

sPHENIX実験INTT検出器のための Event Displayの開発

2023/03/03 高エネルギー物理学研究室卒業研究発表会

奈良女子大学 B4 藤原愛実

目次

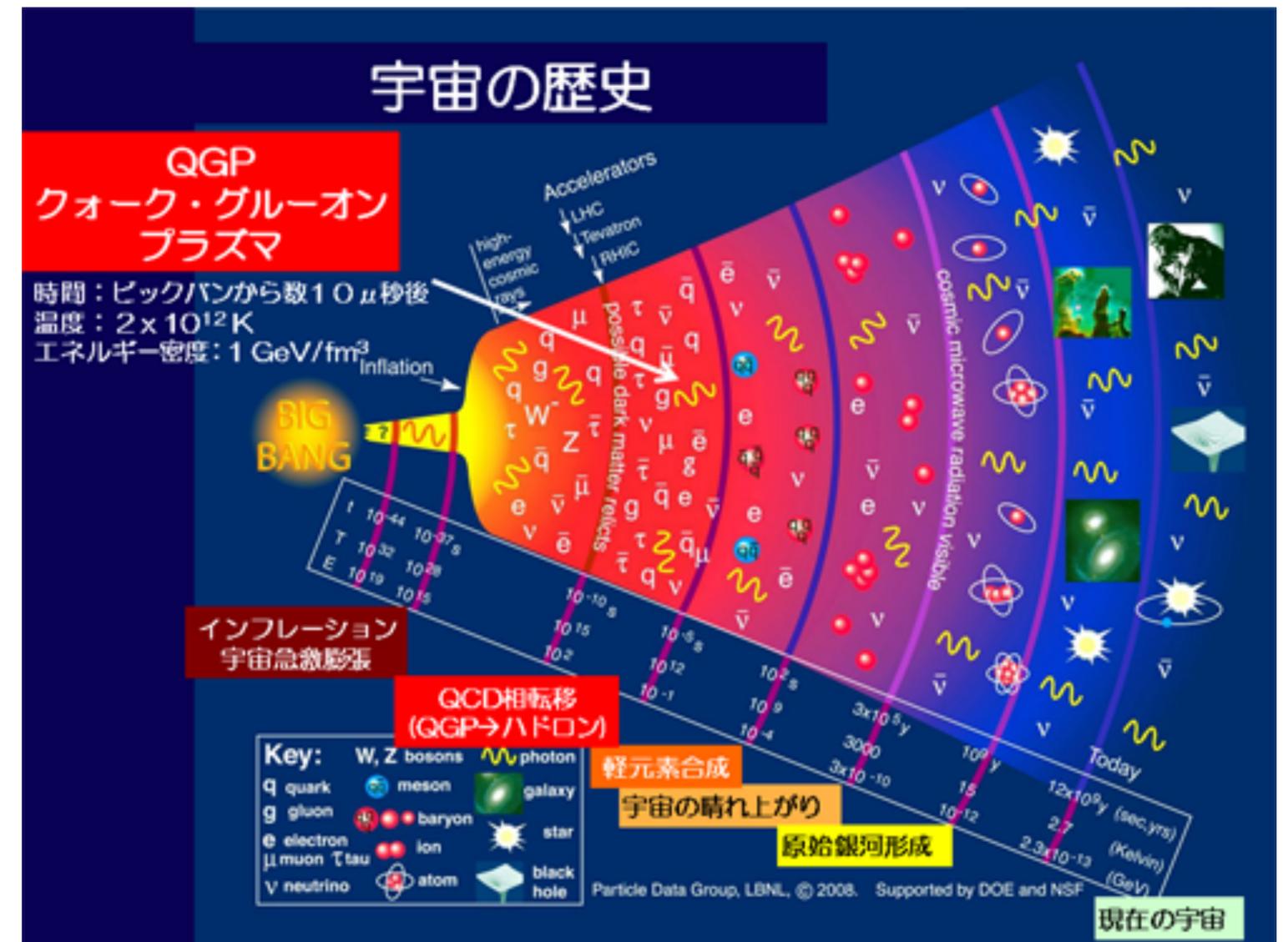
1. 研究背景
2. 研究目的
3. INTT Event Display の開発
4. まとめと今後の課題

1. 研究背景

QGP

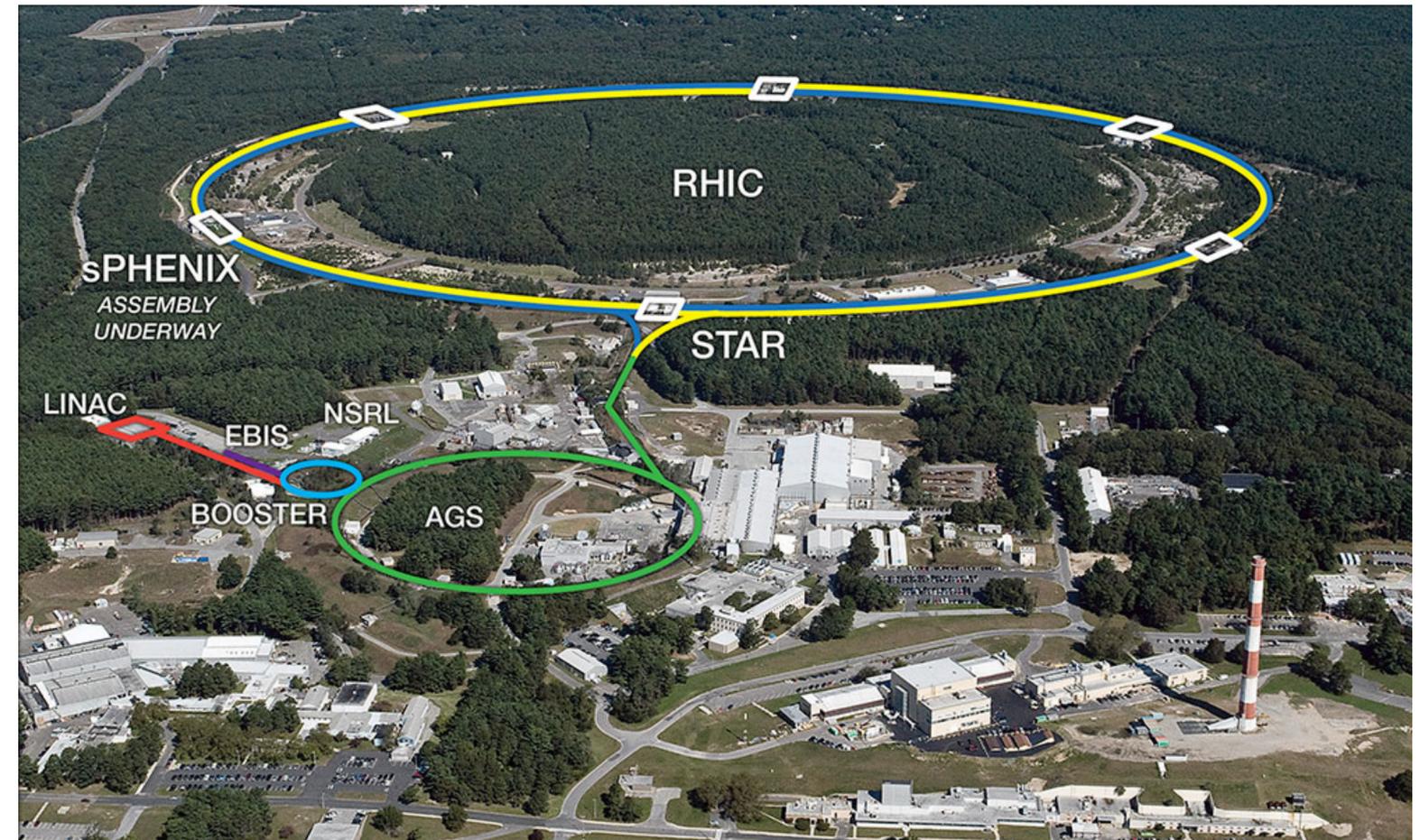
- クォークグルーオンプラズマ(QGP)は、通常では核子内に閉じ込められているクォークやグルーオンが超高温または高密度になることでプラズマ化した状態
- QGP は宇宙誕生後数十 μ 秒後に実現していた

QGP の性質を調べることで宇宙初期の状態を調べることができる



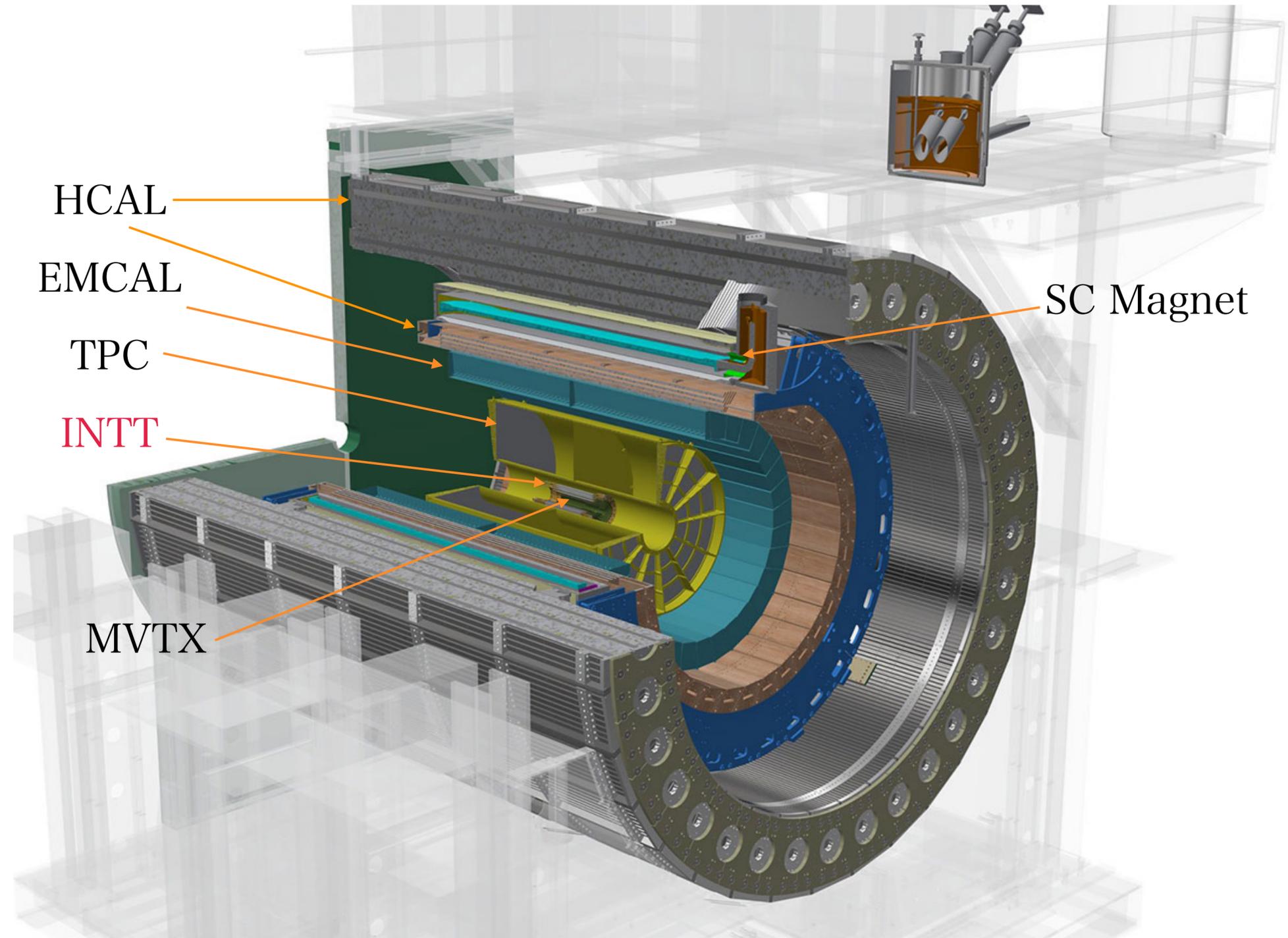
sPHENIX実験

- 米国ブルックヘブン国立研究所(BNL)のRHIC加速器を用いた実験
- 2000-2016年に行われたPHENIX 実験を高度化した実験
- 金原子核対衝突(200GeV)でQGPを実現する
- 2023年から、実験開始予定



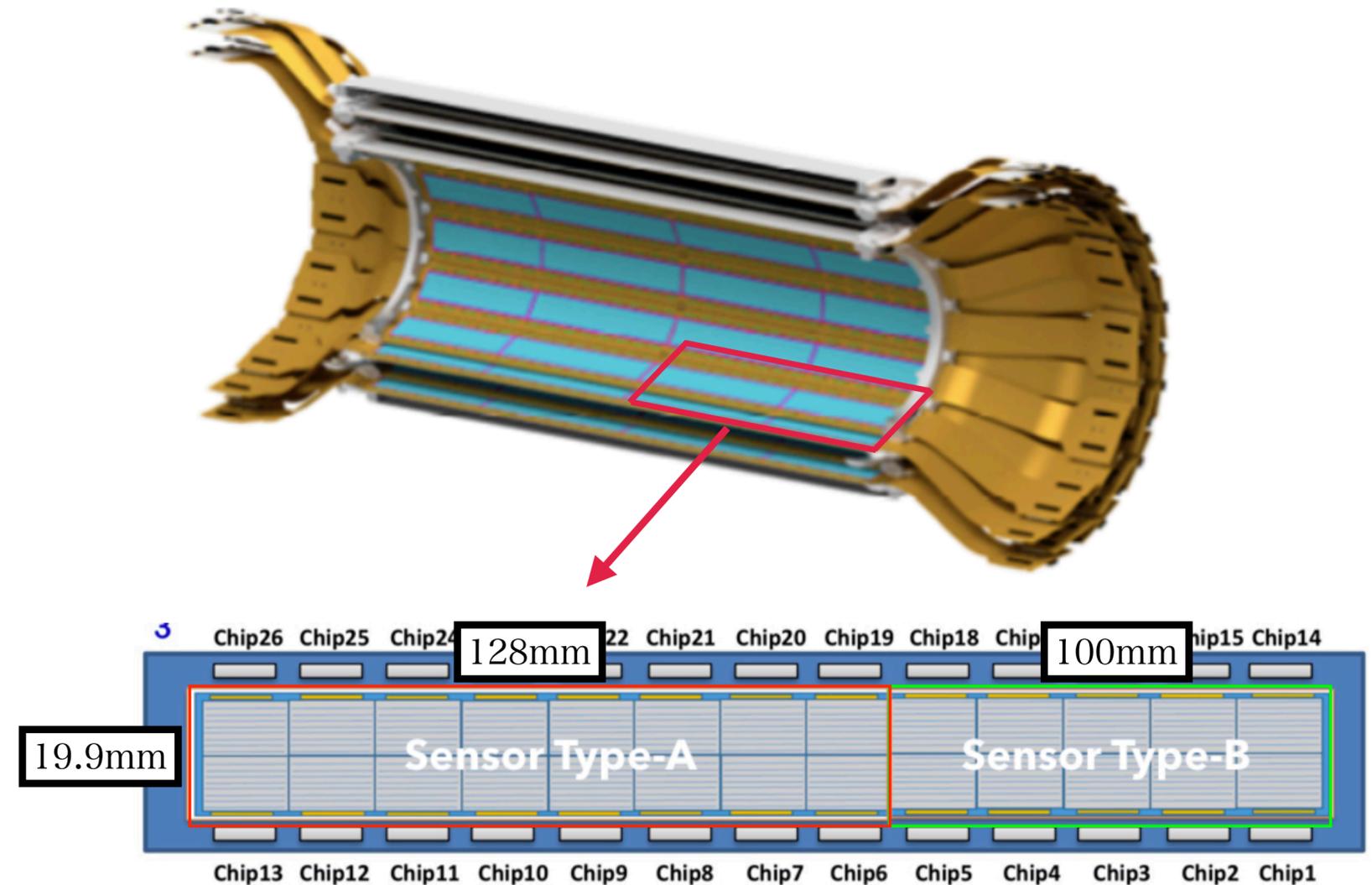
INTermediate Tracker(INTT)

- sPHENIX実験で使用される3つの飛跡検出器のうちの一つで、MVTXとTPCの間に配置される
- 時間分解能が高いことが特徴
- どのビーム交差で起きたイベントかを特定するために大きな役割を果たす



INTT用シリコンセンサー

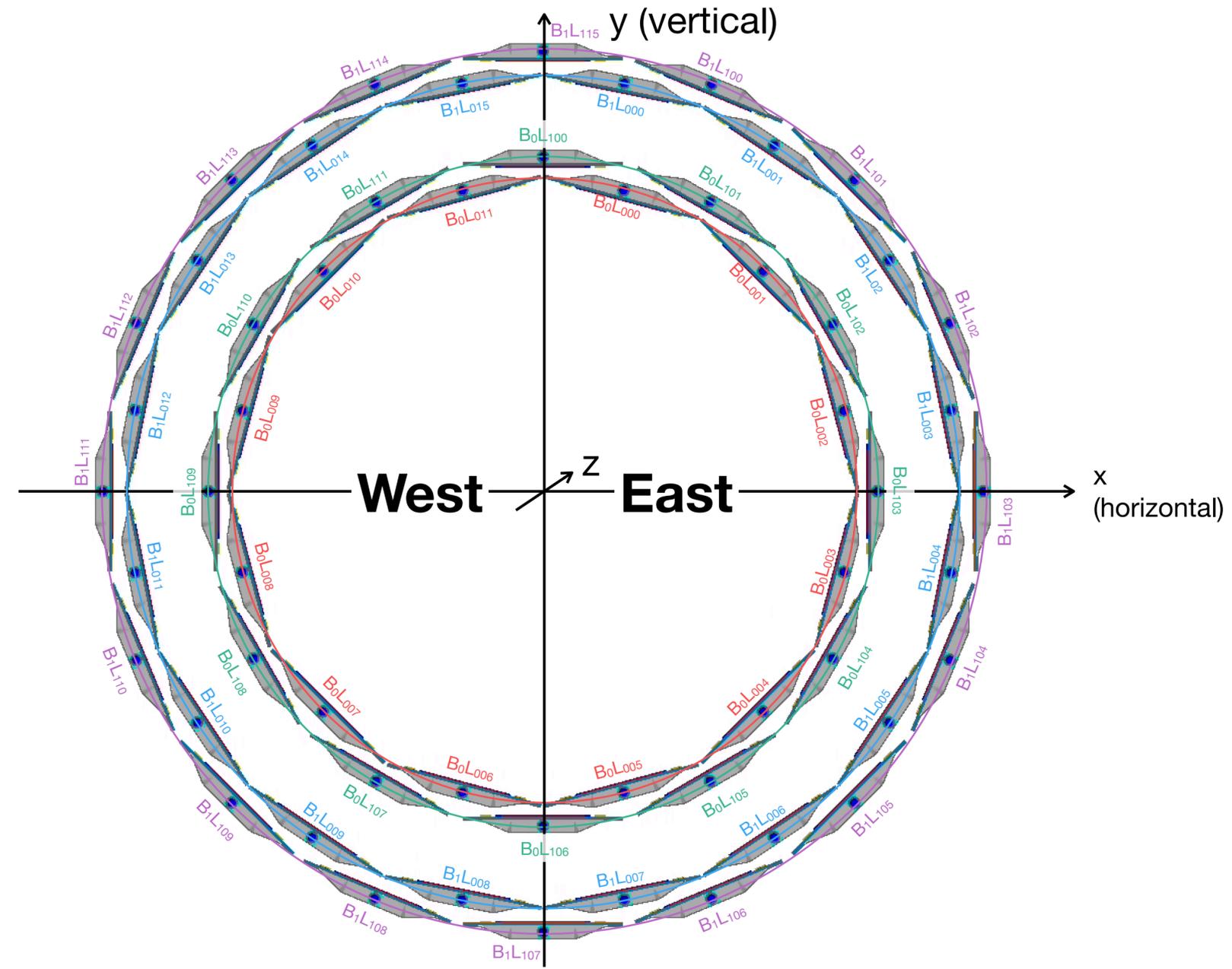
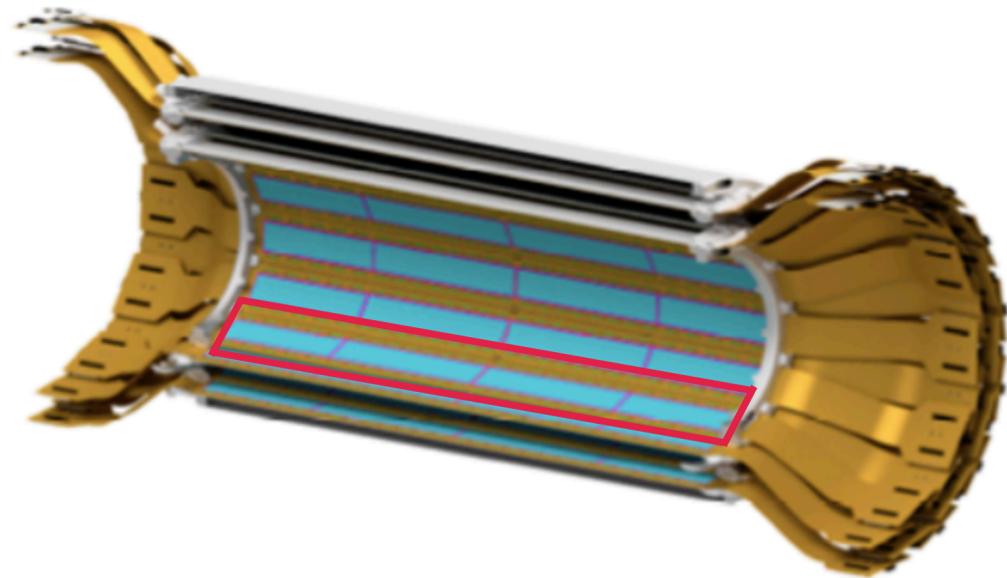
- INTT ではストリップ長の異なる typeA、typeB という 2 種類のシリコンセンサーを使用
- 26個のストリップセンサーで1つのシリコンセンサー(ハーフラダー)が構成されている
- 各センサー領域のサイズはtypeAが128mm×19.9mm、typeBが100mm×19.9mm で厚さはどちらも0.32mm



※以降、typeAセンサー、type Bセンサーのことをセグメントと呼ぶ

INTT用シリコンセンサー

- ハーフラダー2枚でフルラダーを構成している
- INTT はバレル状の2層構造で内側は24枚、外側は32枚のフルラダーで構成されている



2.研究目的

研究目的

INTT Event Display の役目

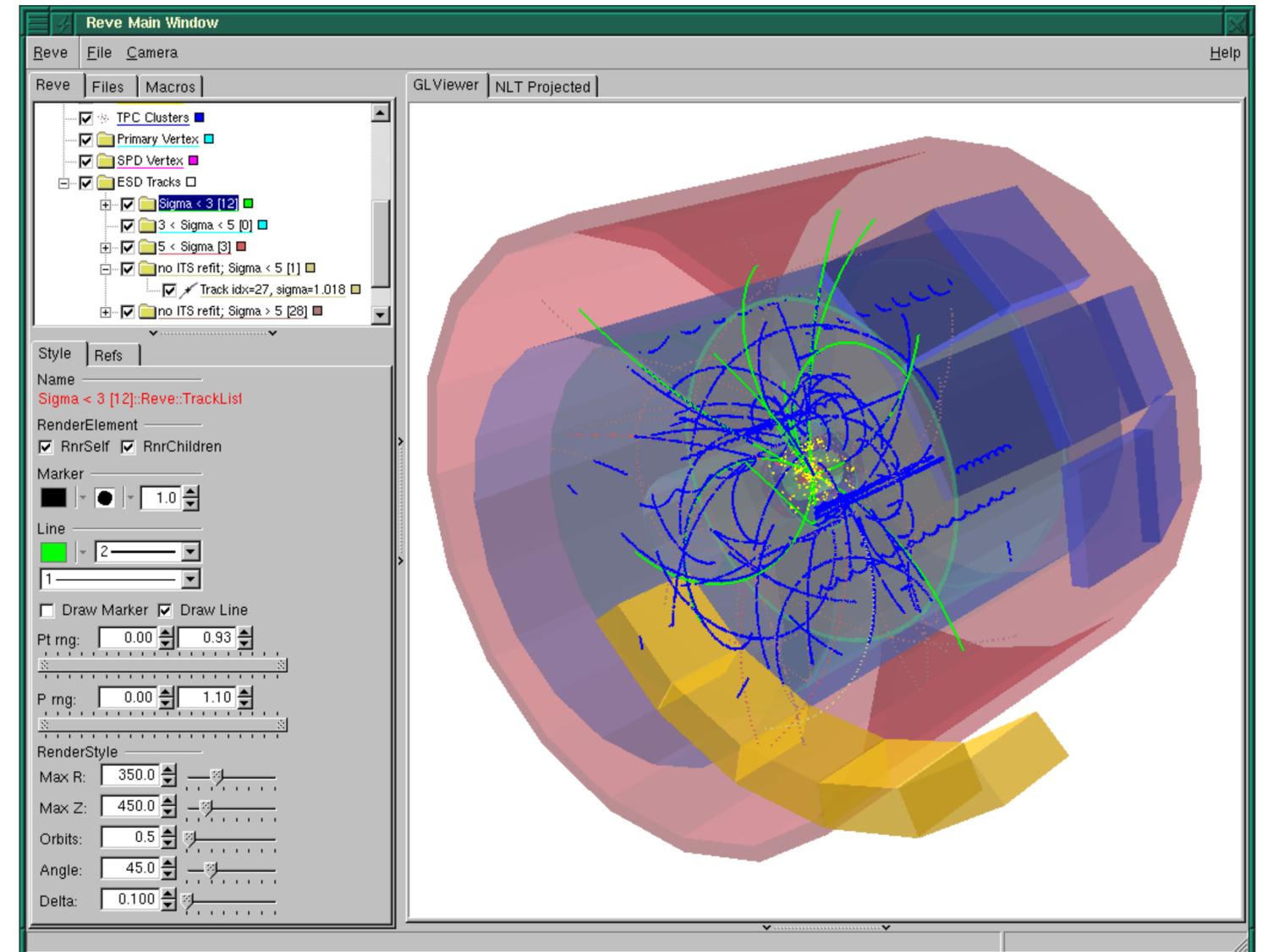
- INTT 上のヒット位置など、INTTに関する情報をイベントごとに一目見て確認する
- ラダーのアライメントを確認する
- INTT が正確に組み立てられていて、正常に稼働しているかを実験時にその場で確認する

EventDisplayとは

- 1イベント毎の粒子のヒット位置を、再構成した飛跡で結び、検出器のGeometryとともに描画したもの
- 一回の衝突を1イベントと数える
- 1イベントごとに複数回のヒットがある

イベント1

Hit 1の座標	(x_1, y_1, z_1)
Hit 2の座標	(x_2, y_2, z_2)
Hit 3の座標	(x_3, y_3, z_3)
⋮	⋮



Event Display 作成に必要な機能

- 座標空間の定義
 - Hit 位置の描画
 - Geometry の描画
- カメラの設定

ROOT

- CERN が開発した高エネルギー物理の解析に広く用いられているソフトウェア
- 基本的なプログラミング言語はC++
- 高エネルギー物理学の解析に特化したライブラリが数多く用意されている
- 本研究で主に使用しているライブラリはgeometry 作成用のTGeoとイベントディスプレイ開発用のTEve

Eve(Event Visualisation Environment)

- ROOTでサポートされているイベント可視化環境
- イベントディスプレイ作成に必要な機能
 - 座標空間の定義
 - Hit 座標の保持、表示
 - カメラの設定
 - TGeoで作成したGeometry を座標や形状を維持したまま、インポートなどが実装されている

TGeoManager

- ROOT のGeometry 作成をサポートしている
 - 検出器のGeometry 作成に必要な機能
 - 座標空間の定義
 - 箱や筒などの簡単な形状の3DCG の作成
 - オブジェクトの階層構造の構築
 - 作成したオブジェクトの回転、移動、複製
 - 作成したGeometry をEveで使える形式でファイルに書き出す
- などが実装されている

TGeoManager

Example

```
1 void Hello(){
2   gSystem->Load("libGeom");
3   TGeoManager *geom = new TGeoManager("hello", "hello");
4
5   //top作成
6   TGeoVolume *top=geom->MakeBox("Top",NULL,100.,100.,100.);
7   geom->SetTopVolume(top);
8
9   //H
10  TGeoVolume * H=geom-> MakeBox("H",NULL,100.,100.,100.);
11  H->SetVisibility(kFALSE);
12  TGeoVolume * bar1=geom->MakeBox("BOX",NULL,0.5,0.5,6.0);
13  TGeoRotation * rot1 =new TGeoRotation("rot1",0,90,0);
14  TGeoCombiTrans * combi1 =new TGeoCombiTrans(0,0,5.5,rot1);
15  H->AddNode(bar1,0,combi1);
16  H->AddNode(bar1,1,0);
17  TGeoCombiTrans * combi2 =new TGeoCombiTrans(0,0,-5.5,rot1);
18  H->AddNode(bar1,2,combi2);
19
20  //l
21  TGeoVolume * l=geom-> MakeBox("l",NULL,100.,100.,100.);
22  l->SetVisibility(kFALSE);
23  TGeoCombiTrans * combi3 =new TGeoCombiTrans(0,0,0,rot1);
24  l->AddNode(bar1,0,combi3);
25
```

座標空間を定義

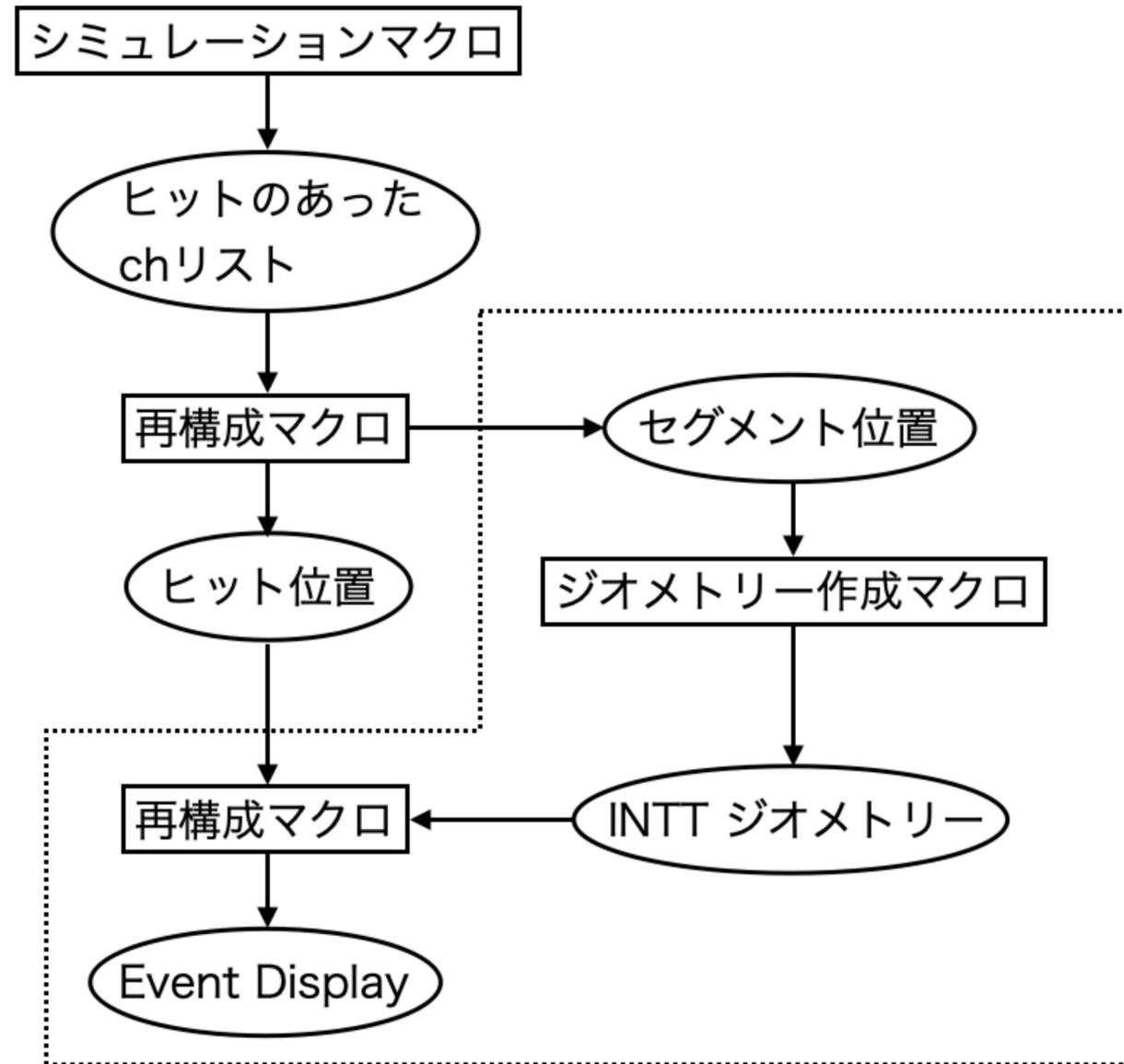
棒状のgeometryを生成
回転角度と移動する
座標を設定

生成した棒に設定し
た角度と移動を反映
して、Hという
Volumeに追加

```
26 //e
27 TGeoVolume * e=geom-> MakeBox("e",NULL,100.,100.,100.);
28 e->SetVisibility(kFALSE);
29 TGeoVolume *bar2=gGeoManager->MakeBox("BOX",NULL,0.5,0.5,4.0);
30 e->AddNode(bar2,0,0);
31 TGeoVolume *tub=gGeoManager->MakeTubs("TUB",NULL,3.,4.,0.5,90.,400.);
32 TGeoRotation * rot2=new TGeoRotation("rot2",270,90,0);
33 TGeoCombiTrans *combi4=new TGeoCombiTrans(0,0,0,rot2);
34 e->AddNode(tub,1,combi4);
35
36 //o
37 TGeoVolume * o=geom-> MakeBox("o",NULL,100.,100.,100.);
38 TGeoVolume *tube=gGeoManager->MakeTube("TUBE",NULL,3.0,4.0,0.5);
39 TGeoCombiTrans *combi5=new TGeoCombiTrans(0,0,0,rot2);
40 o->AddNode(tube,0,combi4);
41
42 TGeoTranslation * tr1 =new TGeoTranslation(0,0,-17.5);
43 top->AddNode(H,1,tr1);
44 TGeoTranslation * tr2 =new TGeoTranslation(0,-2,-5.5);
45 top->AddNode(e,2,tr2);
46 top->AddNode(l,3,0);
47 TGeoTranslation * tr3 =new TGeoTranslation(0,0,4);
48 top->AddNode(l,4,tr3);
49 TGeoTranslation * tr4 =new TGeoTranslation(0,-2,12);
50 top->AddNode(o,5,tr4);
51
52 geom->CloseGeometry();
53 top->Draw("ogl");
54 }
```

3. INTT Event Displayの開発

データフロー



再構成マクロでしている処理

- シミュレーションで生成した、ヒットのあったchのデータをヒット位置に変換
- セグメントの位置を取得、ファイルに保存
- ヒット位置と Geometry を共に描画することでEvent Display 描画

INTT Geometry 作成

- セグメントの座標を再構成マクロから取得する
- INTT のGeometry を作成する
 1. セグメントサイズに合わせた箱を作成
 2. セグメントの(x,y)座標から円の接線の角度を計算
 3. セグメントの座標と角度を設定して箱を複製

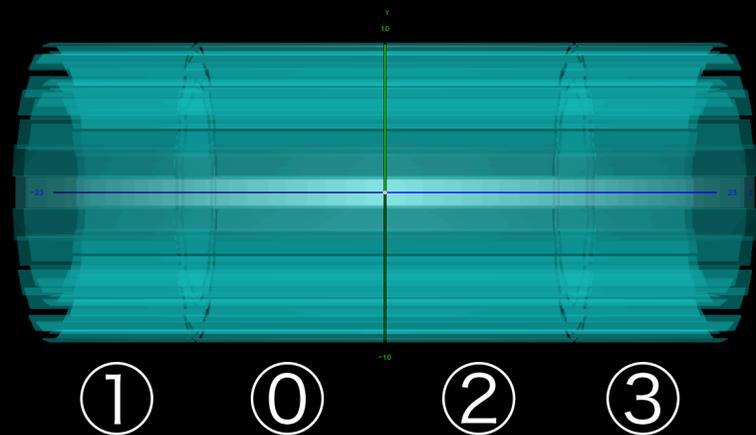
INTT Geometry の作成

セグメントの座標の取得

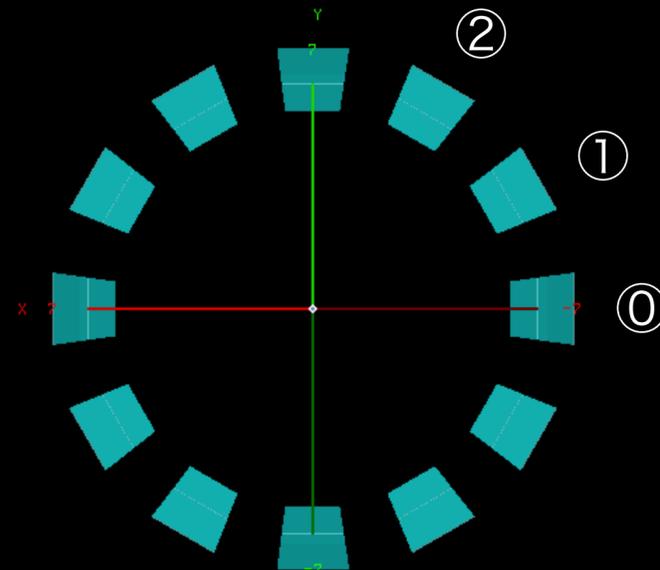
```
1187 //typeA position
1188 ladder_z_id = 0;
1189 for(uint8_t ladder_index=0;ladder_index<ladder_index_max;ladder_index++){
1190     auto genhitkeyintt = InttDefs::genHitSetKey(lyr,ladder_z_id,ladder_phi_id,
1191     time_bucket);
1192     auto surfintt = m_tGeometry->maps().getSiliconSurface(genhitkeyintt);
1193     double ladderLocation[3] = {0.,0.,0.};
1194     m_geom->find_segment_center(surfintt,m_tGeometry,ladderLocation);
1195     //cout << "ladderLocationA=(" << ladderLocation[0] << ", " << ladderLocation
1196     [1] << ", " << ladderLocation[2] << ")" << endl;
```

層, ladder z id,
ladder phi id を指定して
セグメントの座標を取得

Ladder z id



Ladder phi id



INTT Geometry の作成

Code

```
56
57 // segment
58 double segment_thickness = 0.32; // mm
59 double segment_thicknessrphi = 1.0;
60 double segment_widthY = 19.9; // mm
61 double segmentA_widthZ = 128.0; // mm
62 double segmentB_widthZ = 100.0; // mm
63 TGeoVolume *segmentA = geom->MakeBox("BOX", NULL, segment_thickness /
64 10 / 2, segment_widthY / 10 / 2, segmentA_widthZ / 10 / 2);
65 TGeoVolume *segmentB = geom->MakeBox("BOX", NULL, segment_thickness /
66 10 / 2, segment_widthY / 10 / 2, segmentB_widthZ / 10 / 2);
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119 for (int i = 0; i < sum_segmentA; i++)
120 {
121     phi = atan2(segmentALocationY[i], segmentALocationX[i])
122     * 180 / M_PI;
123     rotA[i] = new TGeoRotation(Form("rotA[%d]", i), phi,
124     0., 0.);
125     combitranslyrA[i] = new TGeoCombiTrans(segmentALocationX
126     [i], segmentALocationY[i], segmentALocationZ[i], rotA
127     [i]);
128     lyrA->AddNode(segmentA, i, combitranslyrA[i]);
129 }
130
131
132
133
134
135
```

セグメントのサイズを設定

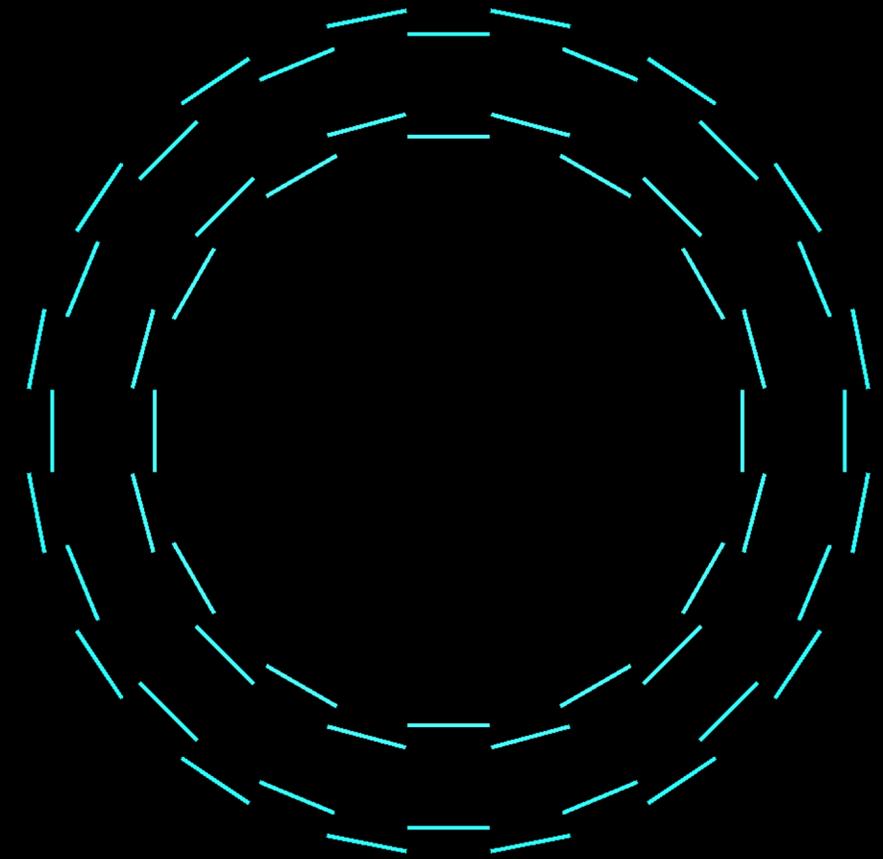
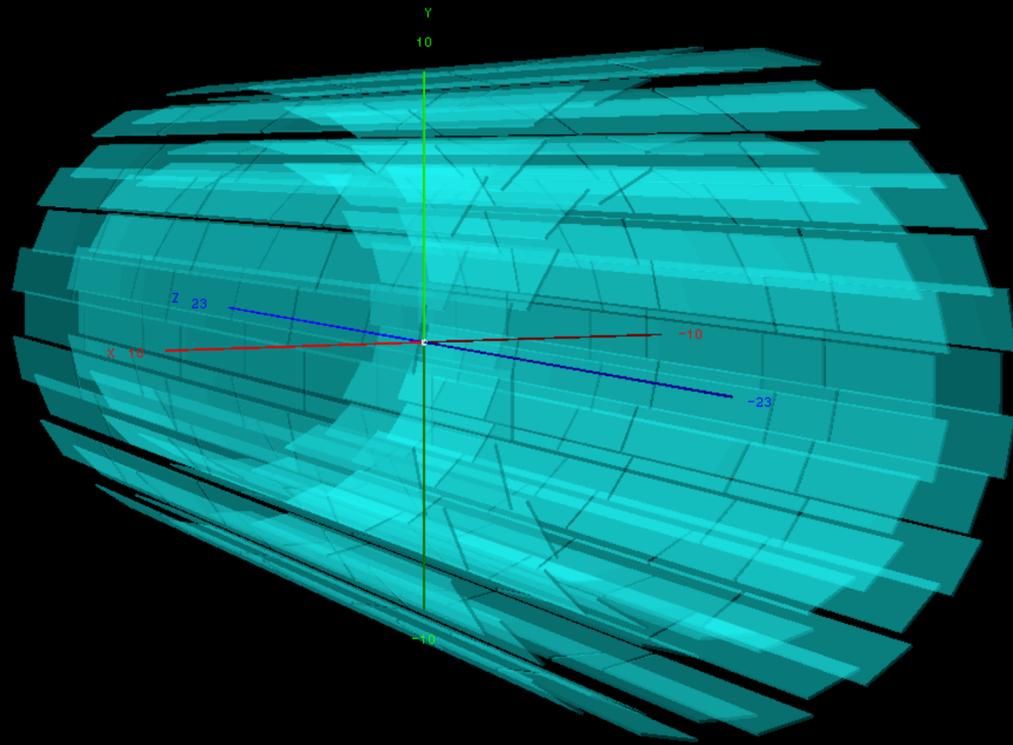
セグメントサイズに合わせた箱を作成

回転角度を計算

回転角度と座標を設定

設定を反映してセグメントを複製

INTT Geometry



INTT Event Displayに実装されている機能

- ヒット位置とINTT Geometry を同時に描画
- 実験データと同じ形式のシミュレーションデータを読み込み、Event Display を出力する

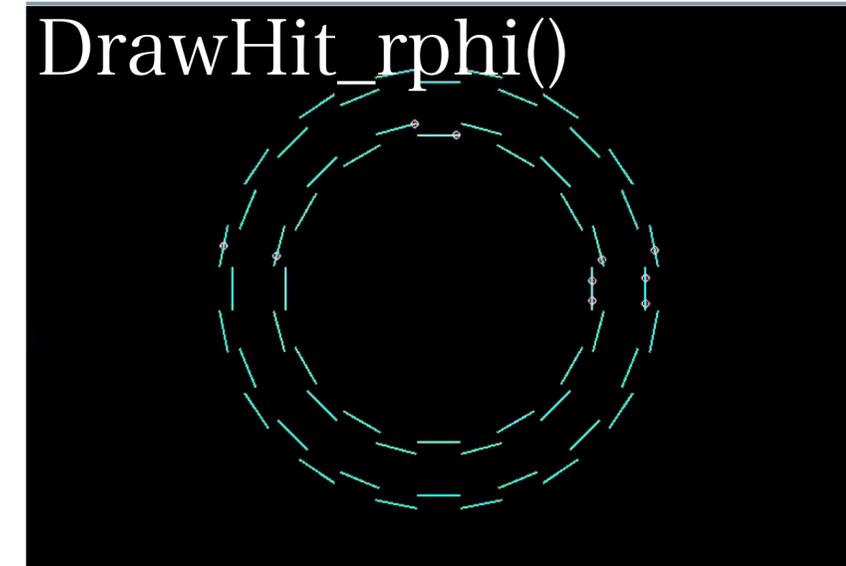
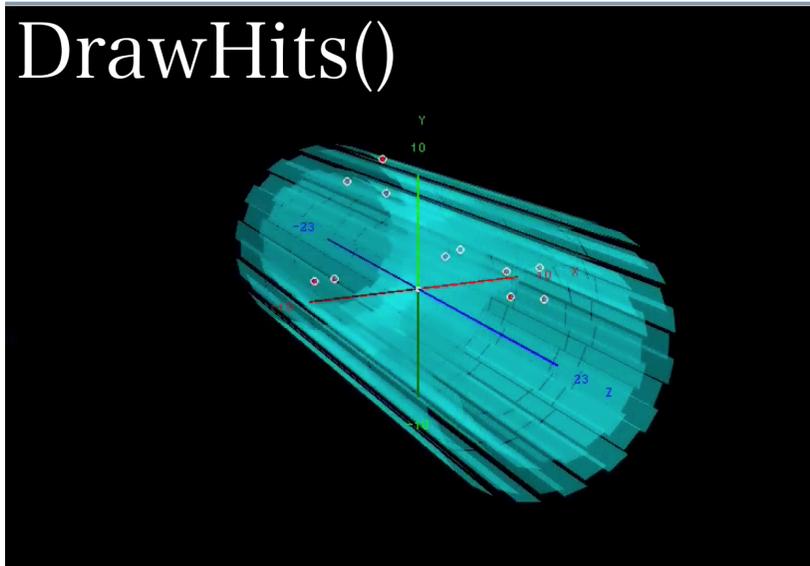
Viewer機能

- $r-\phi$ プロジェクション

Event Display表示の仕方

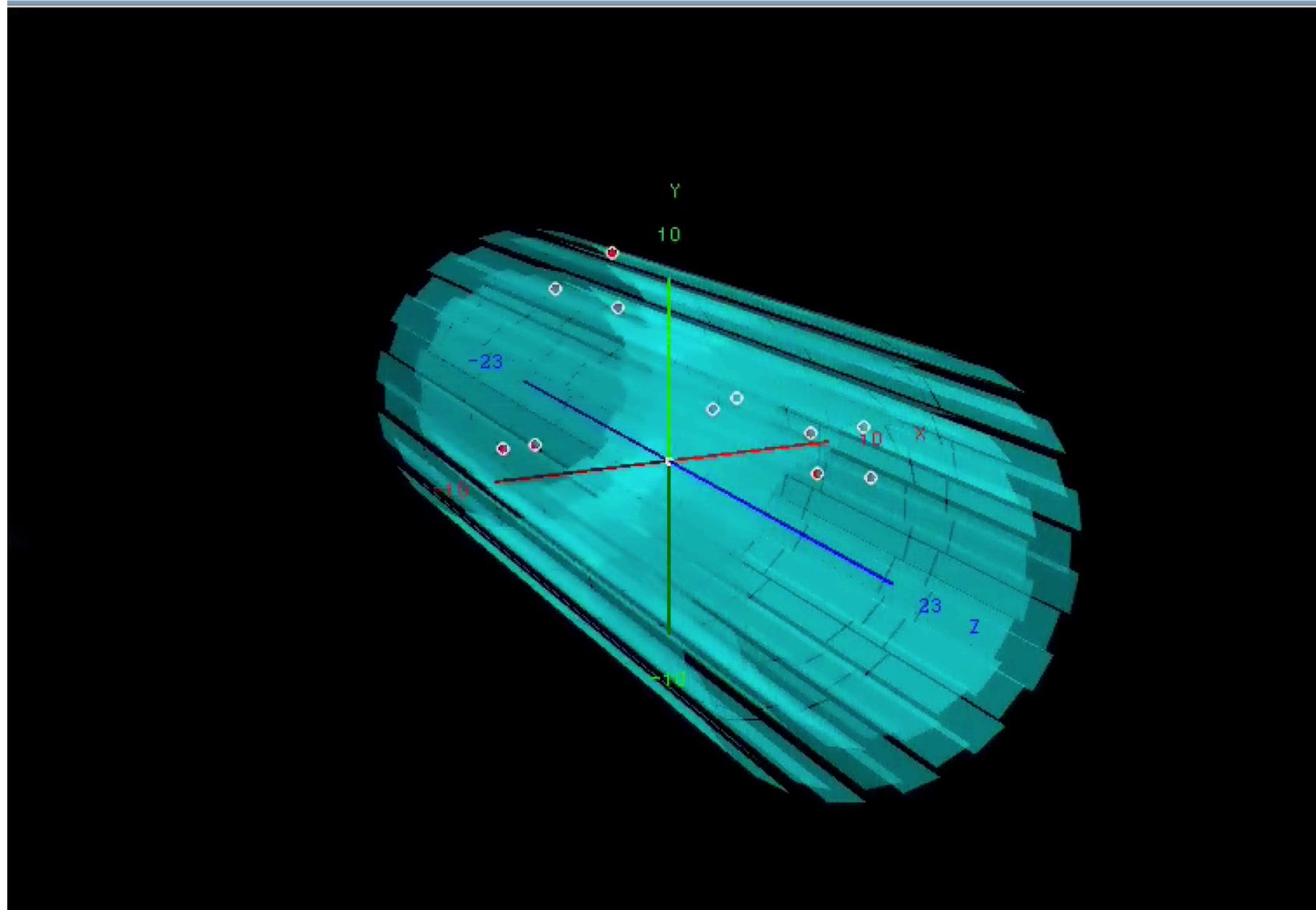
1. root Loadfile.C でファイルを読み込む
2. anaTutorial->DrawHits() or DrawHit_rphi() でEvent Displayを表示
DrawHits()は3D表示、DrawHit_rphi()は輪切りのような表示
3. 次のイベントを見るにはse->run(1) の後に2. のコマンドをもう一度入力

※前のイベントには戻れない



pp衝突のEvent Display

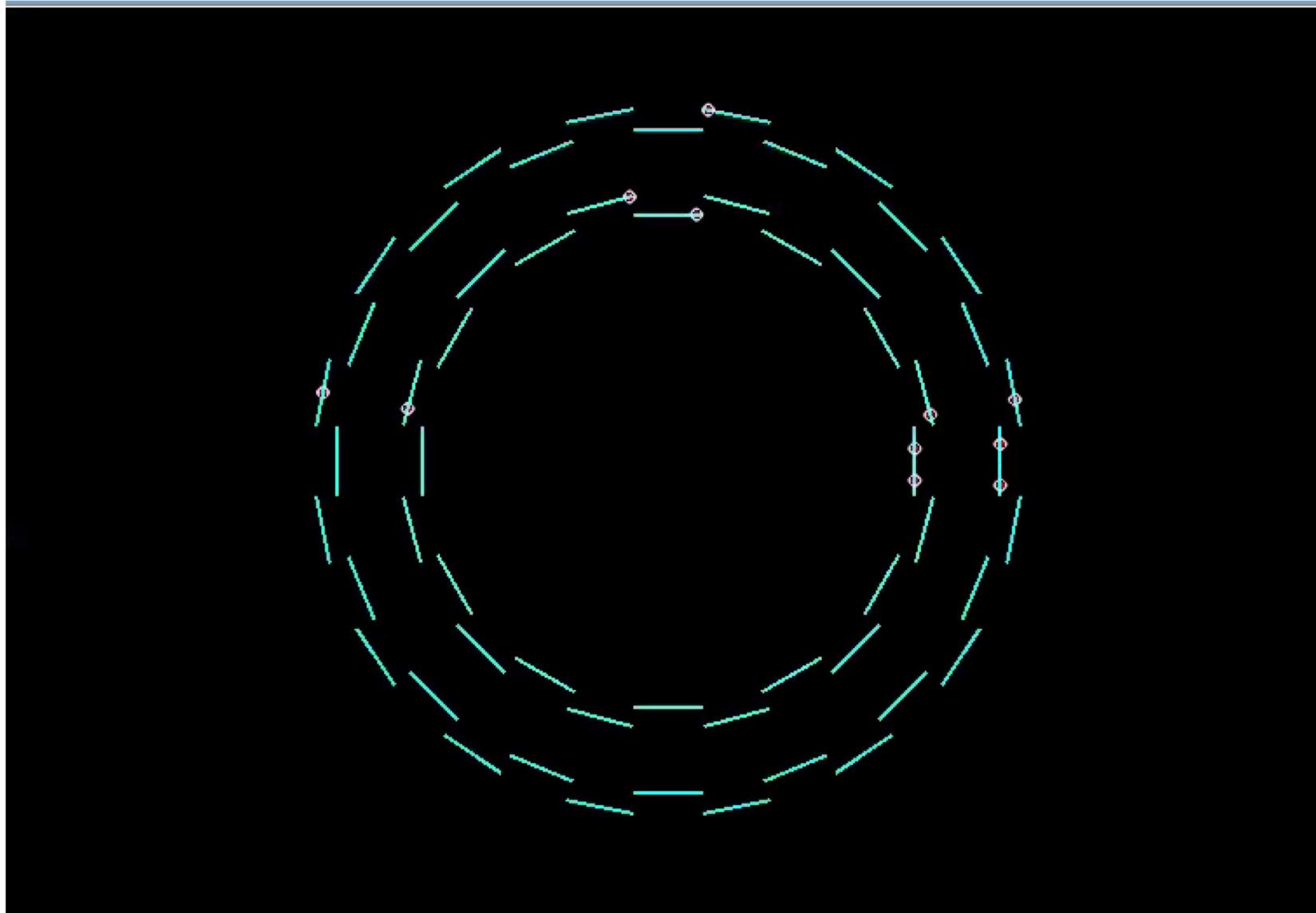
3D表示



- 水色がINTT
- 赤の点がヒット位置

pp衝突のEvent Display

R- ϕ プロジェクション



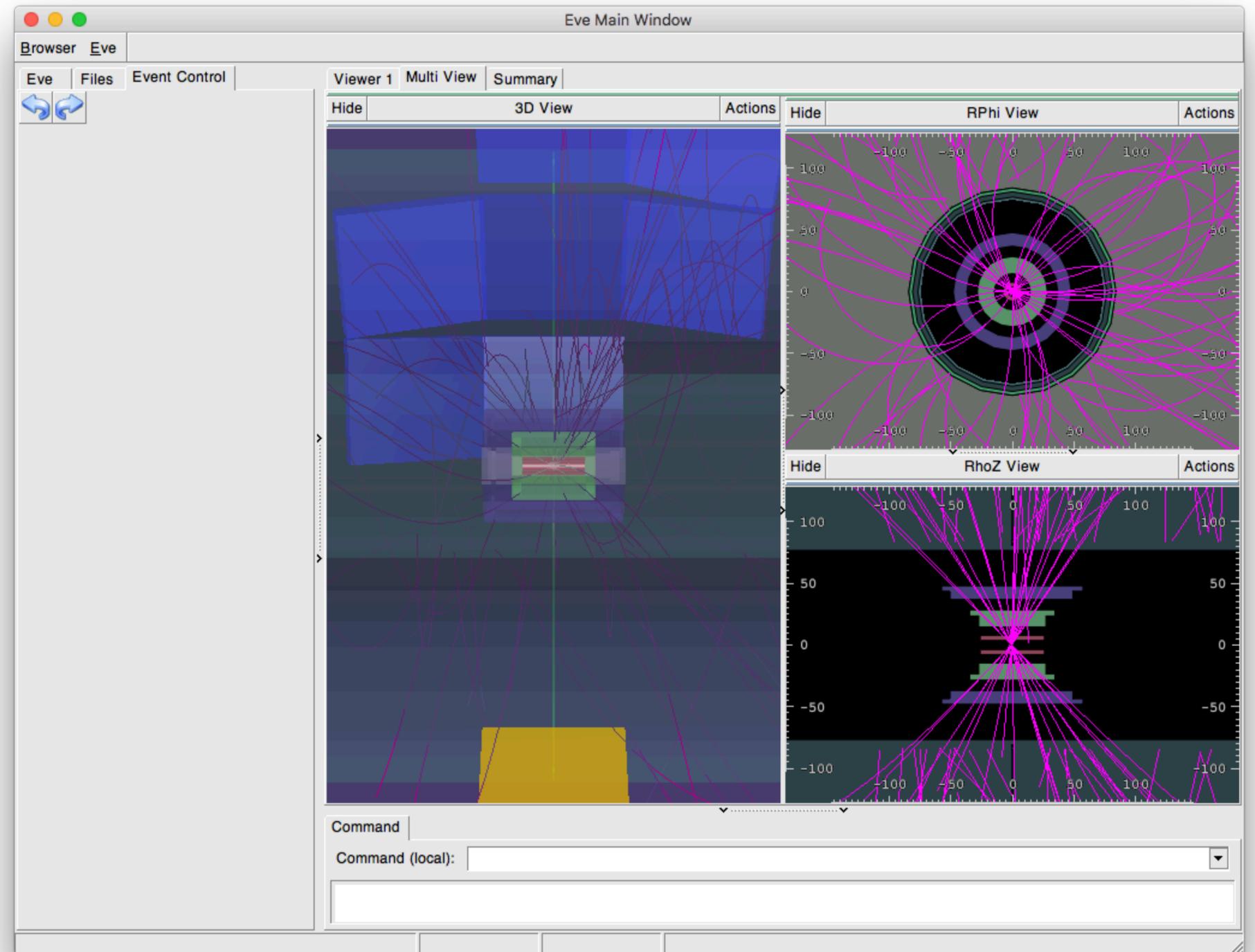
pp衝突のEvent Display

```
NoMachine - bnl.server.com
Terminal
File Edit View Search Terminal Help
-5.21385
  10.5
cluster[0] = -5.71576  cluster[1] = -5.21385  cluster[2] = 10.5
itr = 7.42166
-2.16224
  7.3
cluster[0] = 7.42166  cluster[1] = -2.16224  cluster[2] = 7.3
itr = -2.82383
  9.30399
  -7.3
cluster[0] = -2.82383  cluster[1] = 9.30399  cluster[2] = -7.3
itr = -6.8064
  6.8781
  12.1
cluster[0] = -6.8064  cluster[1] = 6.8781  cluster[2] = 12.1
itr = -7.15387
-6.53063
  12.1
cluster[0] = -7.15387  cluster[1] = -6.53063  cluster[2] = 12.1
itr = -4.6218
-8.55925
  20.1
cluster[0] = -4.6218  cluster[1] = -8.55925  cluster[2] = 20.1
itr = -5.55982
  8.62272
  10.5
cluster[0] = -5.55982  cluster[1] = 8.62272  cluster[2] = 10.5
itr = 9.88857
-2.8696
  7.3
cluster[0] = 9.88857  cluster[1] = -2.8696  cluster[2] = 7.3
Apoints =0
Bpoints =0
TEveManager::GetGeometry loading: '/sphenix/u/mfujiiwara/Documents/inttgeometry.root' -> '/sp
henix/u/mfujiiwara/Documents/inttgeometry.root'.
Warning in <TEveManager::GetGeometry >: TGeoManager is locked ... unlocking it.
root [7] .q
-bash-4.2$
```

4. 今後の課題とまとめ

今後の課題

- Au+Au 衝突のデータの可視化
- ρ -z プロジェクションの実装
- Tracking の実装



まとめ

- INTT の情報を可視化し、実験時に確認するためのツールである Event Display を開発した
- 今後も ρ -z プロジェクション、Tracking など必要な機能を随時実装予定である

Back Up

Event Display 開発

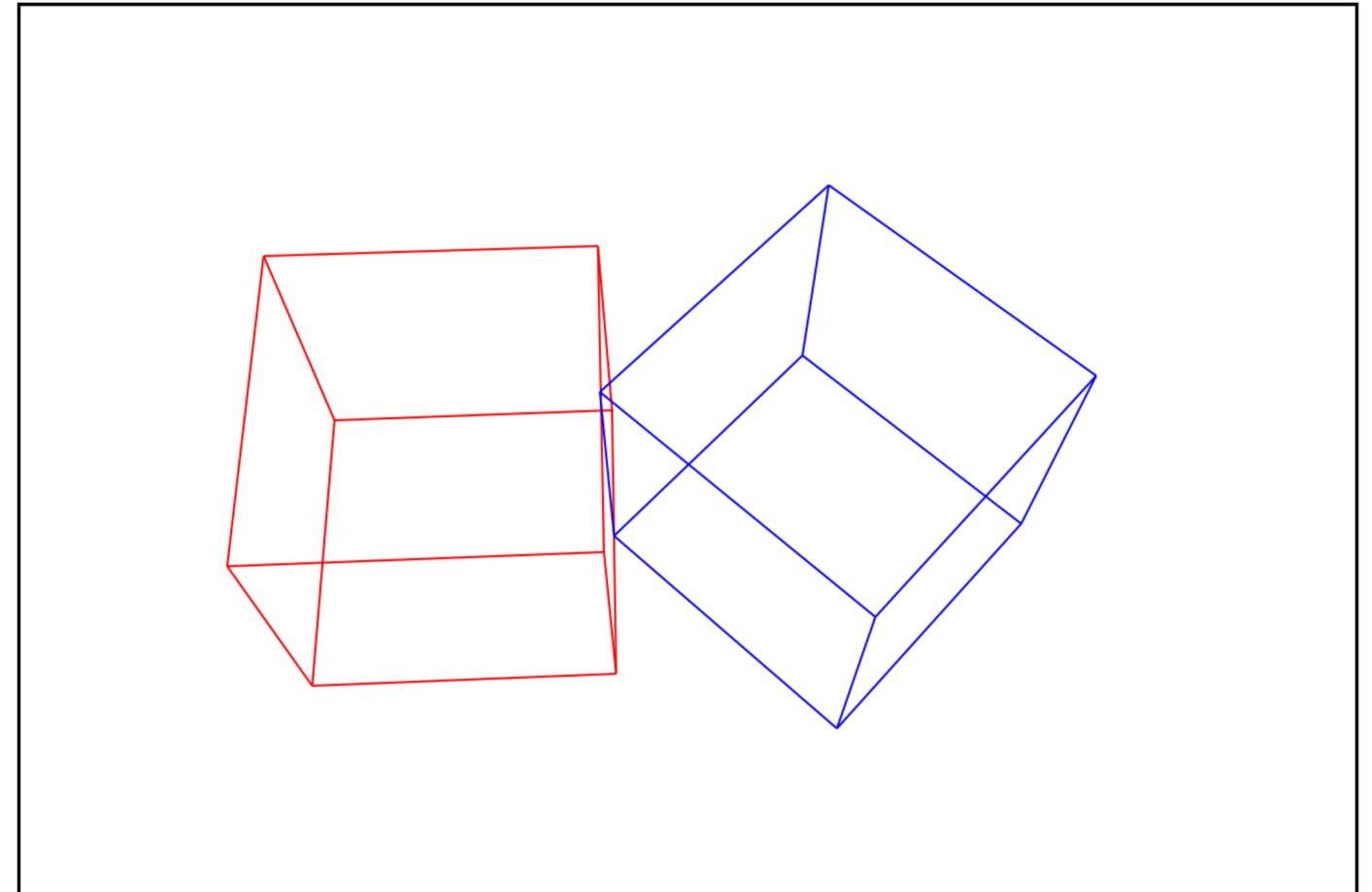
動作環境

- ROOT 6.24/06
- g++ (GCC) 8.3.0
- リモートデスクトップ用ソフト NoMachine を通して実行可能

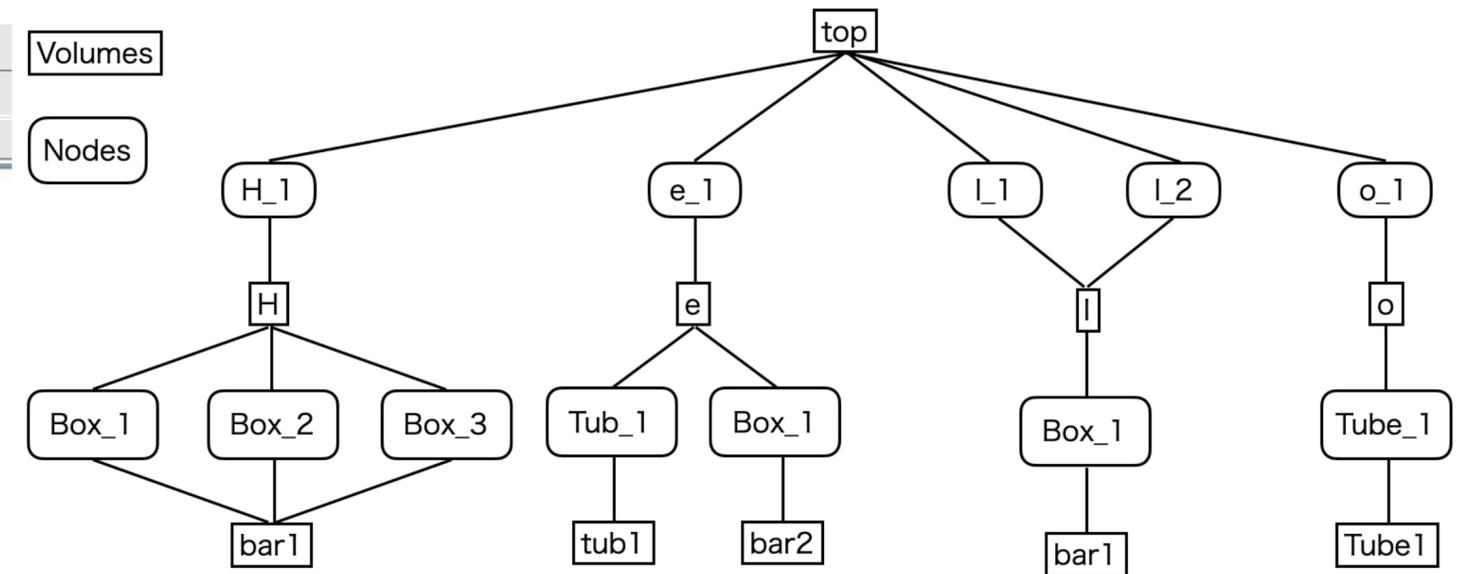
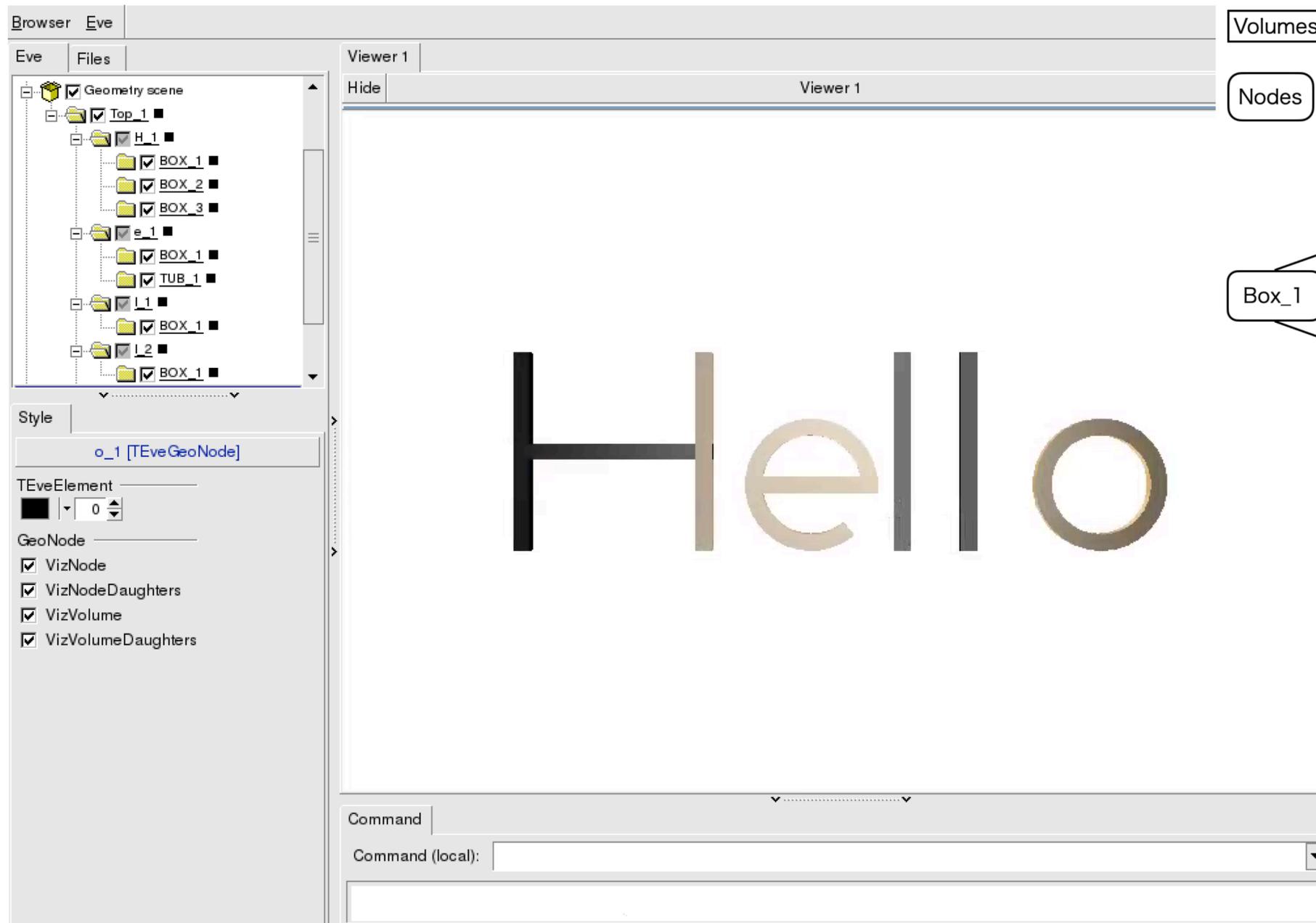
TGeoManager

Example

```
1 void makebox()
2 {
3     TGeoManager *gGeoManager = new TGeoManager("boxes", "boxes");
4     TGeoVolume *top = gGeoManager->MakeBox("Top", NULL, 100., 100., 100.);
5     gGeoManager->SetTopVolume(top);
6     top->SetVisibility(kFALSE);
7
8     TGeoVolume *boxred = gGeoManager->MakeBox("box", NULL, 1., 1., 1.);
9     TGeoVolume *boxblue = gGeoManager->MakeBox("box", NULL, 1., 1., 1.);
10
11     boxred->SetLineColor(kRed);
12     boxblue->SetLineColor(kBlue);
13
14     TGeoRotation *rot = new TGeoRotation("rot", 45., 0., 0.);
15     TGeoCombiTrans *box0 = new TGeoCombiTrans(1.0+sqrt(2), 0., 0., rot);
16
17     top->AddNode(boxred, 1, 0);
18     top->AddNode(boxblue, 2, box0);
19
20     gGeoManager->CloseGeometry();
21     top->Draw();
22 }
```



Exampleの階層構造



- Volume は箱などの3DCGオブジェクト
- Node は他のVolume にVolumeを追加すると自動で生成される
- 親オブジェクトを移動、回転させると子オブジェクトも移動、回転する

データの読み込み

```
43
44 void Loadfile(const char*inputfile="G4sPHENIX_10event.root"){
45     se = Fun4AllServer::instance();
46     se->Verbosity(0);
47     const string &outputFile = "G4sPHENIX";
48     string outputroot = outputFile;
49     const int nEvents = 1;
50
51     Fun4AllInputManager*in = new Fun4AllDstInputManager("DSTin");
52     in->fileopen(inputfile);
53     se->registerInputManager(in);
54
55     ファイルデータ読み込み
56
57     Enable::MICROMEGAS = true;
58
59     TrackingInit();
60     Tracking_Reco();
61
62     cout<<"before Anatutorial"<<endl;
63     anaTutorial = new AnaTutorial("anaTutorial", outputroot + "_anaTutorial.
64     root");
```

↑ データファイルの名前

ファイルデータ読み込み

```
63
64     anaTutorial->setMinJetPt(10.);
65     cout<<"setMinJetPt"<<endl;
66     anaTutorial->Verbosity(0);
67     cout<<"Verbosity(0)"<<endl;
68     anaTutorial->analyzeTracks(true);
69     cout<<"analyzeTracks(true)"<<endl;
70     anaTutorial->analyzeClusters(true);
71     cout<<"analyzeClusters(true)"<<endl;
72     anaTutorial->analyzeJets(true);
73     cout<<"analyzeJets"<<endl;
74     anaTutorial->analyzeTruth(false);
75     cout<<"analyzeTruth(false)"<<endl;
76     se->registerSubsystem(anaTutorial);
77
78     cout<<"after Anatutorial"<<endl;
79
80     se->run(nEvents);
81     return 0;
82 }
```

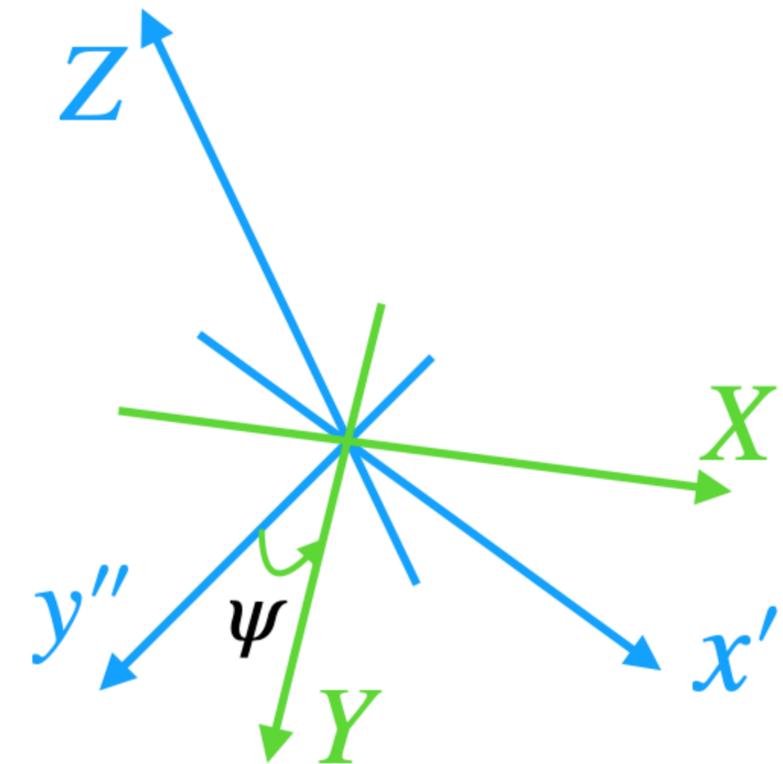
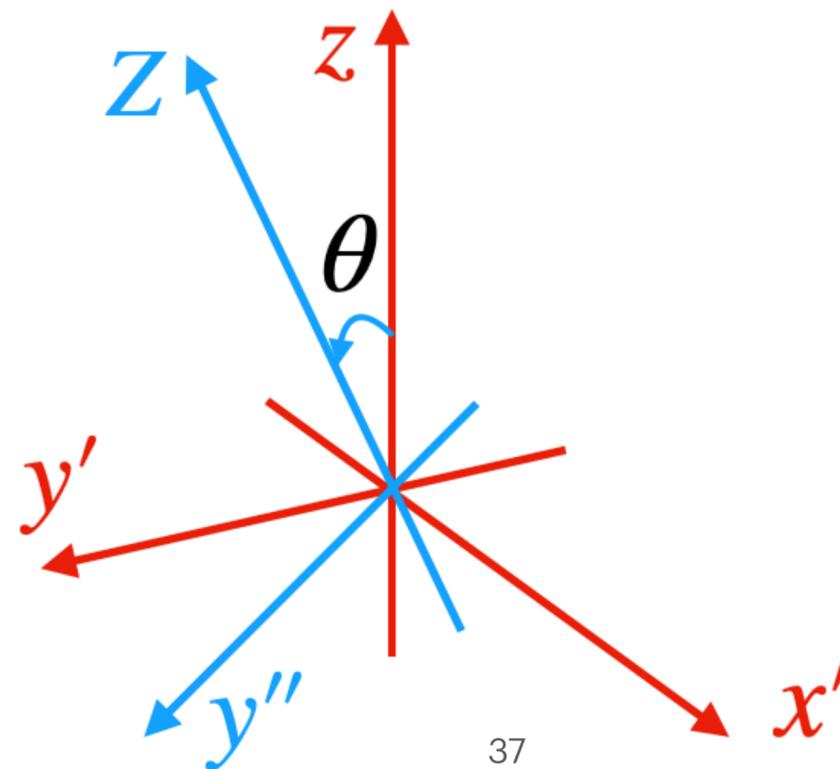
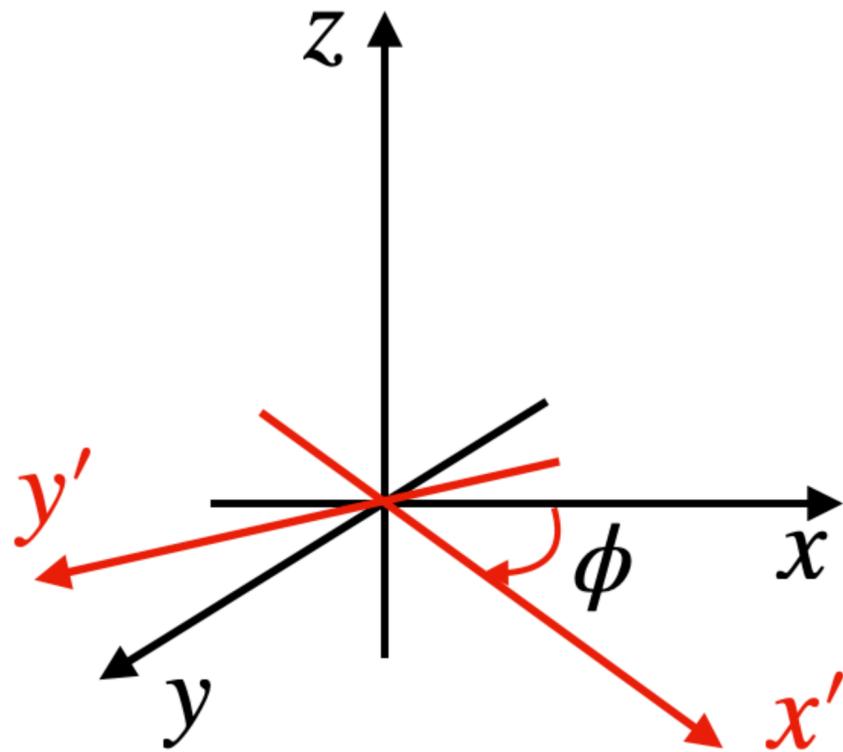
イベントディスプレイ描画

マクロにシミュレーション

データを渡す

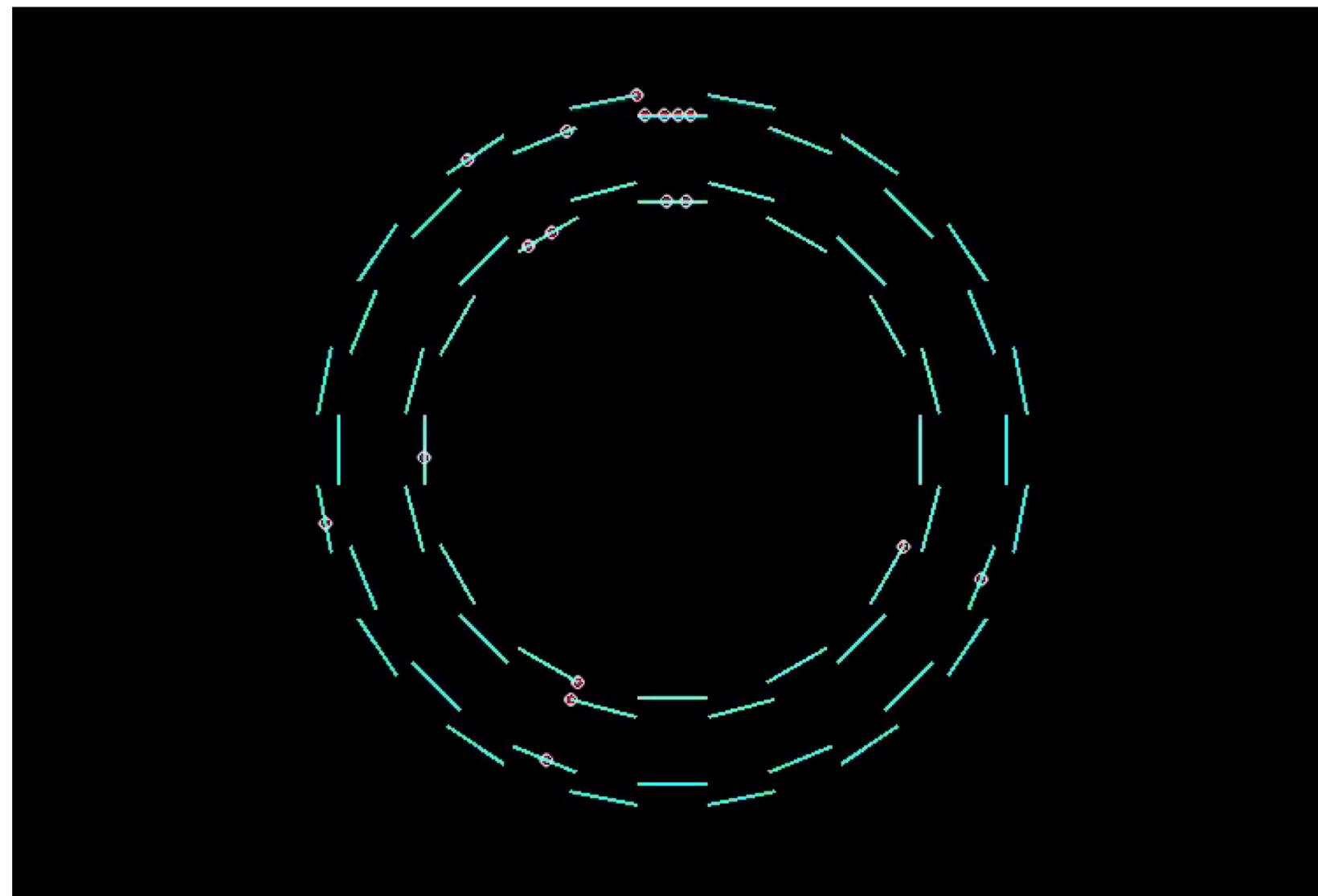
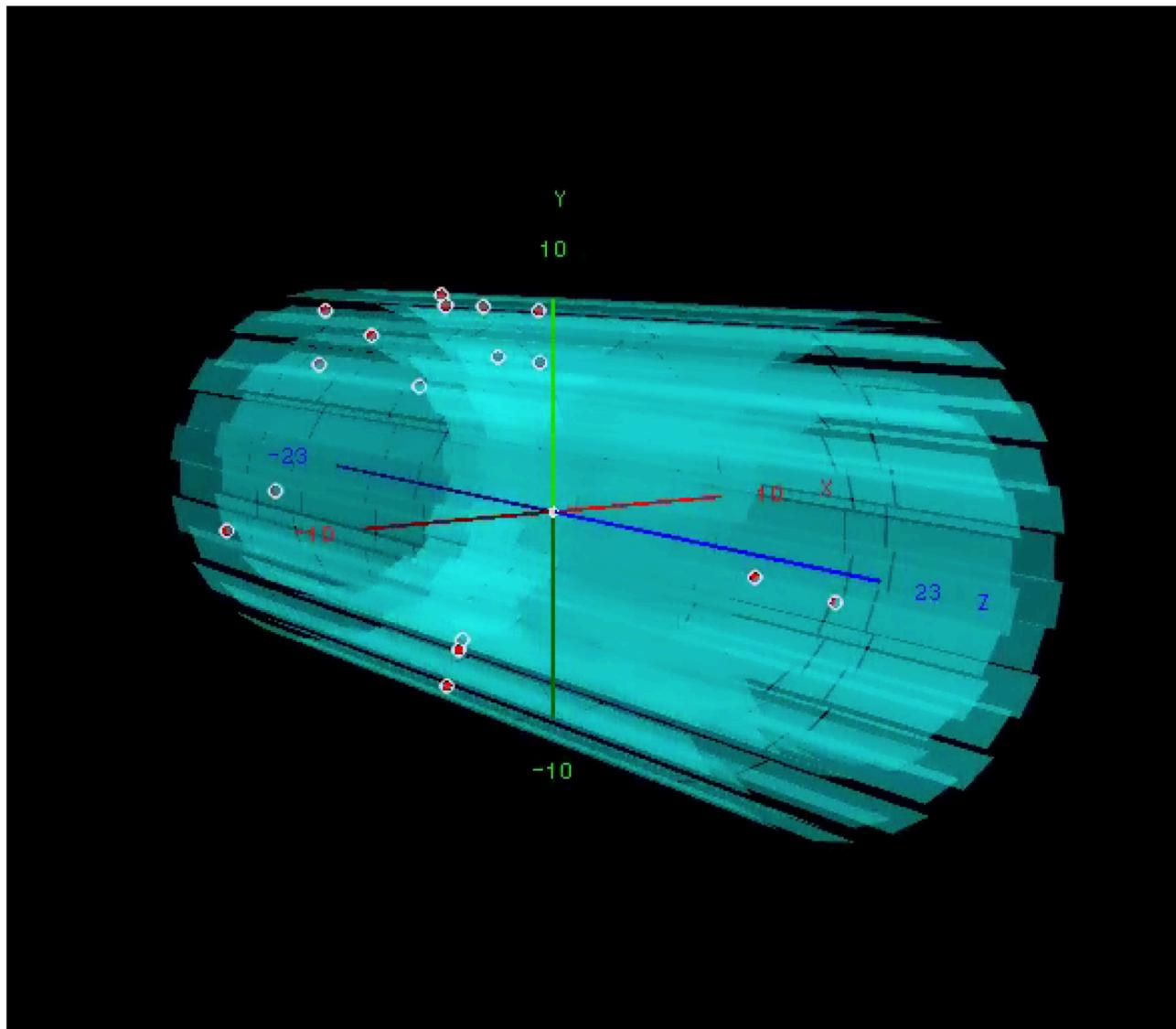
オイラー角

- 最初にz軸を中心に角度 ϕ 回転、2番目にx軸を中心に角度 θ 回転、最後に回転後のz軸を中心に角度 ψ 回転させる。
- TGeoRotation では、単位は度



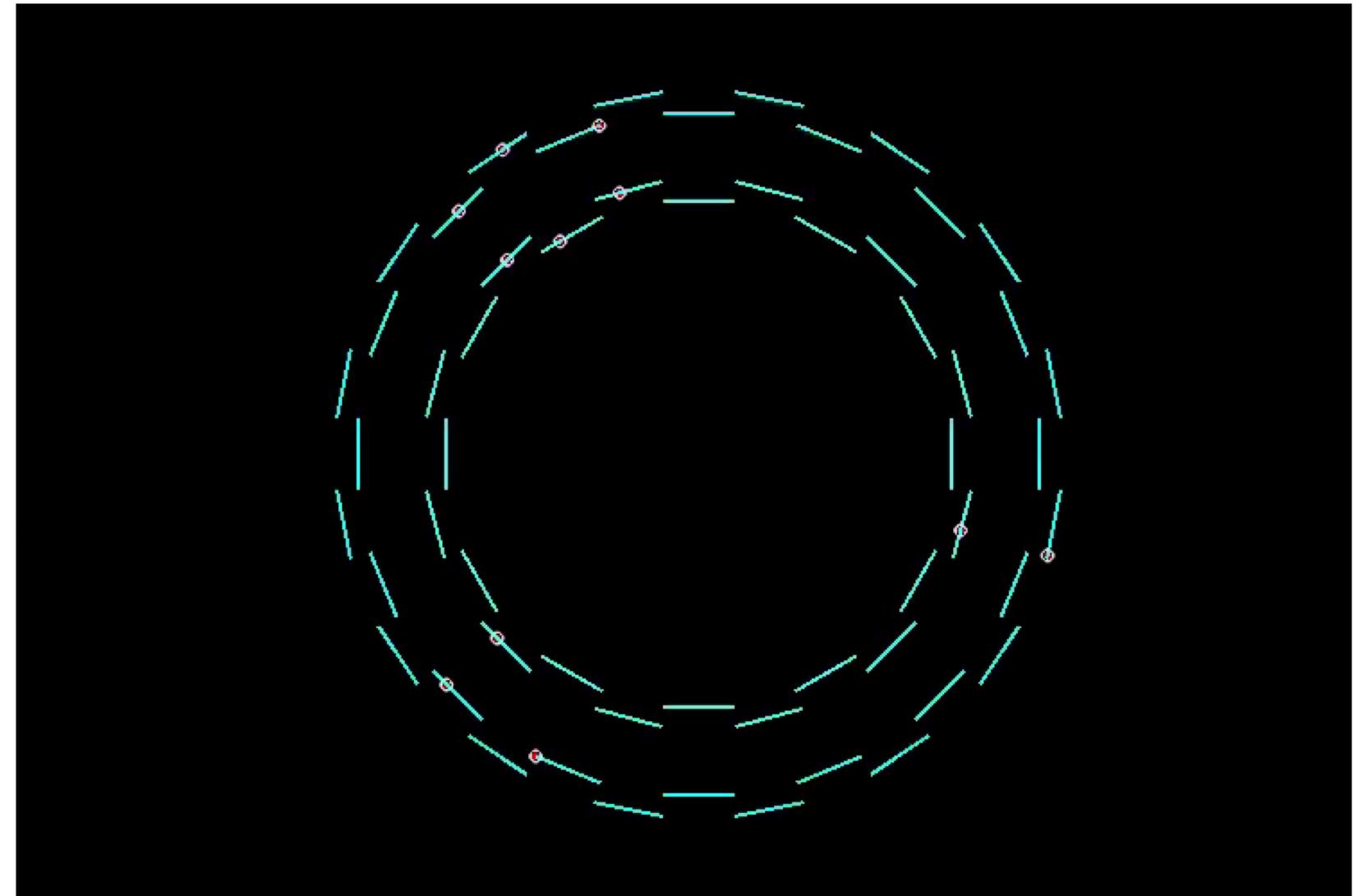
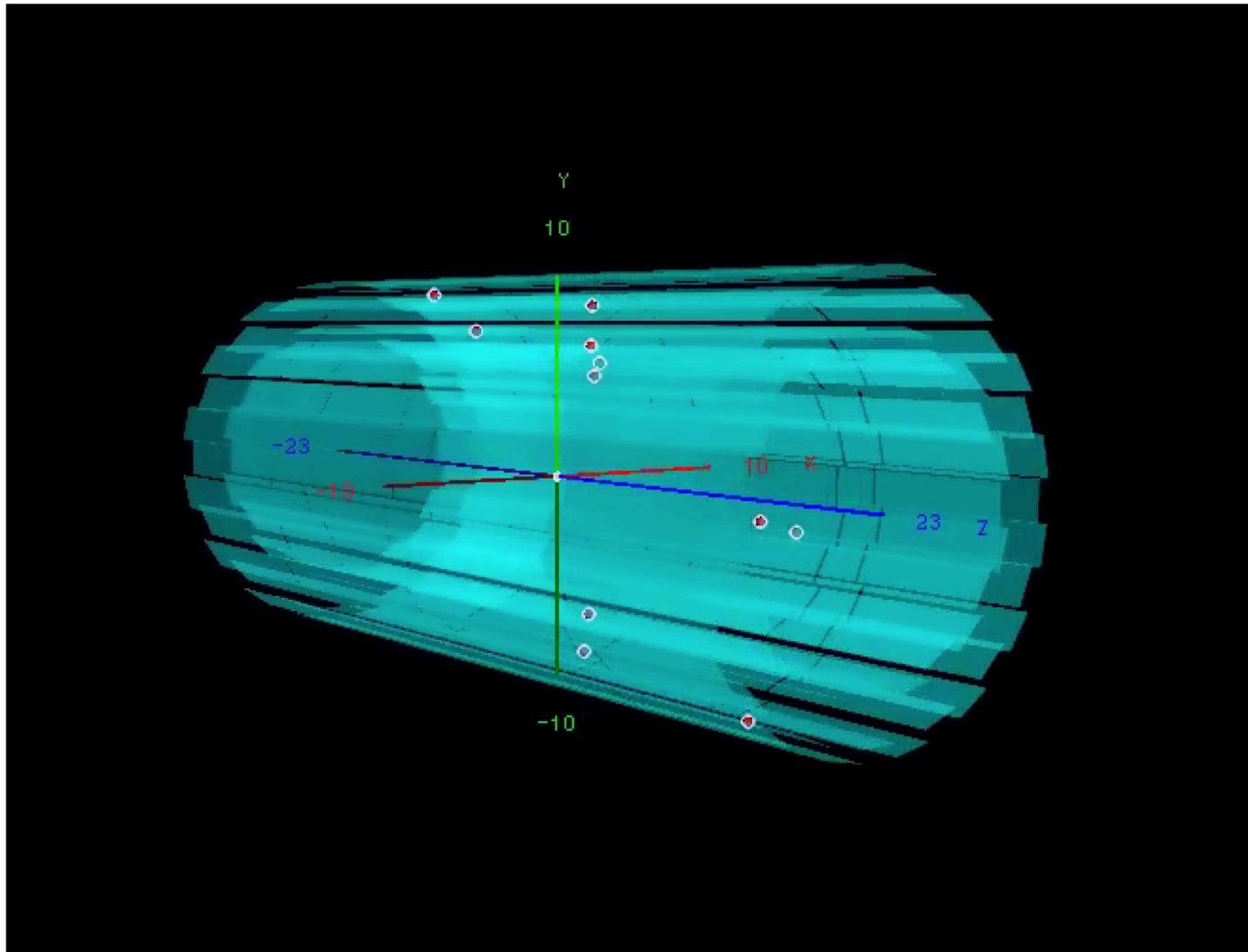
pp衝突のEvent Display

Event2



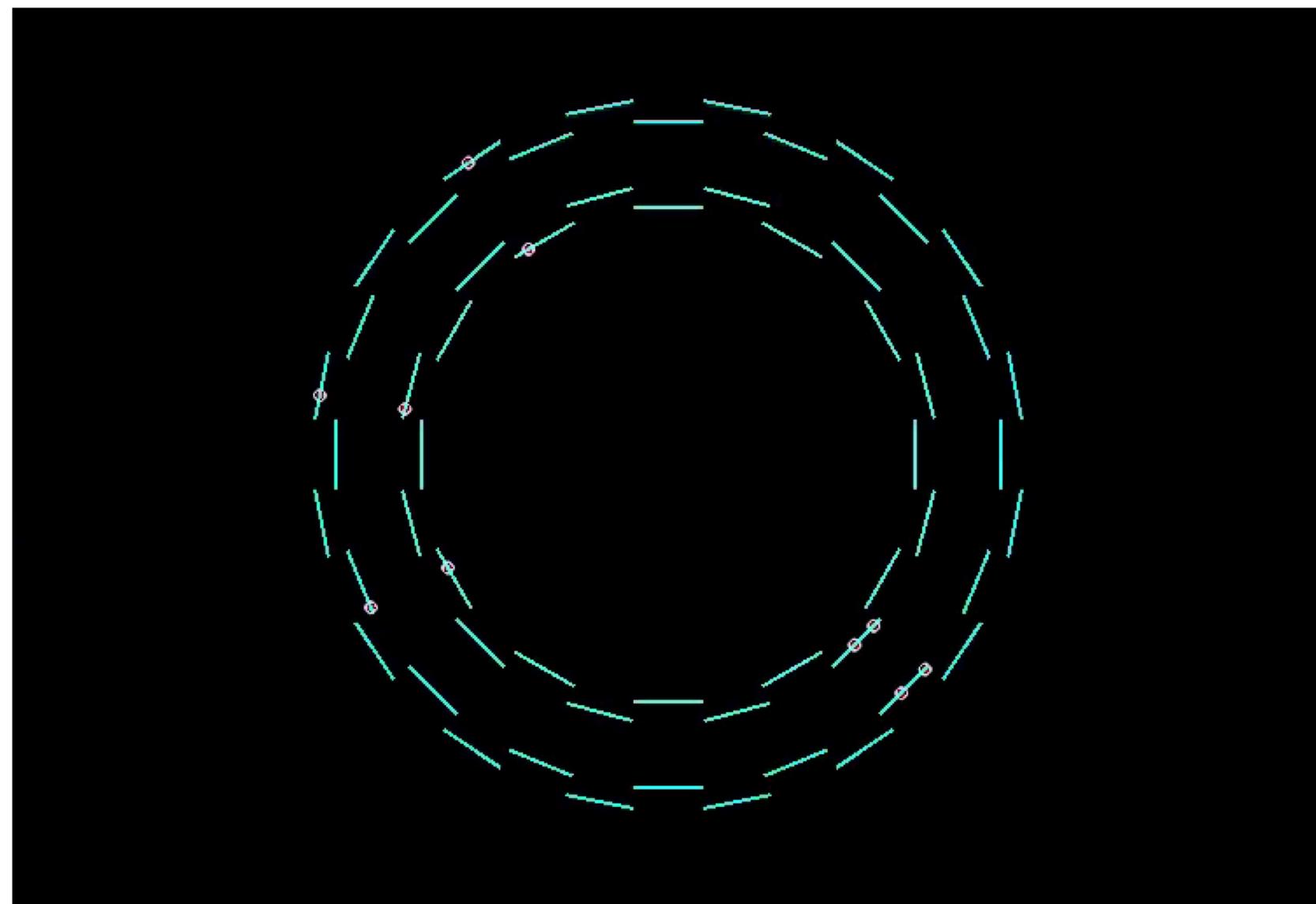
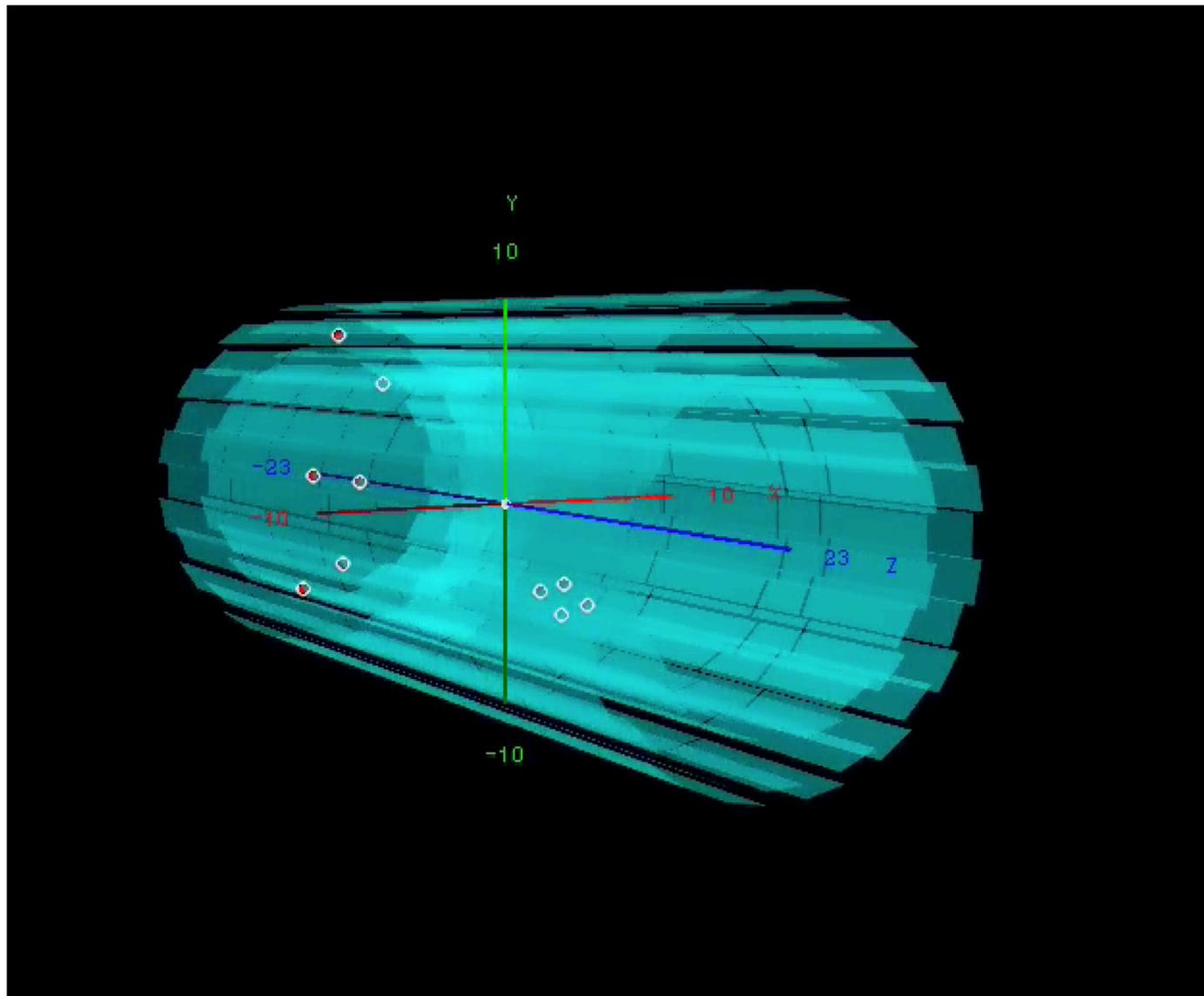
pp衝突のEvent Display

Event3



pp衝突のEvent Display

Event4



画像

- QGP <http://alice-j.org/クォーク・グルーオンプラズマ-qgp/>
- RICH <https://www.bnl.gov/newsroom/news.php?a=119262>
- sPHENIX 検出器 <https://www.bnl.gov/rhic/sphenix.php>
- INTT ladder https://webhepl.cc.nara-wu.ac.jp/old_HP/thesis/4kaisei/2020/nishimori_b4thesis.pdf
- INTT barrel https://wiki.sphenix.bnl.gov/index.php/File:20220504_Geant4_ladder_ID.pdf
- Event Display https://root.cern/doc/v610/group_TEve.html
- Rho-z https://root.cern/doc/v610/alice_esd_8C.html