActNtuMc
  bmRaw               TABMntuRaw                bmActNtu
  bmMc                TAMCntuHit                bmActNtuMc
  vtRaw               TAVTntuRaw                vtActNtu
  vtMc                TAMCntuHit                vtActNtuMc
  itRaw               TAITntuRaw                itActNtu
  itMc                TAMCntuHit                itActNtuMc
  msdRaw              TAMSDntuRaw               msdActNtu
  msdMc               TAMCntuHit                msdActNtuMc
  bmTrack             TABMntuTrack              bmActTrack
  vtTrack             TAVTntuTrack              vtActTrack
  vtVtx               TAVTntuVertex             vtActVtx
  vtClus              TAVTntuCluster            vtActClus
  itClus              TAITntuCluster            itActClus
  msdClus             TAMSDntuCluster           msdActClus
  msdPoint            TAMSDntuPoint             msdActPoint
```

New classes TAMSD(ntu)Point

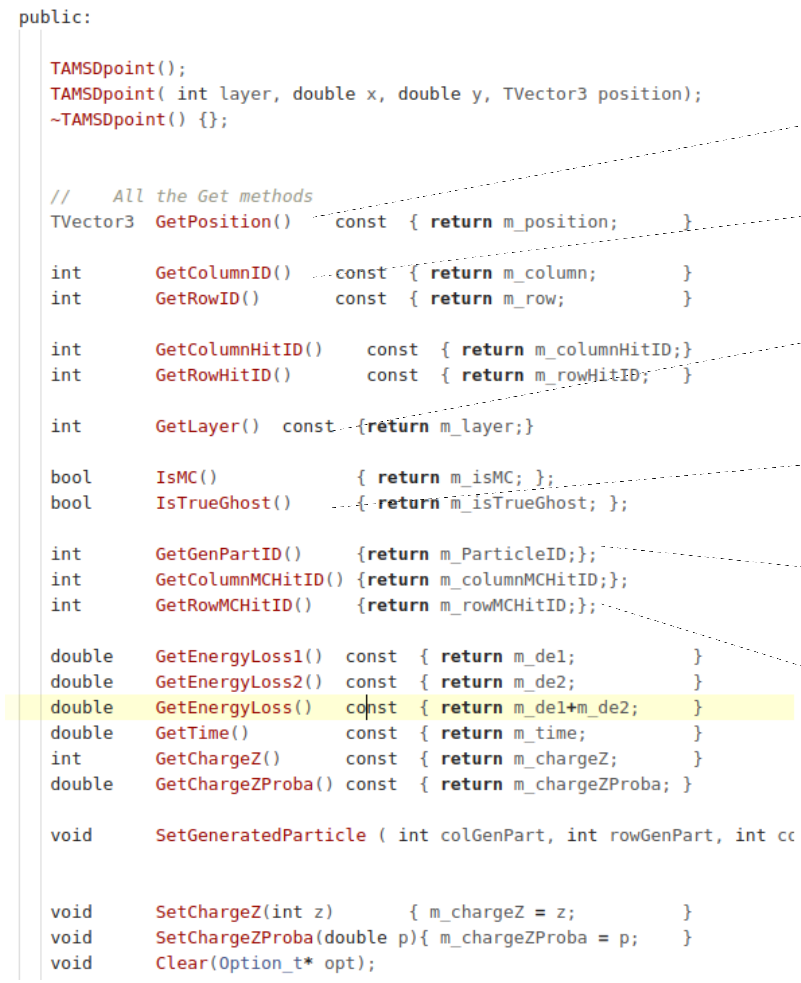


```
class TAMSDpoint : public TAGobject {

private:
   TVector3    m_position;      // position in detector framework
   int         m_layer;         // number of MSD tracking station
   int         m_column;        // column number
   int         m_row;           // row number

   int         m_columnHitID;
   int         m_rowHitID;

   //specific methods for MSD
   int         m_columnParticleID;
   int         m_rowParticleID;
   int         m_ParticleID;

   int         m_columnMCHitID;
   int         m_rowMCHitID;

   bool        m_isMC;
   bool        m_isTrueGhost;

   TVector3    m_posMC;
   TVector3    m_momMC;

   void TrueGhostWarning();

   //generic methods
   Double32_t  m_de1;           // energy loss in 1st strip
   Double32_t  m_de2;           // energy loss in 2nd strip
   Double32_t  m_time;

   int         m_chargeZ;       // raw guess of charge Z
   Double32_t  m_chargeZProba;  // raw guess of charge Z probability
```



```
public:

   TAMSDpoint();
   TAMSDpoint( int layer, double x, double y, TVector3 position);
   ~TAMSDpoint() {};


   //    All the Get methods
   TVector3  GetPosition()    const  { return m_position;       }

   int       GetColumnID()    const  { return m_column;         }
   int       GetRowID()       const  { return m_row;            }

   int       GetColumnHitID()    const  { return m_columnHitID;}
   int       GetRowHitID()       const  { return m_rowHitID;   }

   int       GetLayer()  const  {return m_layer;}

   bool      IsMC()              { return m_isMC; };
   bool      IsTrueGhost()       { return m_isTrueGhost; };

   int       GetGenPartID()      {return m_ParticleID;};
   int       GetColumnMCHitID()  {return m_columnMCHitID;};
   int       GetRowMCHitID()     {return m_rowMCHitID;};

   double    GetEnergyLoss1()  const  { return m_de1;            }
   double    GetEnergyLoss2()  const  { return m_de2;            }
   double    GetEnergyLoss()   const  { return m_de1+m_de2;      }
   double    GetTime()         const  { return m_time;           }
   int       GetChargeZ()      const  { return m_chargeZ;        }
   double    GetChargeZProba() const  { return m_chargeZProba; }

   void      SetGeneratedParticle ( int colGenPart, int rowGenPart, int cc


   void      SetChargeZ(int z)        { m_chargeZ = z;          }
   void      SetChargeZProba(double p){ m_chargeZProba = p;     }
   void      Clear(Option_t* opt);
```

New classes TAMSD(ntu)Point



```
public:

    TAMSDpoint();
    TAMSDpoint( int layer, double x, double y, TVector3 position);
    ~TAMSDpoint() {};


    //    All the Get methods
    TVector3  GetPosition()    const  { return m_position;      }

    int       GetColumnID()    const  { return m_column;        }
    int       GetRowID()       const  { return m_row;           }

    int       GetColumnHitID()    const  { return m_columnHitID;}
    int       GetRowHitID()       const  { return m_rowHitID;   }

    int       GetLayer()  const  {return m_layer;}

    bool      IsMC()              { return m_isMC; };
    bool      IsTrueGhost()       { return m_isTrueGhost; };

    int       GetGenPartID()     {return m_ParticleID;};
    int       GetColumnMCHitID() {return m_columnMCHitID;};
    int       GetRowMCHitID()    {return m_rowMCHitID;};

    double    GetEnergyLoss1()  const  { return m_de1;           }
    double    GetEnergyLoss2()  const  { return m_de2;           }
    double    GetEnergyLoss()   const  { return m_de1+m_de2;     }
    double    GetTime()         const  { return m_time;          }
    int       GetChargeZ()      const  { return m_chargeZ;       }
    double    GetChargeZProba() const  { return m_chargeZProba;  }

    void      SetGeneratedParticle ( int colGenPart, int rowGenPart, int cc


    void      SetChargeZ(int z)       { m_chargeZ = z;           }
    void      SetChargeZProba(double p){ m_chargeZProba = p;     }
    void      Clear(Option_t* opt);
```

position in detector frame (error must be added)

id of hit strip

number of tracking station 0-2

ghost or not?

track index of particle (0 if ghost)

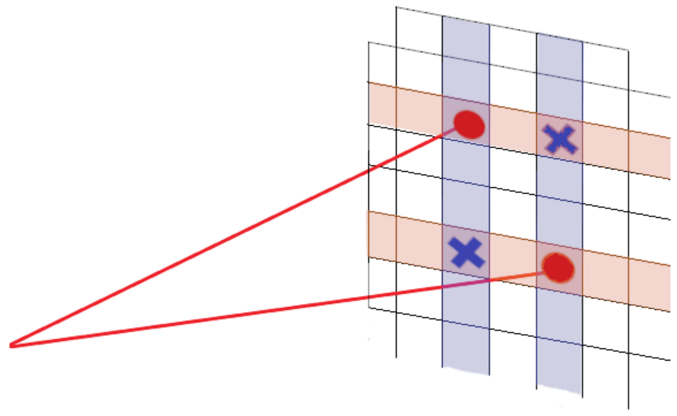
MC index in simulation

Ghost handling

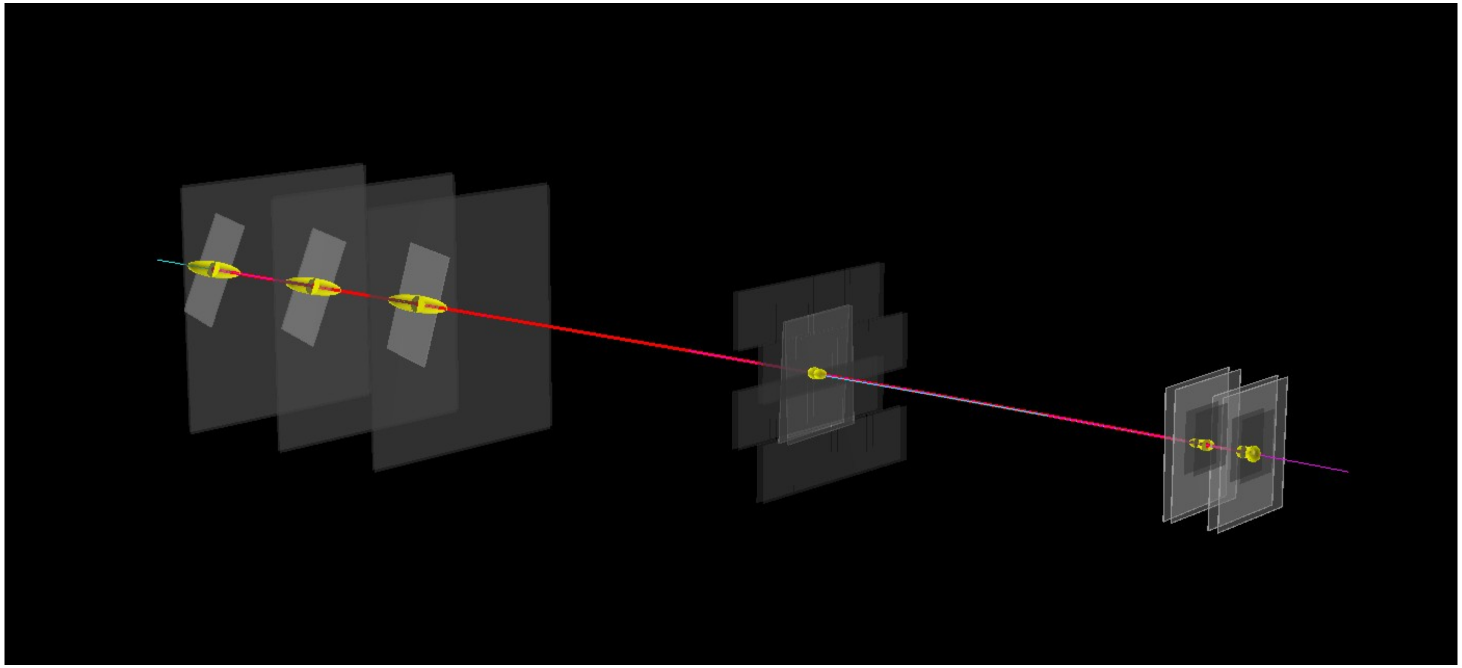
Up to now a point for every pair of hits in two different layers of the same station is created (no check on time/dE is done)

The MC information is carried by the flag isGhost

✓ no assumption on true hits is done (!=TW)
✗ the number of combinations for MSD track finding grows a lot

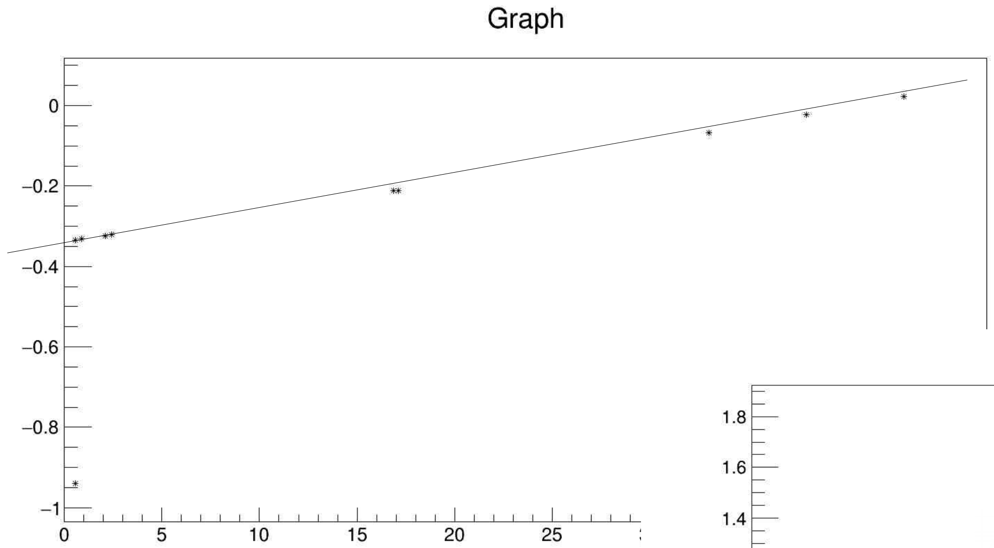


From last meeting

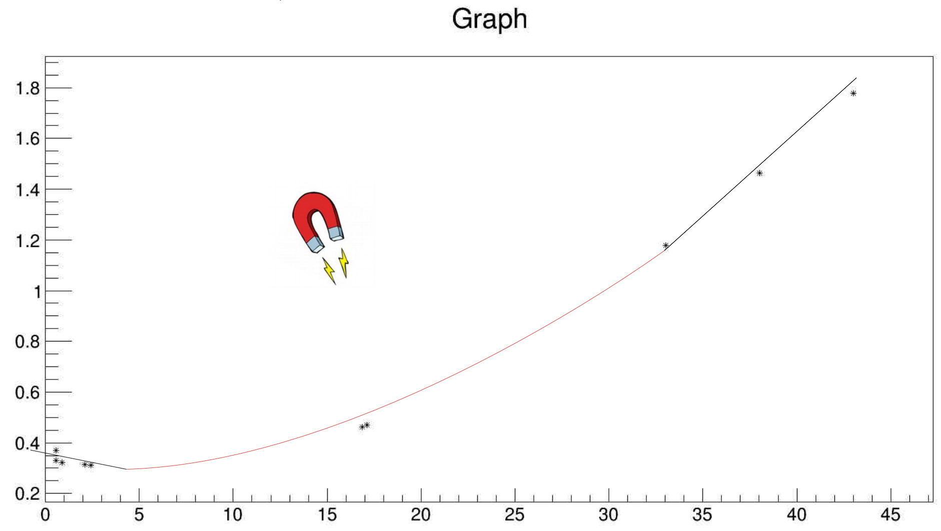


Track finding

XZ and YZ views could be treated independently (bending/no bending)

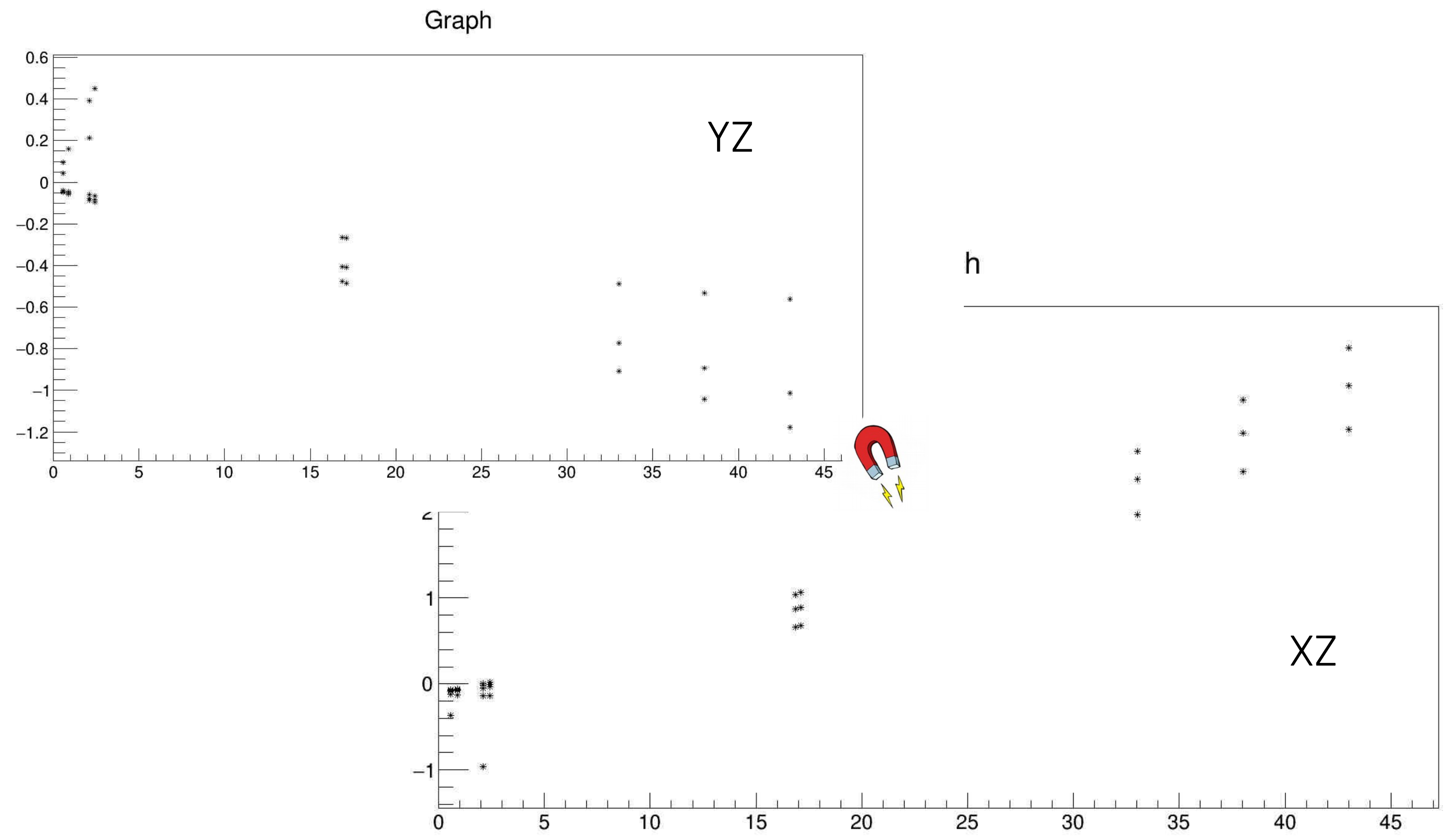


YZ



XZ

Track finding

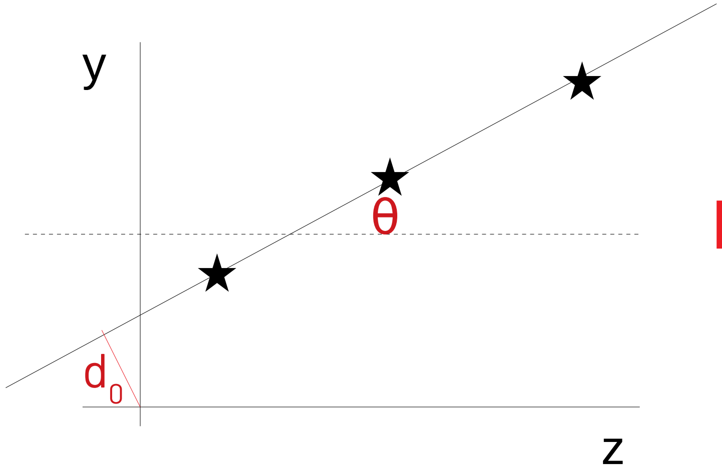


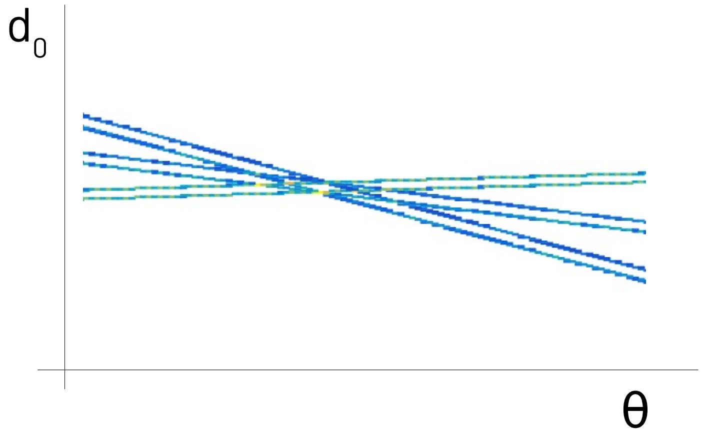
Global method: Hough transform

Test Hough transform on MSD hits in YZ/XZ view

Better to use a kind of "polar representation": from $y = az+b$ to $d_0 = y \cos\theta - z \sin\theta$

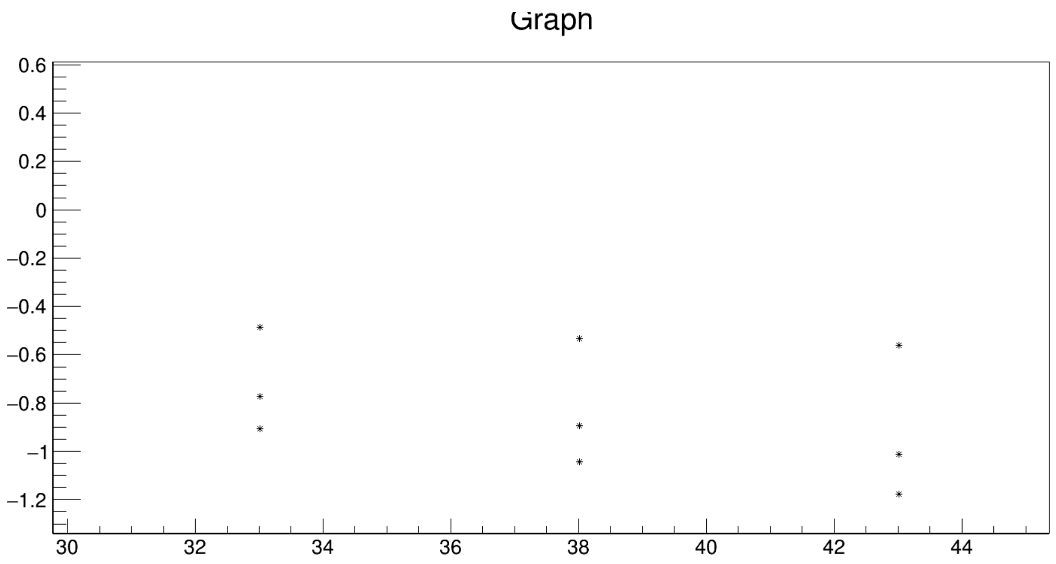
A lot of parameters must be tuned

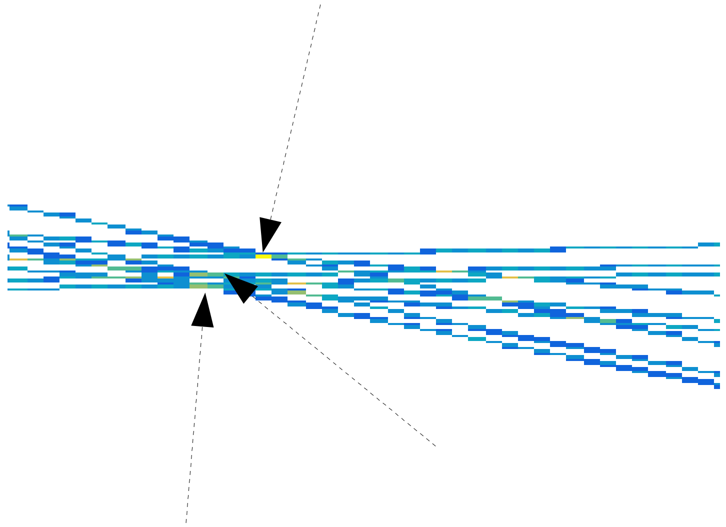




Global method: Hough transform

“Crowded” events could be disentagled but a good “peakfinder” algorithm is needed





Local method

Find every possible combination of 3 points in 3 tracking stations

Search for lines in XZ and YZ views independently and match candidates with all hits in common

Check if track candidates are correct with MC

We deal with a big number of ghosts in crowded events

Once candidate tracks are found we can search for hits in the TW and in the IT

Implementation and automatic selection of candidates is still ongoing