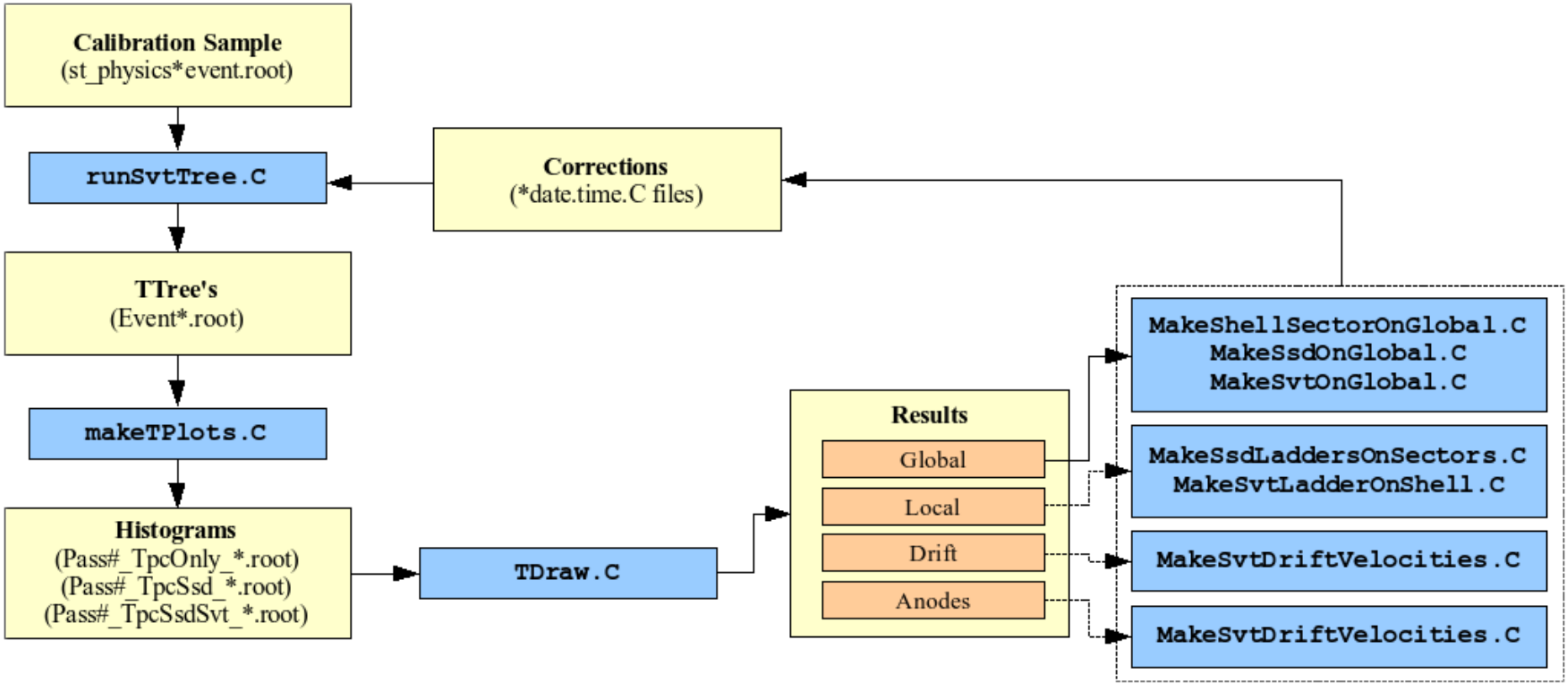
Alignment : implementation status

Outline

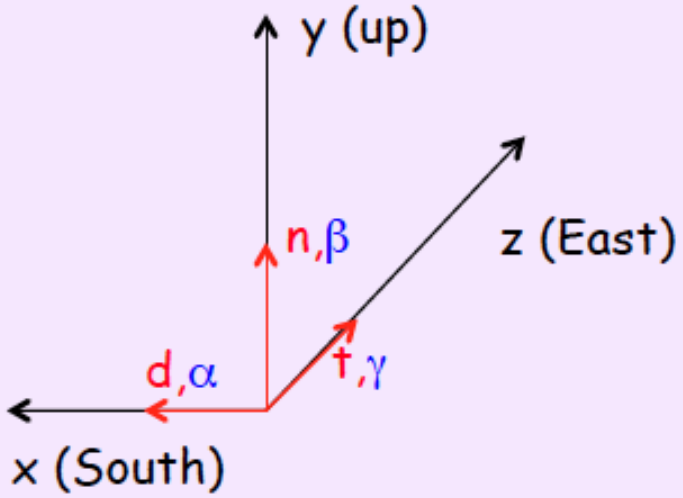
- Geometry tables
- Structures (code, Maker)
- DEV14 setup :
  - Geometry
  - BFC chain
  - Volume hierarchy
- Alignment code : changes for HFT configuration
- Details of implementation :
  - Test with SSD hits shifted
- RESULTS
- Plans forward, other approaches

From SSD-SVT Alignment Procedures



Offline use of Survey Info

Definitions



- Survey info is used in initial Local-to-Global tran
- Done in
- Genera
- This can be sensor or a hit PXL sensors a distortion function (TPS output) will take care of individual pixels in a wafer.
- n,d,t are unit vectors and β,α,γ the corresponding rotation angles, RHS

Measurement is done in local frame

TGeoHMatrix definition

$$\begin{pmatrix} x_G \\ y_G \\ z_G \\ 1 \end{pmatrix} = \begin{bmatrix} \hat{d}_x & \hat{n}_x & \hat{t}_x & d_x \\ \hat{d}_y & \hat{n}_y & \hat{t}_y & d_y \\ \hat{d}_z & \hat{n}_z & \hat{t}_z & d_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} x_L \\ y_L \\ z_L \\ 1 \end{pmatrix}$$

Transform example

$$x_G = \left(\hat{d}_x \cdot x_L + \hat{n}_x \cdot y_L + \hat{t}_x \cdot z_L\right) + d_x$$

Local <-> Global transforms

A series of matrix multiplications

Example: $STAR/StarDb/Geometry/ssd/SsdOnGlobal.upgr01.C

```
WG              = Tpc2Global * GL       * SG          * LS              * WLL;

WaferInGlobal= Tpc2Global *SsdinTpc*SectorInSSD*LadderInSector*WaferInLadder
```

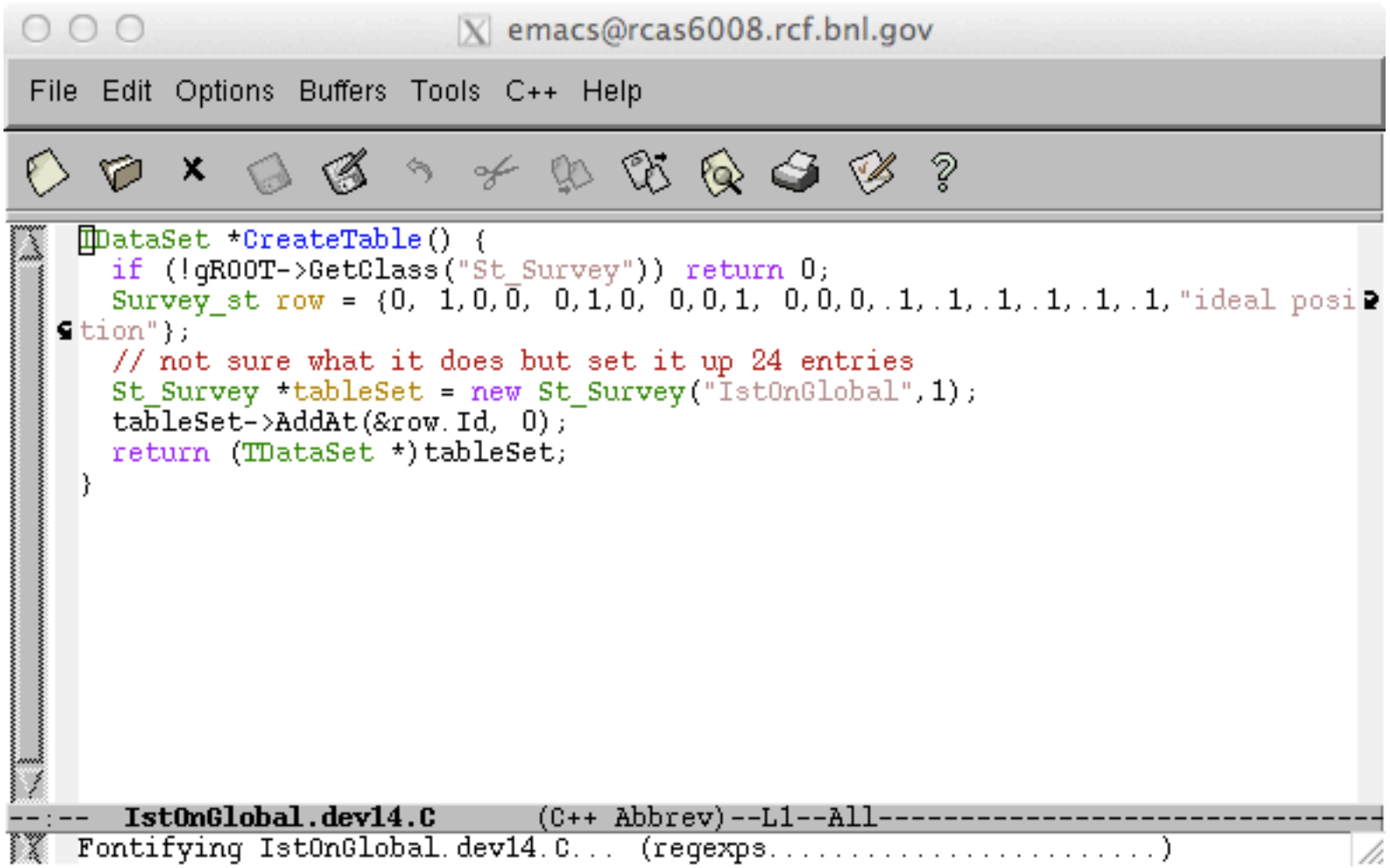
```
// SSD and SVT as whole
St_Survey *SsdOnGlobal = (St_Survey *) GetDa
Survey_st *OnGlobal       = SsdOnGlobal->Ge
  GL.SetRotation(&OnGlobal->r00);
  GL.SetTranslation(&OnGlobal->t0);
// SSD sectors in SSD/SVT system
St_Survey *SsdSectorsOnGlobal = (St_Survey *) Get
// ladders in the SSD sector coordinate systems
St_Survey *SsdLaddersOnSectors = (St_Survey *) GetDataBase
                                                 ry/ssd/SsdLaddersOnSectors");
// wafers in the SSD ladder coordinate systems
St_Survey *SsdWafersOnLadders = (St_Survey *) GetDataBase("Geometry/ssd/SsdWafersOnLadders");
```

Global (STAR) to local (wafer) transformation done using multiplication of rotations matrices

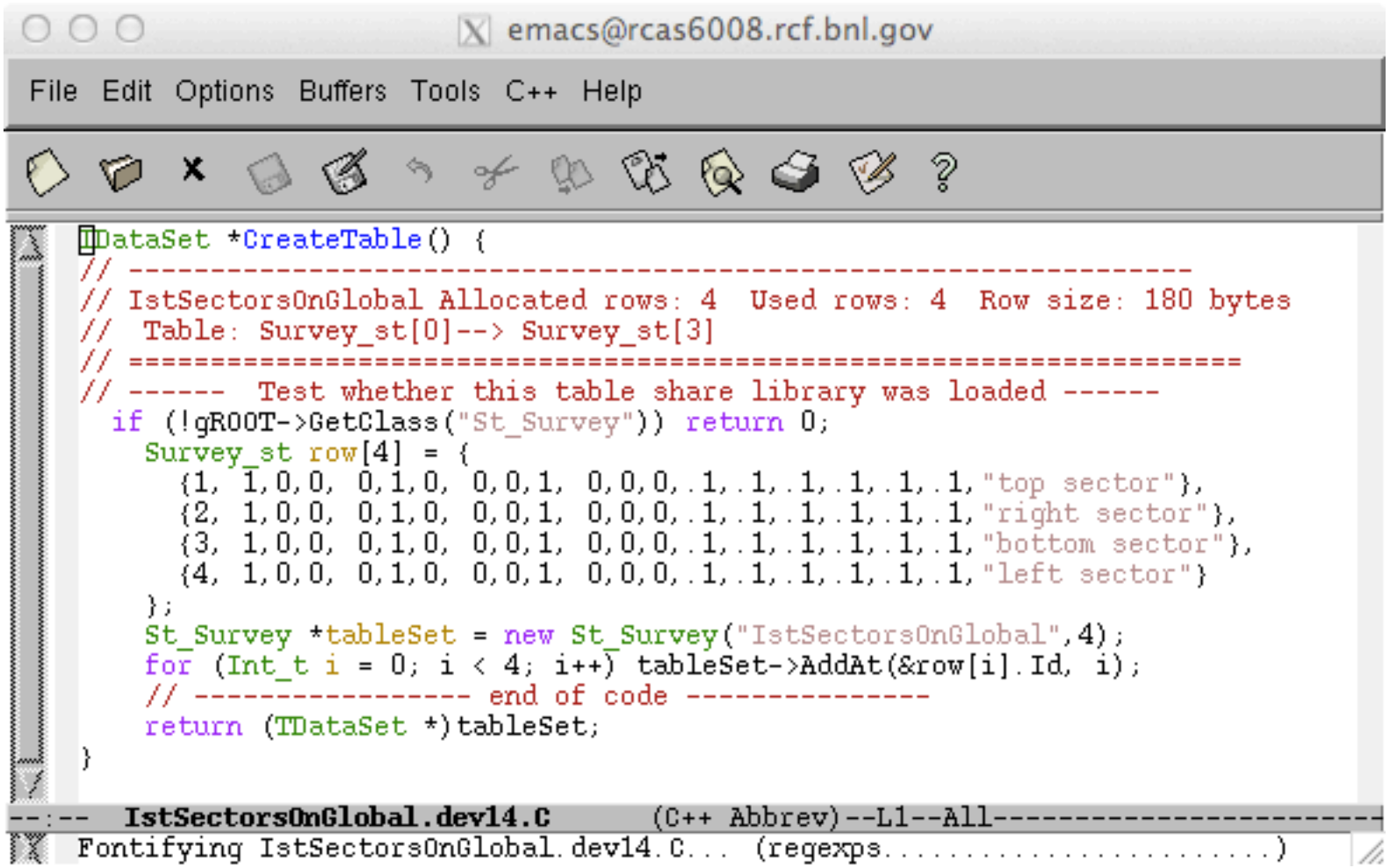
Configuration for HFT(i)

- Templates of tables/codes based on the SSD/SVT
- Geometry tables (from survey)
  - will be placed in StarDb/Geometry/ist (example)
  - Values are retrieved with a dedicated maker : (example)StIstDbMaker *(slide 27 for the role of this maker)* has been created
  - These are internal tables (steps) to position each volume with respect STAR/TPC.
- Final values of alignment are written to STAR database (timestamp), therefore a dedicated table has to be created in StDb
  - (example) istWafersPosition

IstOnGlobal



IstSectorOnGlobal



IstLaddersOnSectors(i)

emacs@rcas6008.rcf.bnl.gov

File Edit Options Buffers Tools C++ Help

```
TDataSet *CreateTable() {
// -----------------------------------------------------------------
//
// Positioning of the IST ladders in the IST sector coordinate systems
  if (!gROOT->GetClass("St_Survey")) return 0;
    //      D[0]      N[0]      T[0]      D[1]      N[1]      T[1]      D[2]      N[2]      T[2]      X[0]      X[1]      X[2]
    //      $1,$4,$7,$2,$5,$8,$3,$6,$9,$10,$11,$12
    //      $1,$2,$5,$8,$3,$6,$9,$4,$7,$10,$11,$12,$13
    //      $1",\t"$2",\t"$5",\t"$8",\t"$3",\t"$6",\t"$9",\t"$4",\t"$7",\t"$10",\t"$11",\t"$12",\t"$13
    //      sed -e 's/,/ /g' istWaferPosition.sim | awk '{print  $1",\t"$2",\t"$5",\t"$8",\t"$3",\t"$6",\t"$9",\t"$4",\t"$7",\t"$10",\t"$11",\t"$12",\t"$13}'
    //      id = 3*layer + 100*wafer + ladder

   // sectors from 1 to 4 , as for thr SSD : test
   Survey_st row[24] = {
{ 101,   0.9659258, 0.2588190, 0.0000000,  -0.2588190, 0.9659258, 0.0000000,   0.0000000,-0.0000000, 1.0000000,    1.83259,  13.91989,   0.00000,    0.10000,
  0.10000,   0.10000,   0.10000,   0.10000,   0.10000},// IBM0_1/IBAM_1
{ 102,   0.8660254, 0.5000000, 0.0000000,  -0.5000000, 0.8660254, 0.0000000,   0.0000000,-0.0000000, 1.0000000,    5.37288,  12.97127,   0.00000,    0.10000,
  0.10000,   0.10000,   0.10000,   0.10000,   0.10000},// IBM0_1/IBAM_2
{ 203,   0.7071068, 0.7071068, 0.0000000,  -0.7071068, 0.7071068, 0.0000000,   0.0000000,-0.0000000, 1.0000000,    8.54701,  11.13868,   0.00000,    0.10000,
  0.10000,   0.10000,   0.10000,   0.10000,   0.10000},// IBM0_1/IBAM_3
{ 204,   0.5000000, 0.8660254, 0.0000000,  -0.8660254, 0.5000000, 0.0000000,   0.0000000,-0.0000000, 1.0000000,   11.13868,   8.54701,   0.00000,    0.10000,
  0.10000,   0.10000,   0.10000,   0.10000,   0.10000},// IBM0_1/IBAM_4
{ 205,   0.2588190, 0.9659258, 0.0000000,  -0.9659258, 0.2588190, 0.0000000,   0.0000000,-0.0000000, 1.0000000,   12.97127,   5.37287,   0.00000,    0.10000,
  0.10000,   0.10000,   0.10000,   0.10000,   0.10000},// IBM0_1/IBAM_5
{ 206,   0.0000000, 1.0000000, 0.0000000,  -1.0000000,-0.0000000, 0.0000000,   0.0000000,-0.0000000, 1.0000000,   13.91989,   1.83259,   0.00000,    0.10000,
  0.10000,   0.10000,   0.10000,   0.10000,   0.10000},// IBM0_1/IBAM_6
{ 207,  -0.2588190, 0.9659258, 0.0000000,  -0.9659258,-0.2588190, 0.0000000,   0.0000000,-0.0000000, 1.0000000,   13.91989,  -1.83259,   0.00000,    0.10000,
  0.10000,   0.10000,   0.10000,   0.10000,   0.10000},// IBM0_1/IBAM_7
{ 208,  -0.5000000, 0.8660254, 0.0000000,  -0.8660254,-0.5000000, 0.0000000,   0.0000000,-0.0000000, 1.0000000,   12.97127,  -5.37287,   0.00000,    0.10000,
  0.10000,   0.10000,   0.10000,   0.10000,   0.10000},// IBM0_1/IBAM_8
{ 209,  -0.7071068, 0.7071068, 0.0000000,  -0.7071068,-0.7071068, 0.0000000,   0.0000000,-0.0000000, 1.0000000,   11.13868,  -8.54701,   0.00000,    0.10000,
  0.10000,   0.10000,   0.10000,   0.10000,   0.10000},// IBM0_1/IBAM_9
{ 210,  -0.8660254, 0.5000000, 0.0000000,  -0.5000000,-0.8660254, 0.0000000,   0.0000000,-0.0000000, 1.0000000,    8.54701, -11.13868,   0.00000,    0.10000,
  0.10000,   0.10000,   0.10000,   0.10000,   0.10000},// IBM0_1/IBAM_10
{ 211,  -0.9659258, 0.2588190, 0.0000000,  -0.2588190,-0.9659258, 0.0000000,   0.0000000,-0.0000000, 1.0000000,    5.37288, -12.97127,   0.00000,    0.10000,
  0.10000,   0.10000,   0.10000,   0.10000,   0.10000},// IBM0_1/IBAM_11
{ 312,  -1.0000000,-0.0000000, 0.0000000,   0.0000000,-1.0000000, 0.0000000,   0.0000000, 0.0000000, 1.0000000,    1.83259, -13.91989,   0.00000,    0.10000,
  0.10000,   0.10000,   0.10000,   0.10000,   0.10000},// IBM0_1/IBAM_12
{ 313,  -0.9659258,-0.2588190, 0.0000000,   0.2588190,-0.9659258, 0.0000000,   0.0000000, 0.0000000, 1.0000000,   -1.83259, -13.91989,   0.00000,    0.10000,
  0.10000,   0.10000,   0.10000,   0.10000,   0.10000},// IBM0_1/IBAM_13
{ 314,  -0.8660254,-0.5000000, 0.0000000,   0.5000000,-0.8660254, 0.0000000,   0.0000000, 0.0000000, 1.0000000,   -5.37288, -12.97127,   0.00000,    0.10000,
  0.10000,   0.10000,   0.10000,   0.10000,   0.10000},// IBM0_1/IBAM_14
{ 315,  -0.7071068,-0.7071068, 0.0000000,   0.7071068,-0.7071068, 0.0000000,   0.0000000, 0.0000000, 1.0000000,   -8.54701, -11.13868,   0.00000,    0.10000,
  0.10000,   0.10000,   0.10000,   0.10000,   0.10000},// IBM0_1/IBAM_15
{ 316,  -0.5000000,-0.8660254, 0.0000000,   0.8660254,-0.5000000, 0.0000000,   0.0000000, 0.0000000, 1.0000000,  -11.13868,  -8.54701,   0.00000,    0.10000,
  0.10000,   0.10000,   0.10000,   0.10000,   0.10000},// IBM0_1/IBAM_16
{ 317,  -0.2588190,-0.9659258, 0.0000000,   0.9659258,-0.2588190, 0.0000000,   0.0000000, 0.0000000, 1.0000000,  -12.97127,  -5.37288,   0.00000,    0.10000,
```

-:-- IstLaddersOnSectors.dev14.C (C++ Abbrev)--L1--Top--

IstLaddersOnSectors(ii)

- This geometry table places azimuthally each ladder as a block
- Each ladder is recognized by an **ID** which has the sector (SSD legacy) information
  - For the IST, I have re-used the same numbering (for now) : 4 sectors
- To create this table, I have used the positions of GEANT volumes from dev14 geometry (macro LoopOverTgeo)

IstWafersOnLadders

emacs@rcas6008.rcf.bnl.gov

File Edit Options Buffers Tools C++ Help

```
TDataSet *CreateTable() {
// -----------------------------------------------------------------
//
  cout <<" CreateTable::IstWafersOnLadders : " << endl;
// Positioning of the IST wafers in the IST ladder coordinate systems
  if (!gROOT->GetClass("St_Survey")) return 0;
  Survey_st row = {0, 1,0,0, 0,1,0, 0,0,1,  0,0,0, .1,.1,.1,.002,.1,.002,"Ideal"};
  Int_t NW = 12;
  Int_t NL = 24;
  //dimension of IST are :
  // 48 cm long , wafer is 4 cm long
  St_Survey *tableSet = new St_Survey("IstWafersOnLadders",NW*NL);
  for (Int_t wafer = 1; wafer <= NW; wafer++) {
    for (Int_t ladder = 1; ladder <= NL; ladder++) {
      cout <<" wafer : " << wafer <<" ladder : " << ladder << " id : " << 3000 + 100*wafer + ladder <<" zpos : " <<  -22 + 4*(wafer-1) << endl;
      row.Id = 3000 + 100*wafer + ladder;
      row.t2 = -22 + 4*(wafer-1);
      tableSet->AddAt(&row.Id);
    }
  }
  return (TDataSet *)tableSet;
}
```

--:-- IstWafersOnLadders.dev14.C (C++ Abbrev)--L1--All----------

Fontifying IstWafersOnLadders.dev14.C... (regexps.....)

- This geometry table places the wafers individually along the Z direction. It will be frozen by survey.

istWafersPosition.idl

emacs@rcas6008.rcf.bnl.gov

File Edit Options Buffers Tools IDL Help

```
/*    istWafersPosition.idl
*
*  Table: istWafersPosition
*
*  description: IST Wafers position
*  absolute orientations (must become relative to
*  parent ladder)
*/


struct istWafersPosition {

  long    id;                     /* Identifier: 3000+wafer(from 1 to 24)*100+ladder (from 1 to 12)*/
  long    id_shape;               /* Wafer shape id index   */
  long    ladder;                 /* Ladder number wafer is on  */
  long    layer;                  /* Layer number wafer is on.    */
  long    num_chip;               /* Chip number within ladder  */
  double  driftDirection[3];      /* Drift direction  */
  double  normalDirection[3];     /* Coordinates of unit vector normal to wafer */
  double  transverseDirection[3]; /* t=nxd, transverse to drift and normal direction    */
  double  centerPosition[3];      /* coordinates of the wafer center in the Star syst  */
```

--:-- **istWafersPosition.idl** (IDL)--L13--Top----------

Wrote /star/u/bouchet/dev14/StDb/idl/istWafersPosition.idl

- This table holds the final positions for each detector volumes (of a given subsystem)
- Stored in STAR DB with appropriate timestamp.

Configuration for HFT(ii)

- Hits collection for HFT, StRnD hit, (pixel and IST) :
  - do not have the local coordinates of hits
  - do not have sub hit collection (per ladder, per wafer)
- Numbering scheme / detectors layout are different than SVT/SSD → impact on the alignment code which was written explicitly for SVT/SSD *(slide 28)*

Setup for the alignment review

- Use dev14 geometry *(slide 29 ,30)*
- STARDEV
- Subsystems used : SSD and IST
  - PIXEL has a different numbering scheme
  - SSD and IST share the same structure (1 barrel) therefore the geometry tables should have the same structure
- SSD and IST are placed in the CAVE.
  - Easier to reproduce the geometry tables for the IST based on the SSD (the SSD was always in the case since y2007x → upgr15 geometries)

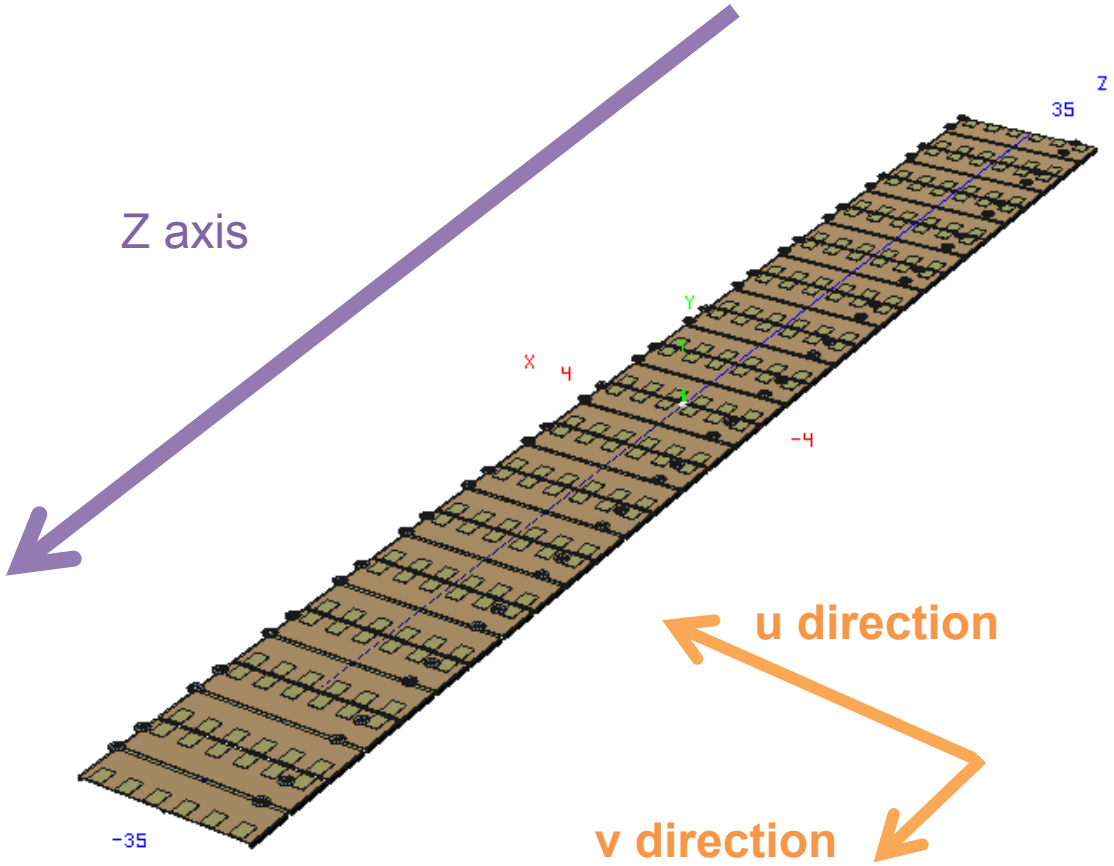
Alignment code details

- First step : histogramming
  - Loop over tracks (i)
  - Loop over list of rotation matrices (j) :
    - Get prediction (uP,vP) of track i on volume j
    - If (uP,vP) $\in$ local dimension of volume j :
      - Loop over hits of track i
      - If there are hits in volume j → Fill Tree with hits and track information.
- Second step : straight line fit
  - For each volume, residuals (hit position deviation from track) vs. derivative matrix component.
- Primary tracks are used (Nfit>15)

TEST

- Simulation consists of :
  - Pions per event (low stat : 50 pions/event ; high stat : 150 pions/event), we tried AuAu200 central hijing but the track density is too high
  - 800 and 400 events respectively for low(high) statistics
  - Dev14 geometry
  - No hits smearing in SSD and IST fast simulator
  - Shift “by hand” the SSD hits (semi blind) from a same sector.
  - Run alignment code and check the misalignment is found

How we "*shifted*" hits ?



- The shift along the Z axis is done in local coordinate
- Global (X,Y,Z) →local (u,v) :
  - u : transverse to Z axis (beam)
  - v : along Z axis

Results (fixed)

FIXED – LOW STATS

dX mkm	dY mkm	dZ mkm	alpha mrad	beta mrad	gamma mrad	Comment
47.37+-13.13	-2.35+- 1.71	-54.34+-135.76	-0.65+- 0.37	-0.53+- 0.18	4.80+- 2.33	Average for SSD Sector 1
20.30+-10.22	-15.94+- 9.64	2.87+-31.68	-0.07+- 0.03	0.04+- 0.04	-0.03+- 0.20	Average for SSD Sector 2
7.89+- 8.25	-0.21+- 1.65	-51.85+-98.45	0.05+- 0.27	-0.62+- 0.15	0.05+- 0.04	Average for SSD Sector 3
39.43+-11.80	15.98+- 7.41	31.78+-27.96	-0.15+- 0.03	-0.13+- 0.04	-0.28+- 0.21	Average for SSD Sector 4
12.46+- 4.21	**20.10+- 4.37**	-21.72+-15.56	-0.02+- 0.02	0.04+- 0.02	-0.04+- 0.02	Average for All Ssd

FIXED – HIGHER STATS

dX mkm	dY mkm	dZ mkm	alpha mrad	beta mrad	gamma mrad	Comment
-6.83+- 2.88	0.11+- 0.27	39.30+-44.76	0.02+- 0.00	0.21+- 0.01	-1.59+- 0.13	Average for SSD Sector 1
-0.77+- 0.42	-30.48+- 2.41	-15.87+- 5.73	-0.00+- 0.01	-0.00+- 0.00	0.82+- 0.05	Average for SSD Sector 2
-0.74+-55.97	0.28+- 0.28	-6.15+- 2.83	0.01+- 0.01	-0.00+- 0.01	**3.26+- 0.26**	Average for SSD Sector 3
0.24+- 0.45	**-66.33+- 1.63**	**-51.49+- 7.30**	-0.05+- 0.00	-0.00+- 0.00	-0.10+- 0.05	Average for SSD Sector 4
-19.39+- 1.16	**-0.32+- 1.58**	**-25.83+- 3.93**	-0.01+- 0.00	-0.01+- 0.00	-0.07+- 0.01	Average for All Ssd

- Statistics matter (up to a point)
- Some mis-behavior detected (next slides)

Results (shift test)

```
gGeoManager->GetCurrentMatrix()->MasterToLocal(xg,xl);
LOG_DEBUG <<Form("local position x=%f y=%f z=%f",xl[0],xl[1],xl[2])<<endm;
gGeoManager->GetCurrentMatrix()->Print();

Double_t xlSmear[3]={0,0,0};
xlSmear[1] = xl[1];
if(currLadder>=12 && currLadder<=18)
  {
    xlSmear[2] = (ShiftHit(xl[2], .125));
    xlSmear[0] = xl[0];
  }
```

Input code

FIXED – HIGHER STATS

dX mkm	dY mkm	dZ mkm	alpha mrad	beta mrad	gamma mrad	Comment
-6.83+- 2.88	0.11+- 0.27	39.30+-44.76	0.02+- 0.00	0.21+- 0.01	-1.59+- 0.13	Average for SSD Sector 1
-0.77+- 0.42	-30.48+- 2.41	-15.87+- 5.73	-0.00+- 0.01	-0.00+- 0.00	0.82+- 0.05	Average for SSD Sector 2
-0.74+-55.97	0.28+- 0.28	-6.15+- 2.83	0.01+- 0.01	-0.00+- 0.01	**3.26+- 0.26**	Average for SSD Sector 3
0.24+- 0.45	**-66.33+- 1.63**	**-51.49+- 7.30**	-0.05+- 0.00	-0.00+- 0.00	-0.10+- 0.05	Average for SSD Sector 4
-19.39+- 1.16	-0.32+- 1.58	-25.83+- 3.93	-0.01+- 0.00	-0.01+- 0.00	-0.07+- 0.01	Average for All Ssd

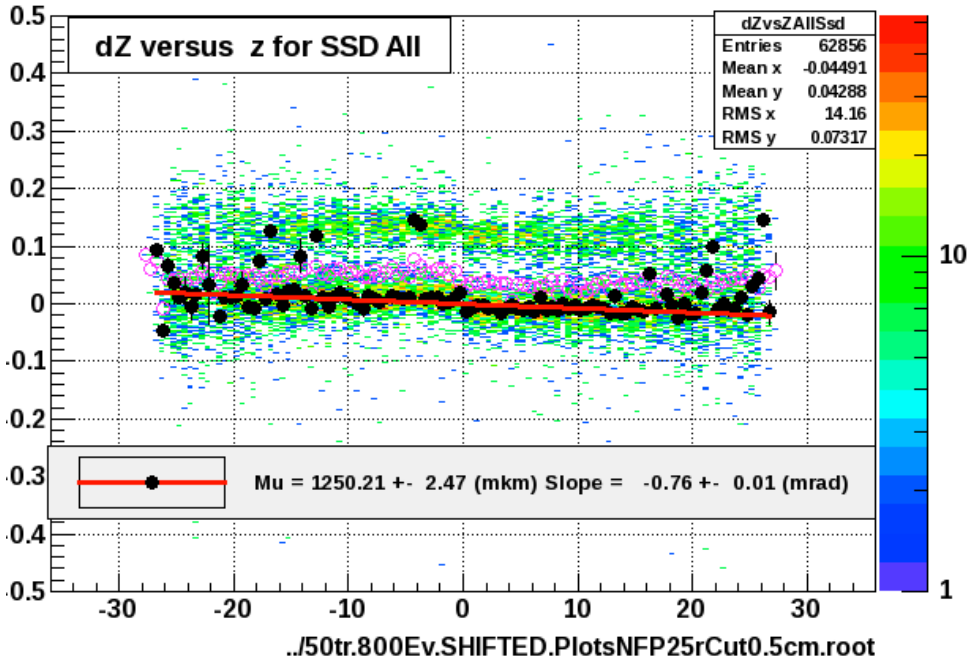
Sector-4 SHIFTED by 1.25 mm (+1250 mkm)

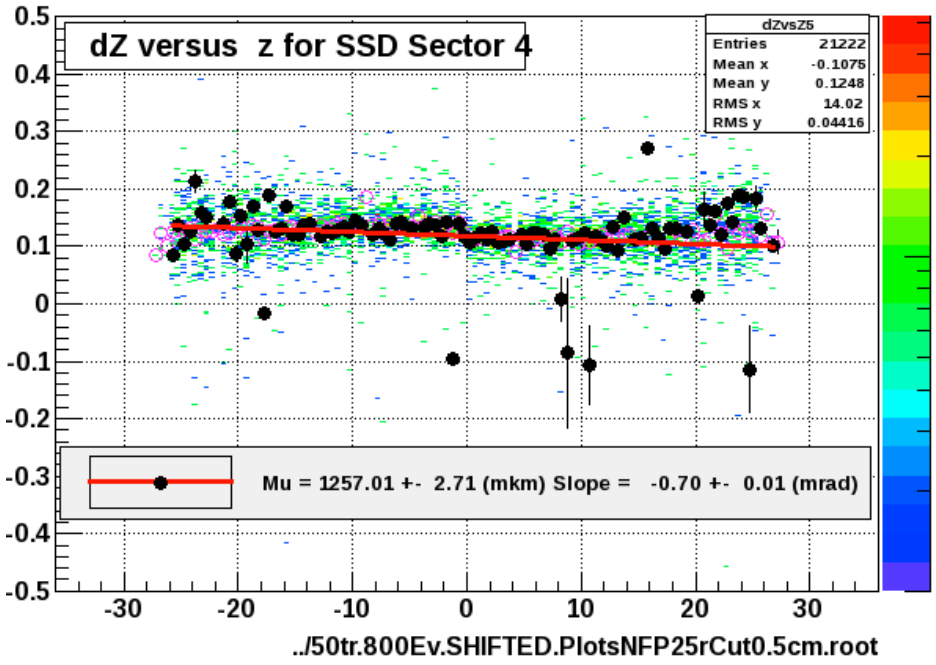
dX mkm	dY mkm	dZ mkm	alpha mrad	beta mrad	gamma mrad	Comment
70.65+-14.72	-0.21+- 0.23	-36.55+- 5.13	0.00+- 0.00	0.13+- 0.01	2.01+- 0.04	Average for SSD Sector 1
-8.03+- 2.75	-6.10+- 2.29	-12.56+- 3.96	0.06+- 0.00	-0.03+- 0.00	0.84+- 0.04	Average for SSD Sector 2
-8.74+-10.68	0.58+- 0.24	-39.07+-11.77	0.01+- 0.00	0.05+- 0.01	**1.78+- 0.10**	Average for SSD Sector 3
10.02+- 2.68	27.94+- 2.37	**-1257.02+- 1.51**	0.01+- 0.00	0.00+- 0.00	-1.13+- 0.04	Average for SSD Sector 4
-8.02+- 1.37	9.18+- 1.31	**-434.10+-25.80**	-0.01+- 0.00	-1.95+- 0.00	-0.38+- 0.01	Average for All Ssd

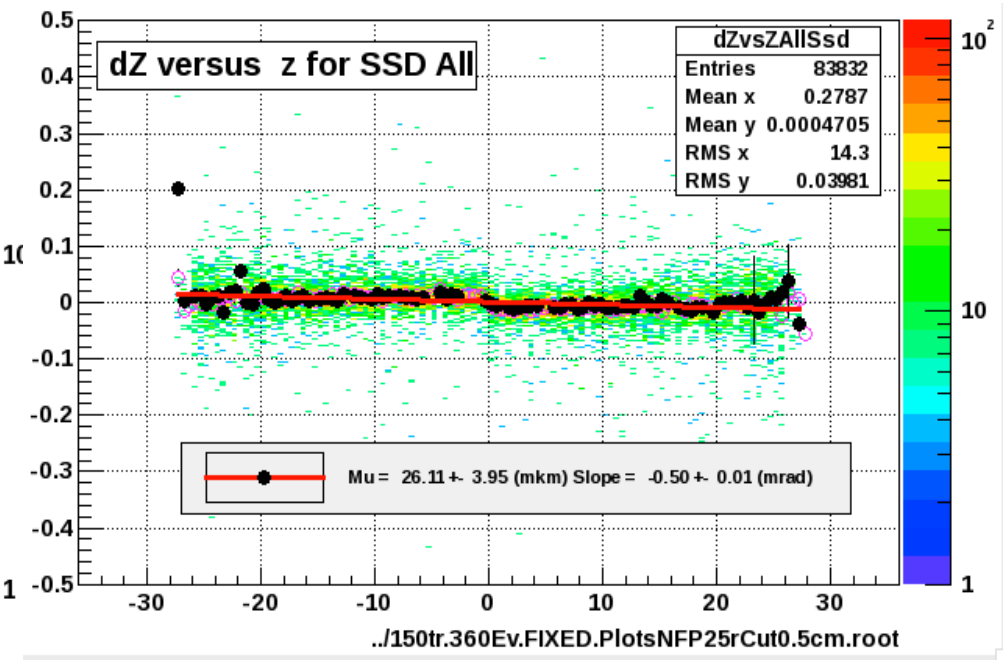
- Shift detected within a few microns
- Some mis-behavior still exists. Needs Investigation (next slides)
- One Ladder can distort the Global positioning of a detector as a whole
- Iterations and/or Reversal of procedure is needed

Results - Global

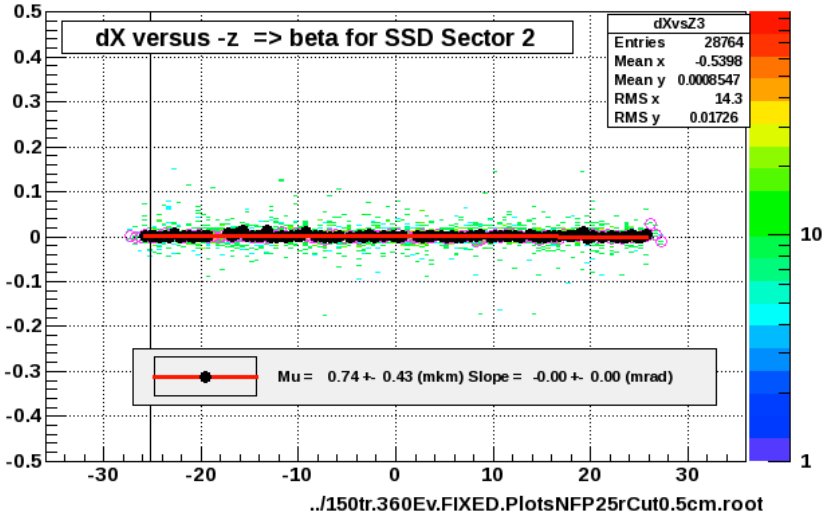
- Fixed and Shifted dZ
- One Sector or Ladder affects dZ ALL
- A kink exists in the middle (next slides)

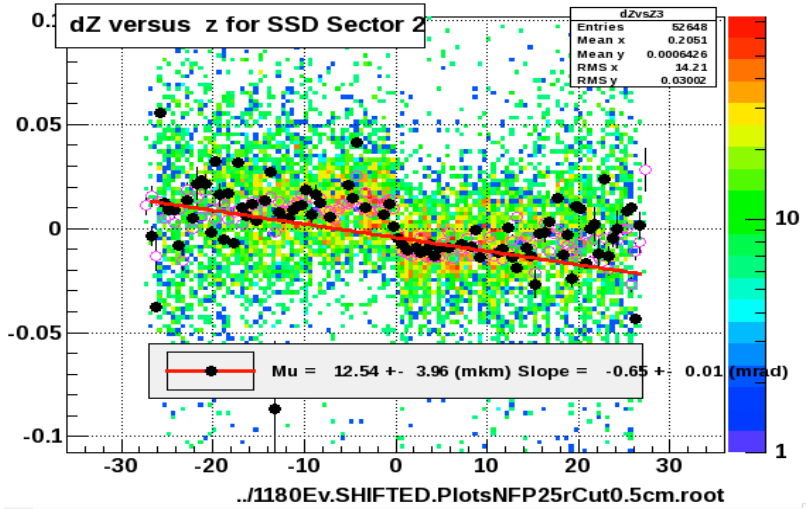


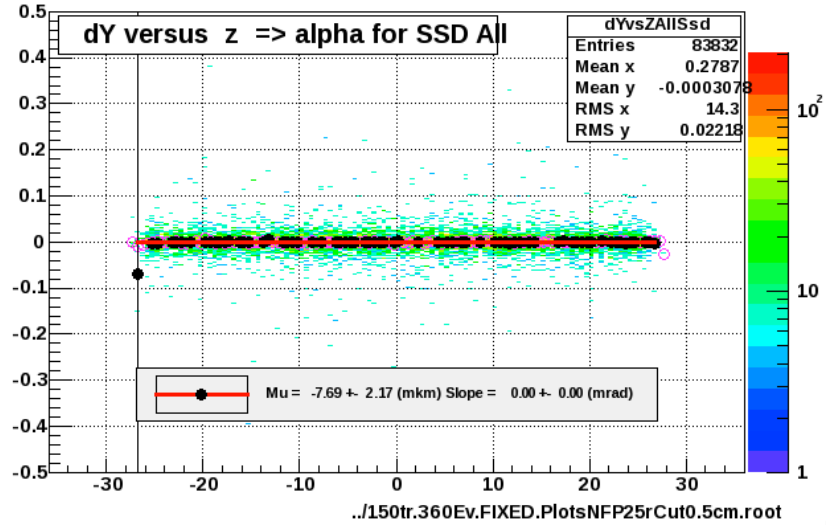


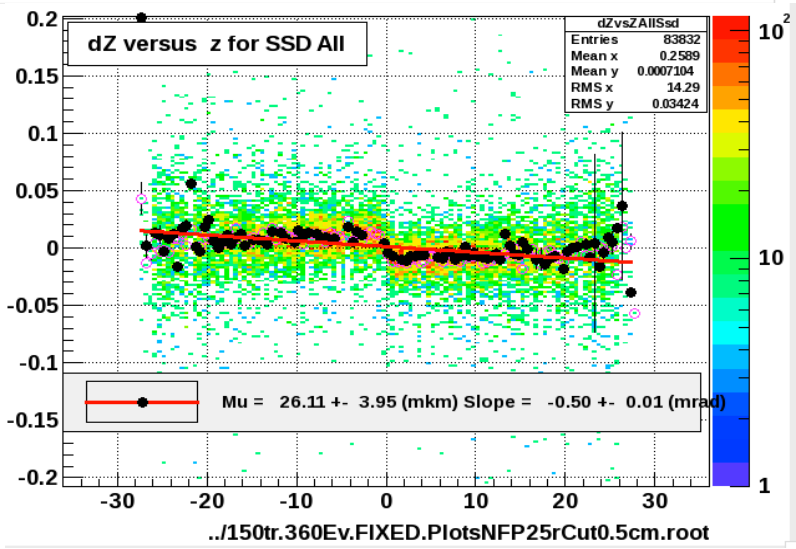


Results - Global





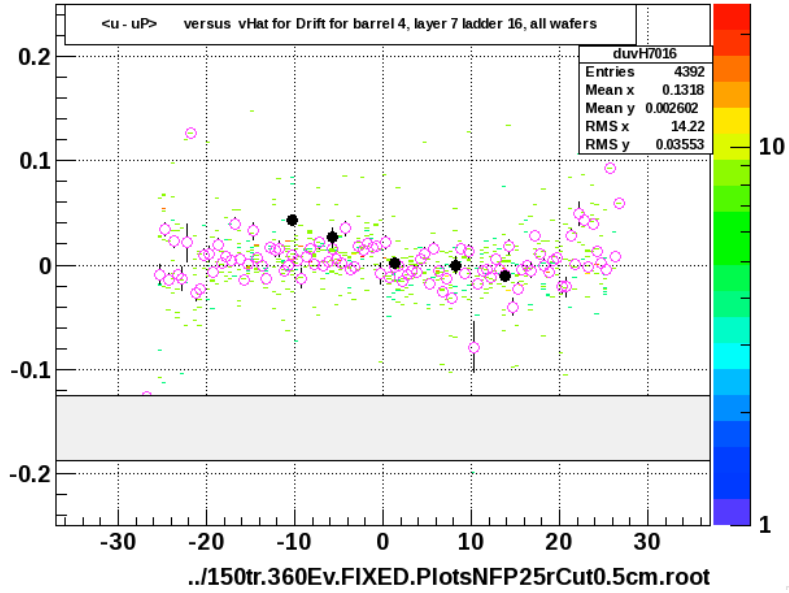


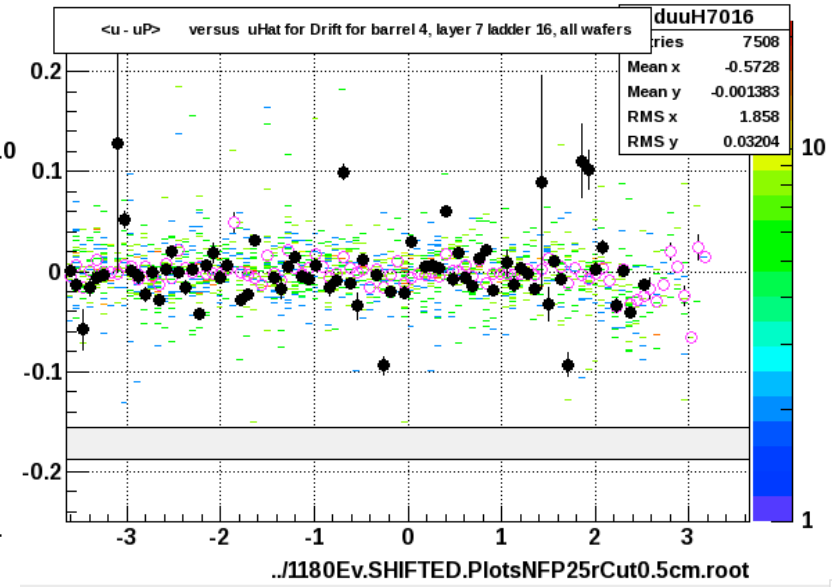


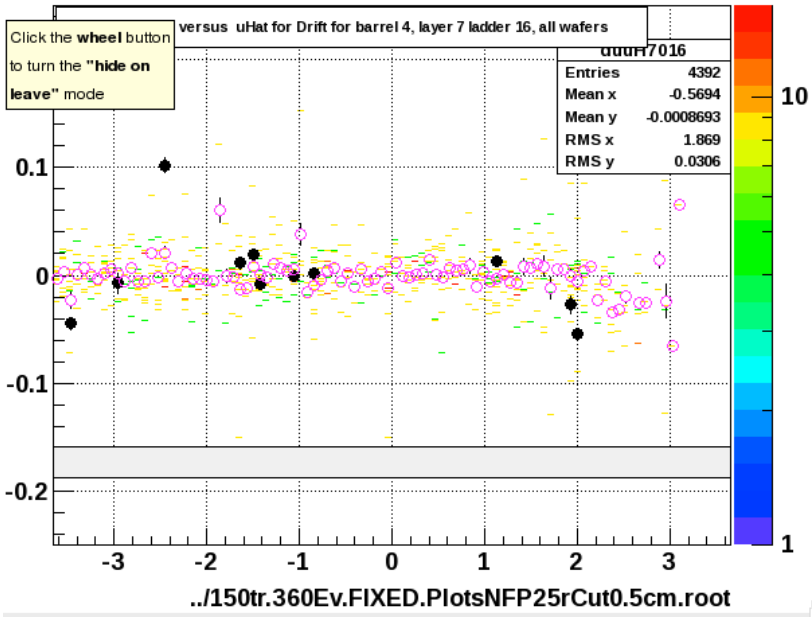
- dX/dY are fine...dZ shows TPC t0 effect (!)
- This is also responsible for the γ angle effect

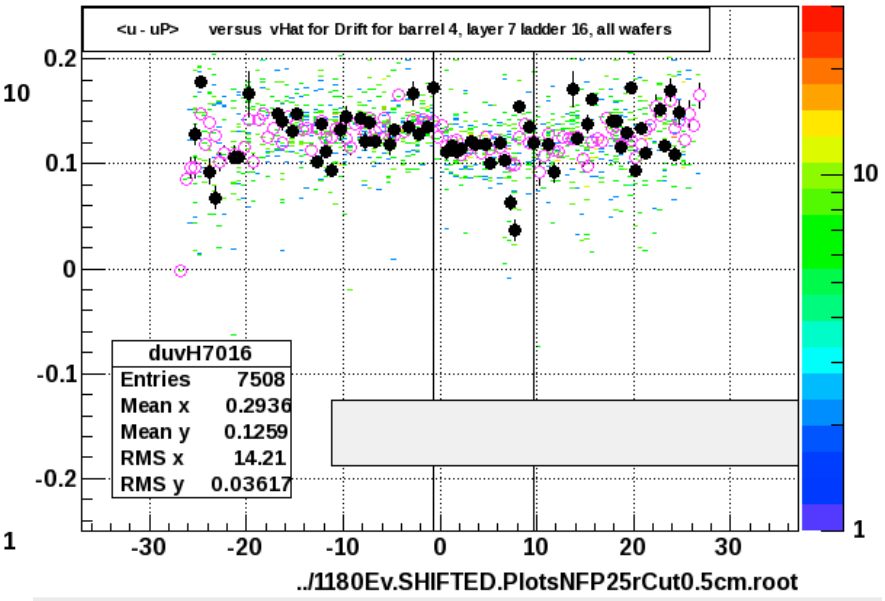
Results – Ladder/local

- FIXED left - SHIFTED right
- u (r/ϕ) uP - v (beam) right

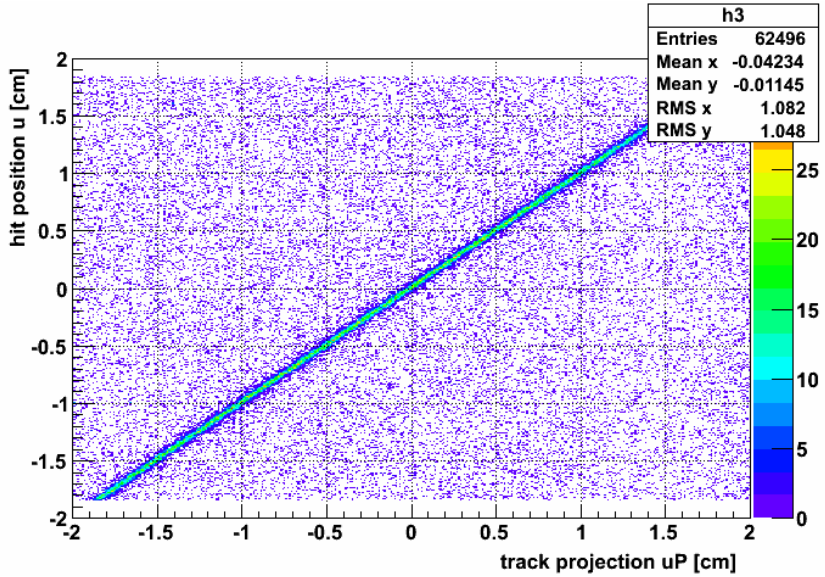


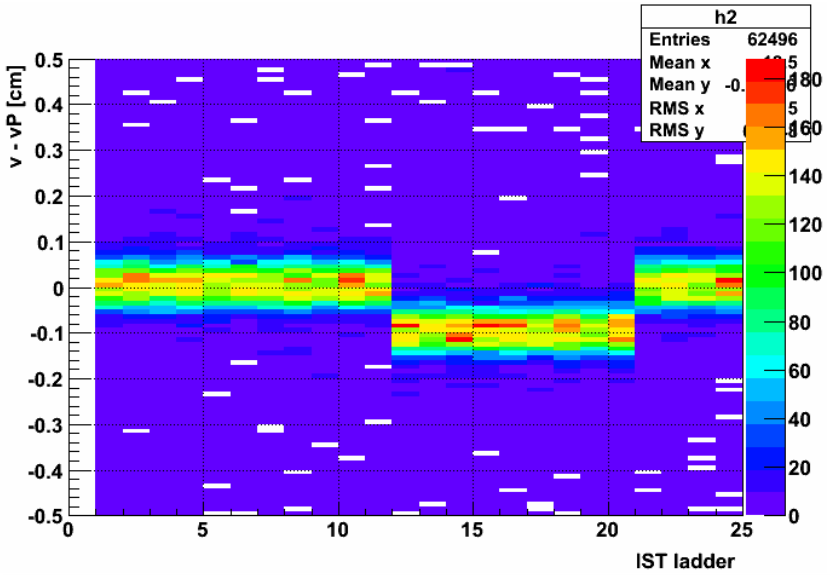


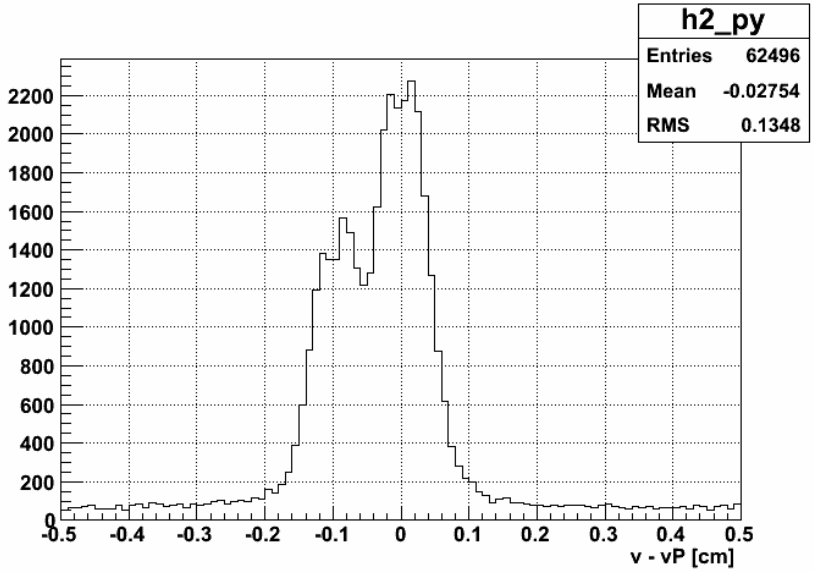




Test with a shift in IST hits (z axis)







- An IST sector has been shifted.
- top left : correlation **u** vs. **uP**
- Top right : **v** - **vP** (z axis) vs Ladder Id :
  - ladders belonging to the shifted sector are clearly visible
- Bottom right : distribution **v** – **vP** showing the 2 components : non shifted ladders and shifted ladders

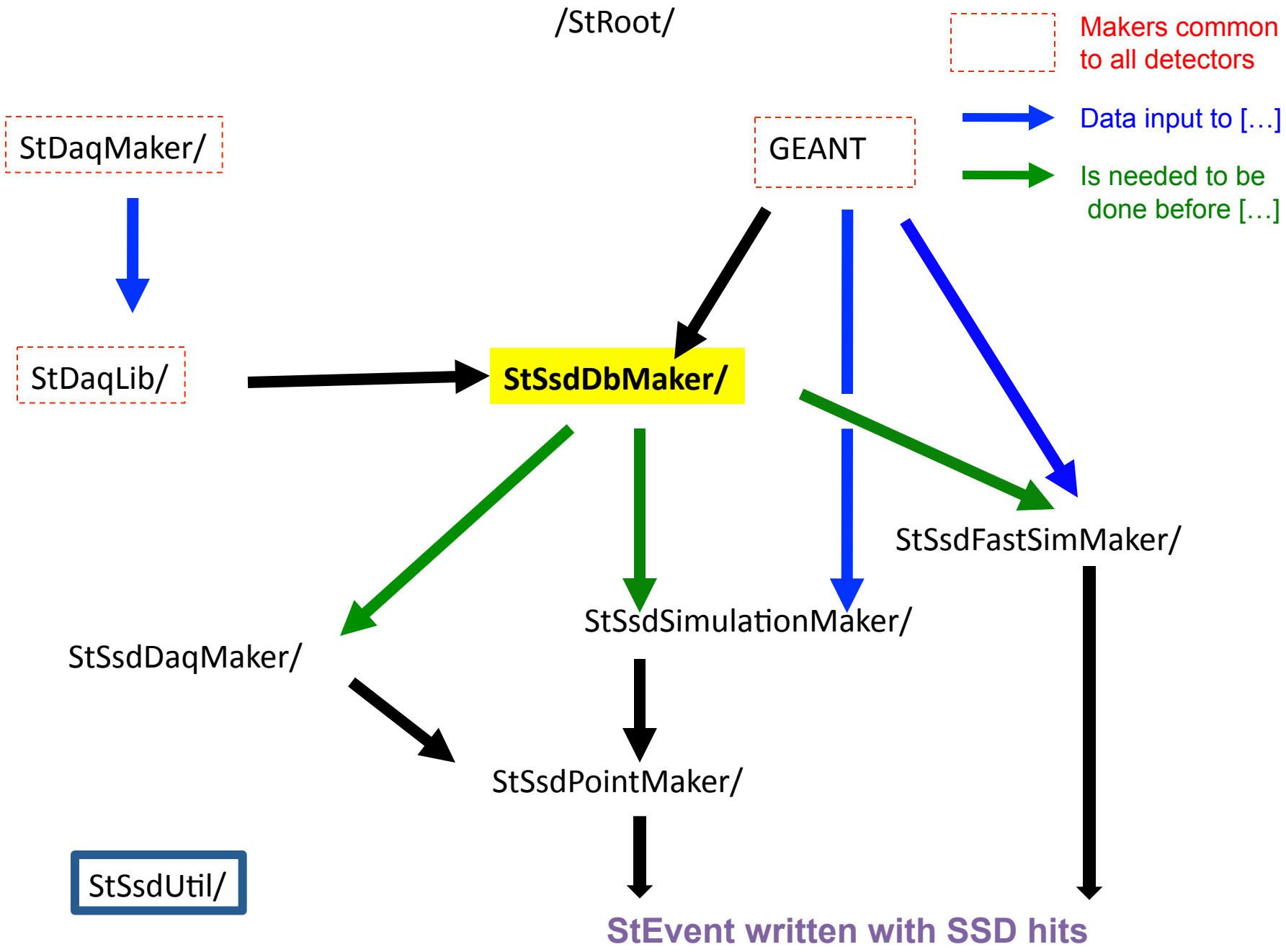
Summary

- Revival of alignment code (~5 years old) for HFT era :
  - Successively works for (new) SSD
  - In progress for the IST (notation has to be changed)
  - Exercising BFC chain option for HFT hits reconstruction/tracking

Work Forward

- Modify the alignment code for the IST and PXL subsystems (numbering scheme)
- Run the FULL chain of alignment (including corrections of successive passes)
- Revisit offline structures :
  - StRnD hit not adapted
- Hierarchy :
  - Is the current scheme : Cave→IDSM→(PXL/IST/SSD) adapted *(slide 31,32)*
- Other approaches to look :
  - Millipede (Blobel,Kleinwort: hep-ex/02080)

END


/StRoot/
Makers common to all detectors
Data input to […]
Is needed to be done before […]
StDaqMaker/
GEANT
StDaqLib/
StSsdDbMaker/
StSsdFastSimMaker/
StSsdSimulationMaker/
StSsdDaqMaker/
StSsdPointMaker/
StSsdUtil/
StEvent written with SSD hits

Numbering/notation for (SSD-SVT)

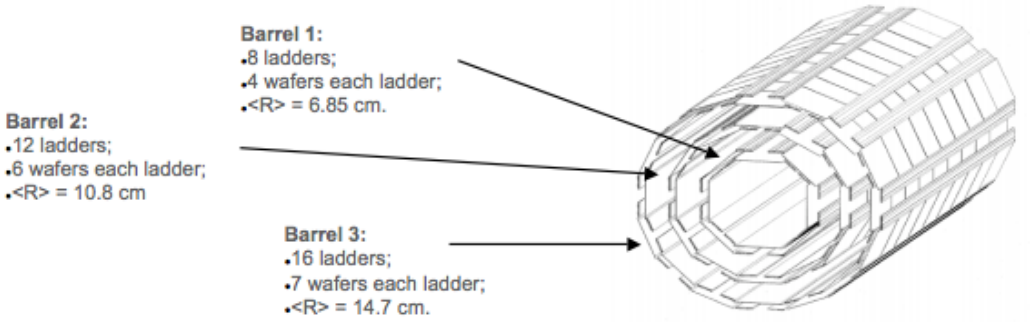


Figure 2. Silicon Drift Detector

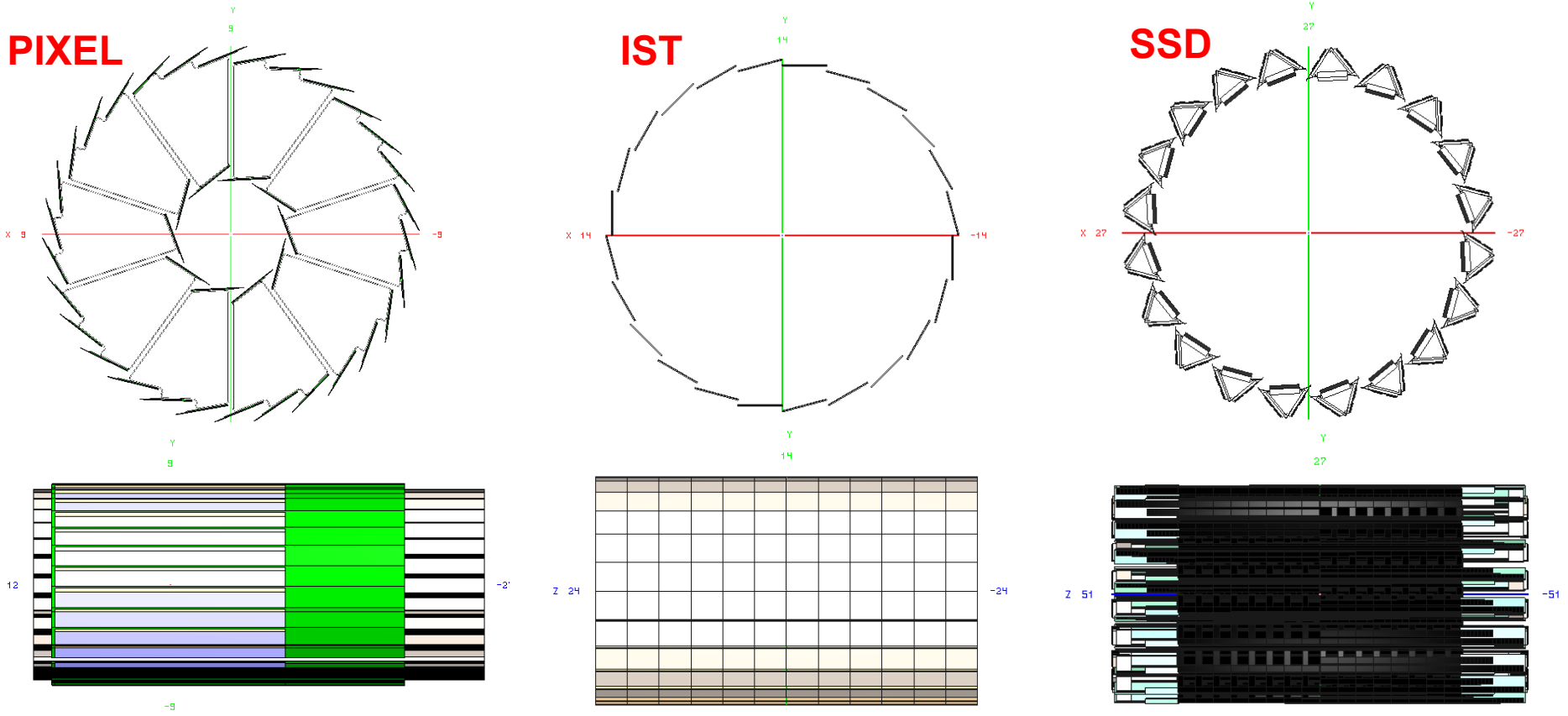
- SVT/SSD wasn't using the same definition for grouping active volumes/

	BARREL	LADDERS	WAFERS per LADDER
SVT	1	8	4
SVT	2	12	6
SVT	3	16	7
SSD	4	20	16

What the alignment code used :
- SVT : 2 clamshells, 6 layers
- SSD : 1 barrel, 1 layer

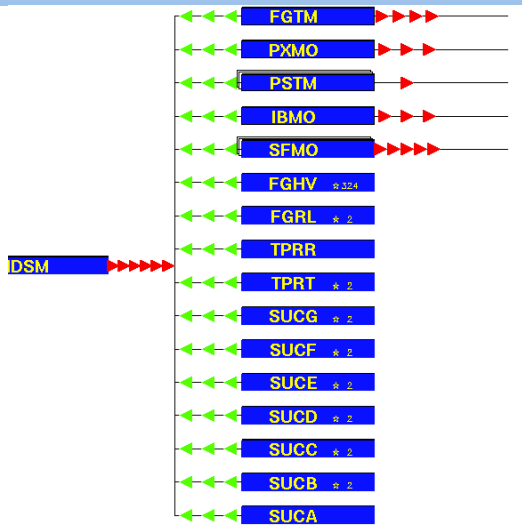
■♏♏♎ to change all notations for the HFT era.

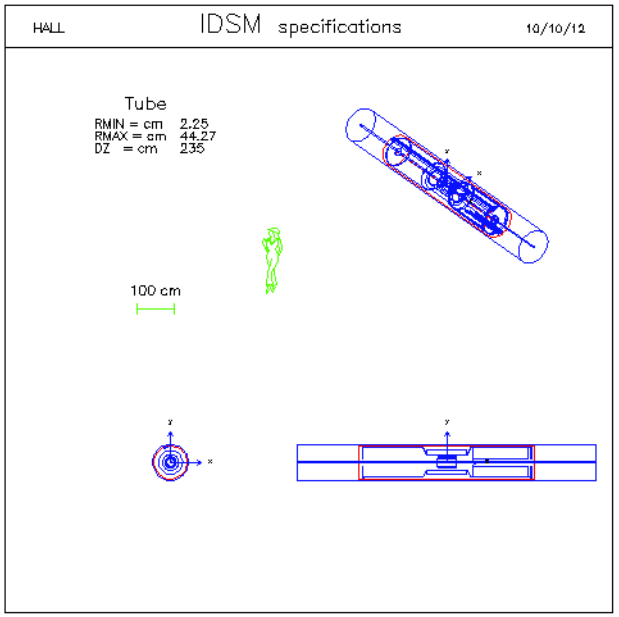
Dev14 : subsystems



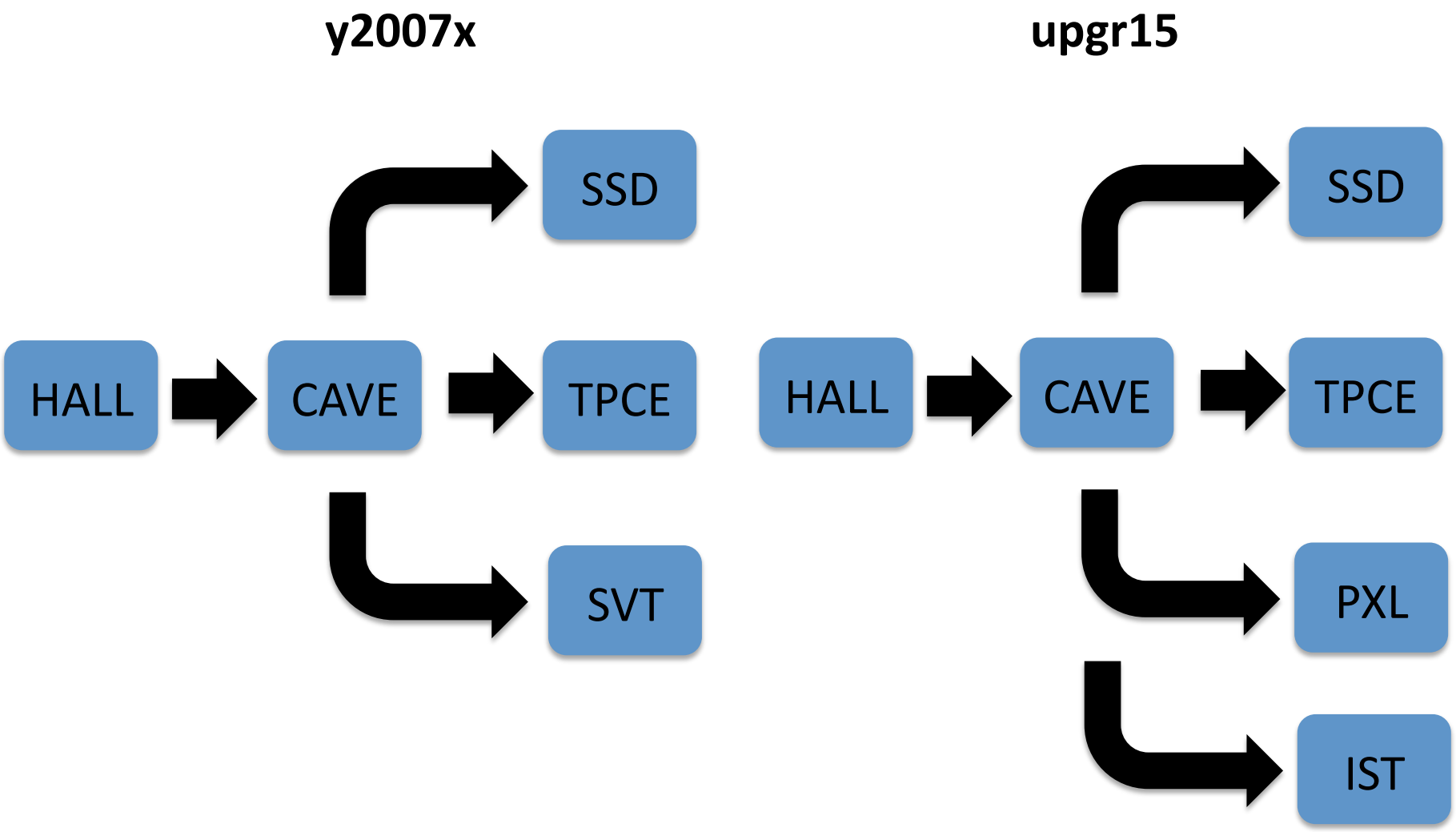
detector	Radius [cm]	technology	Hit resolution : R/ϕ - Z [µm]	Thickness [% X_0]
SSD	22	Double sided silicon strips	30 - 850	1
IST	14	Silicon strips pad sensors	170 - 1700	1.2
PIXEL	2.5 ; 8	Active pixels CMOS	8.6 - 8.6	0.37

Dev14 : geometry layout

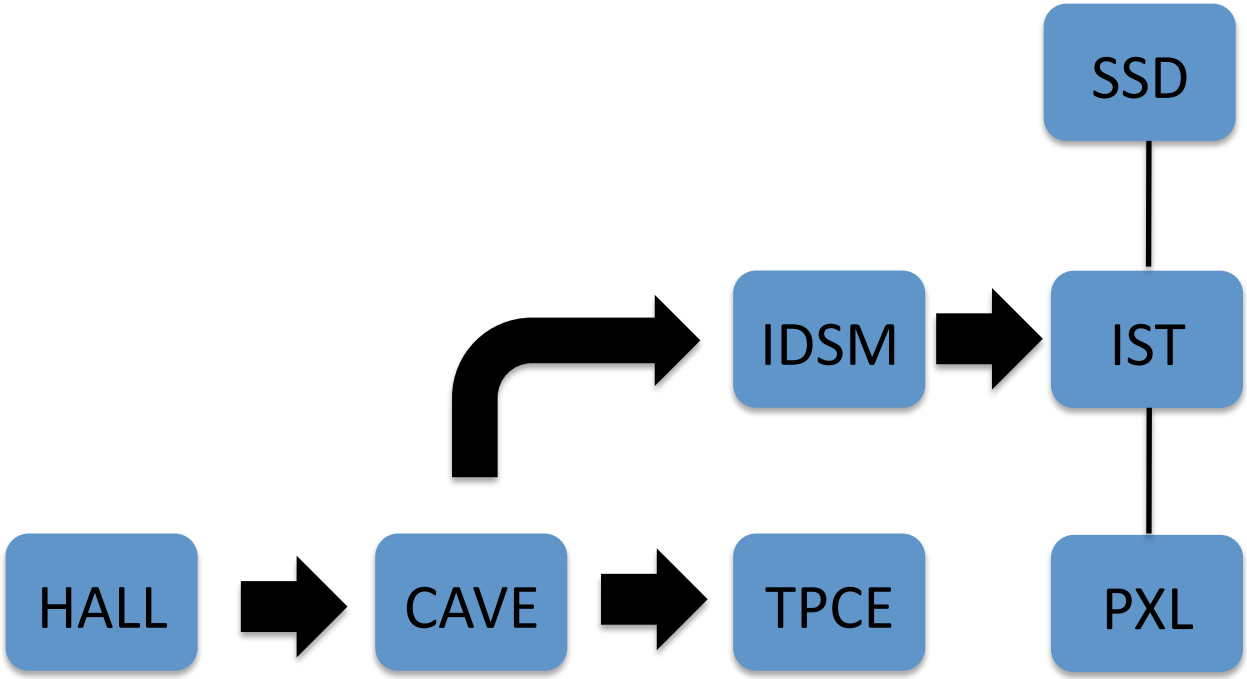




Past geometries



Dev14 geometry



procedures

Creation of IstWaferPositions by using root geometry

Creation of the individual geometry tables(LaddersOnSectors, WafersOnLadder, etc …) in StIstDbMaker using IstWaferPositions

StIstDbMaker has in memory these tables, that can be retrieved/used later

StPixelFastSimMaker : GlobalToLocal for Hits smearing

SvtMatchedTree : for alignment code (StRndHit needs to be updated with the local hit positions)

Slight differences

- There are still slight differences btw GEANT path and Geometry tables (Tz component)
- The reason is because there is a Tpc2Global table that locates the TPC with respect the zero of STAR

```
StPixelFastSimMaker:DEBUG - ist hit collection found
StPixelFastSimMaker:DEBUG - there are 25 ist hits
StPixelFastSimMaker:DEBUG - mc ist hit location x: 2.40095; y: 13.7267; z: -15.0875
StPixelFastSimMaker:DEBUG -  path: /HALL_1/CAVE_1/IBMO_1/IBAM_1/IBLM_3/IBSS_1
matrix global_5 - tr=1  rot=1  refl=0  scl=0
  0.965926    0.258819    0.000000    Tx =   1.822235
 -0.258819    0.965926    0.000000    Ty =  13.881249
  0.000000    0.000000    1.000000    Tz = -14.140000
StPixelFastSimMaker:DEBUG - ist hit local x: 0.599 y: 0.00049994 z: -0.9475
 Idcur : 3301
matrix R3301 - tr=1  rot=1  refl=0  scl=0
  0.965926    0.258819   -0.000000    Tx =   1.832590
 -0.258819    0.965926    0.000000    Ty =  13.919890
  0.000000    0.000000    1.000000    Tz = -14.000000
```

GEANT path

Geometry tables