



# **Laser calibration system and photodot reconstruction**

Szymon Bugiel

AGH University of Science and Technology, Poland

Summer student Program 2012

Supervisors: Klaus Zenker, Ralf Dinner

September 5, 2012

## **Abstract**

During summer student program 2012 in DESY I was working for the ILC TPC group. The main aim of my project was to develop photodot reconstruction process used in laser calibration.

The laser calibration system is used in order to determine field distortions inside time projection chamber. It is a simple tool with can be use to find out if electrons moves straight to the anode, or there are some deviation with influence data. With this system you are also able to calculate some basic parameters of a TPC like for example drift velocity or diffusion.

# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	Time Projection Chamber . . . . .	3
<b>2</b>	<b>Laser calibration system</b>	<b>3</b>
2.1	Motivations . . . . .	3
2.2	Working principle . . . . .	4
2.3	Setup . . . . .	4
2.4	Analyzed data . . . . .	5
<b>3</b>	<b>Photodot reconstruction</b>	<b>5</b>
3.1	Overview . . . . .	5
3.2	Running a reconstruction . . . . .	6
3.3	Methods . . . . .	7
3.3.1	void findPhotodotPulses() . . . . .	7
3.3.2	int findNearestHit(Photodot thisPhotodot) . . . . .	7
3.3.3	int findCentreColumn(int currentPulse) . . . . .	7
3.3.4	Vector2D findHitCentre(int centreColumn) . . . . .	8
3.4	Electron drift simulation . . . . .	9
3.5	Output file . . . . .	10
<b>4</b>	<b>Results</b>	<b>11</b>
4.1	Photoline Influence . . . . .	11
4.2	B field inhomogeneities . . . . .	11
4.3	Different E fields . . . . .	13
4.4	Different B field values . . . . .	14
<b>5</b>	<b>Geometry note</b>	<b>14</b>
<b>6</b>	<b>Conclusions</b>	<b>17</b>

# 1 Introduction

This report is a dedicated summary of my work during the summer student program 2012 at DESY in Hamburg. I worked in a FLC TPC group and my task was to get familiar with laser calibrations system and develop the photodot reconstruction process. I wouldn't focus here on well known things, I will rather try to mark things which can cause some problems.

## 1.1 Time Projection Chamber

Time projection chamber (TPC) is going to be the main tracker in the International Linear Collider (ILC). It is usually the cylinder filled with gas, with high electric field applied between the endplates.

### Measurement Principle

A charged particle traversing the gas volume of the TPC ionize the atoms of the gas mixture (around 90% noble gas and 10% quencher) along its trajectory. Then released electrons move, due to electric field, to the anode, where they are multiplied and detected. See Figure 1.

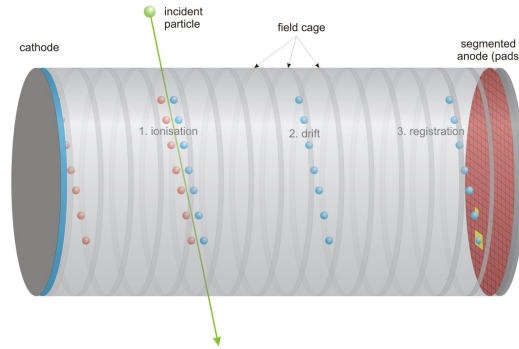


Figure 1: Working principle of a TPC.

## 2 Laser calibration system

### 2.1 Motivations

The main application of laser calibration system is to measure the field distortions inside the drift volume of TPC. But it can be also use to calculate basic gas and drift parameters. Such a system will be also a usefull tool during the data acquisition, because between bunch crossing we can simply have a look at the well known pattern to monitor if everything works fine.

## 2.2 Working principle

Whole laser calibration system consist of cathode with an aluminum pattern and a pulsed light source. In principal we illuminate the cathode using a pulsed laser. The electrons are released from the aluminum pattern and drift through the whole TPC volume to the readout plane on anode. Then we can simply compare the measured image with the pattern at the cathode. See Figure 2.

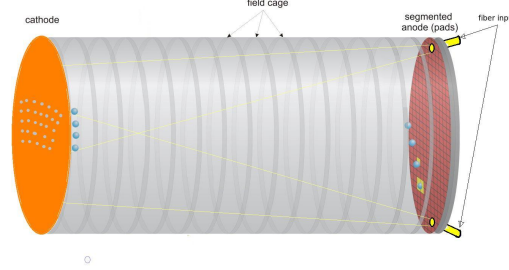


Figure 2: Laser calibration principle.

## 2.3 Setup

Cathode contain seven same patterns with corresponds to seven rodout modules in a Large Prototype. Each of them contain 56 photodots in 8 rows and 7 columns, and one photoline. Photodots are 2.0 mm in diameter. For more details see 1.

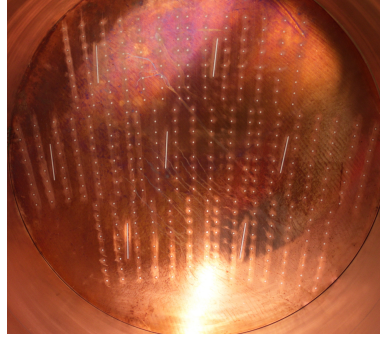


Figure 3: Cathode with photodot pattern.

Laser light must have wavelength with allows for releasing electrons from aluminum but not from copper. So a pules of UV light are generated from an Nd:YAG laser. Then the light is split by a beamsplitter into two equal beams, with travel by the fiber-optic cables to the TPC. Laser system also provides an electronics trigger using a photodiode.

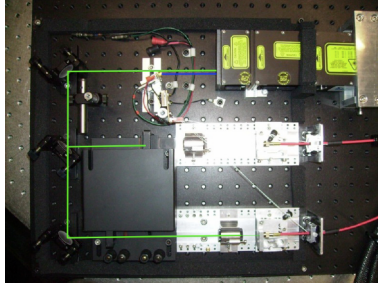


Figure 4: Nd:YAG Laser setup.

## 2.4 Analyzed data

Data analyzed by me were taken in august 2009 with micromegas module without resistive foil and T2K electronics. Measurements were performed with several drift fields (100, 150, 175, 200, 215, 230 [V/cm]), two different B fields (0.5 and 1 [T]). There were also the measurements for different magnetic field inhomogeneities, which were obtained by changing the position of the TPC inside the magnet (Figure 5).

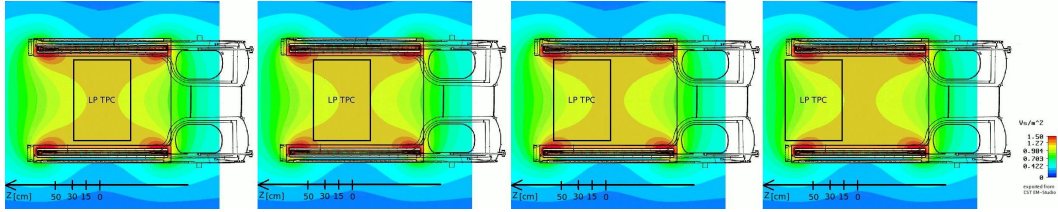


Figure 5: Different TPC positions inside magnet (0, 15, 30, 50 [cm] from the center).

## 3 Photodot reconstruction

### 3.1 Overview

Photodot reconstruction process is based on two MarlinTPC classes which are: PhotodotReconstructionProcessor and ElectronCloudDrifterProcessor. The first one controls whole reconstruction and processes the data, whereas the second one makes a simple simulation of electron drift through the TPC volume.

Photodot reconstruction is divided into four main steps:

1. **findPhotodotPulses 3.3.1** method which uses the simulation to estimate expected arrival time of electrons at the endplate, to cut off all pulses which cannot come from the photo-pattern.

2. **findNearestHit 3.3.2** method which matches the registered image with the particular photodot
3. **findCentreColumn 3.3.3** method which finds out the center column of cluster of pulses that corresponds to one photodot
4. **findHitCentre 3.3.4** method which calculate the center of a cluster

## 3.2 Running a reconstruction

To run a reconstruction you need to create a xml file with the input data in LCIO file and several processors activated. At first using ConditionsProcessor and GlobalField-Processor you can insert the map of the magnetic field. Then if you use RAW data as an input, you need TrackRawDataToDataConverterProcessor to convert them to Data type. Next you use a PulseFinderProcessor in which you can set a pulse selection parameters. Finally you use a PhotodotReconstructionProcessor in which you have to set following parameters:

- **geometr\_filename** - the name of the ASCII file containing the positions of photodots in global TPC polar coordinate system,
- **compared\_photodots\_filename** - the name of the output ASCII file containing the information about photodot position, expected image, and image position.

and some additional parameters mainly for the electron drift simulation:

- **diffusion** - transverse diffusion of electrons along the drift distance,
- **omega\_tau** - the gas mixture parameters for electrons,
- **drift\_velocity** - the average drift velocity (mm/us),
- **tpc\_outer\_radius** - the outer radius of the TPC (mm),
- **tpc\_inner\_radiu** -the inner radius of the TPC (mm) ,
- **outer\_z** - the value of the z coordinate of the readout padplane in magnet coil coordinate system (mm),
- **inner\_z** - the value of the z coordinate of the cathode with aluminum pattern in magnet coil coordinate system (mm),

I have added also a possibility of providing an external pad mapping. If this parameter will be empty a default mapping will be use.

- **pad\_mapping\_filename** - the name of the ASCII file containing: row number, pad number, ordinal number and the padID.

### 3.3 Methods

In this section I will describe in details each steps of the photodot reconstruction.

#### 3.3.1 void findPhotodotPulses()

This method, at first, runs the simulation ( 3.4 ) to get an information how long it takes to drift through the whole TPC volume. It process all 56 dots and remember the lowest and highest arrival time, and also calculate the mean arrival time. Then it sets the window of  $(t_{high} - t_{low} + 50)$ ns around the mean value, and check which pulses falls in this time range. If they don't they are removed.

Alas this function is extremely sensitive to external parameters. It is not hard to provide this information using simulated data, but it could be a problem for a test beam data. To use it you have to have well defined gas properties and drift velocity and set the proper readout frequency. Moreover you have to be sure that the electronic trigger starts at the time when photoelectrons are released.

Unfortunately, I can not understand the time structure of data which I was analyzing, so I had to disable removing pulses. I think that matching the time starting point with the electron releasing time is the thing that you should cry about during tests.

#### 3.3.2 int findNearestHit(Photodot thisPhotodot)

This function takes the structure called Photodot as a parameter. This structure contains an information about photodot position and diameter. The function checks which pulse was registered nearest "thisPhotodot", just comparing the distance between the center of "thisPhotodot" and the center of each pad which had some signal. The ID of the pad with the smallest distance is returned and becomes "currentPulse".

Such algorithm is accurate only if image distortions are less than about 1cm and each photodot left the image. In other case some images are mismatched with photodots or the images are assigned to more than one dot. So some functionality can be added in here to avoid double matching, but the real mismatching appears only for strongly inhomogeneous magnetic fields.

#### 3.3.3 int findCentreColumn(int currentPulse)

This method starts from the "currentPulse" and looks at the existence of the adjacent pad in the row, then checks if the charge on that pad exceeds the charge on the current pad. If it does, such pad becomes "currentPulse", and repeat checks again. If not, it checks if the pad below collect any "currentPulse", if does it moves there and repeat whole procedure, if not ends at current pad. ID of this pad returned and becomes the "centreColumn". See Figure 6.

This method has hardcoded numbers of boundaries pads, which must have been changed for a different module geometry. It also looks for an adjacent pad only inside one module.

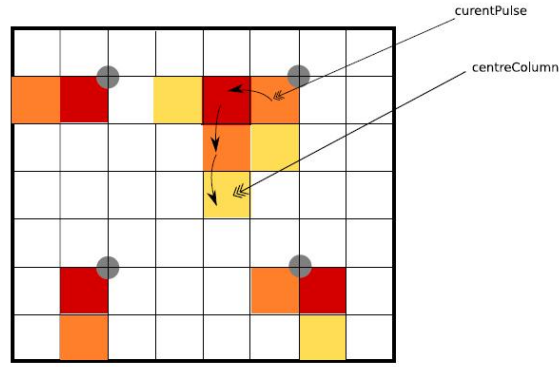


Figure 6: Algorithm of finding “centreColumn”.

### 3.3.4 Vector2D findHitCentre(int centreColumn)

That is the part where the reconstructed position is calculated. It based on the assumption that for each row we have Gaussian charge distribution over the pads, where the width of this distribution depends on the diffusion.

For each row it takes the right edge of the “centreColumn”, then sum up whole charge gathered on the left side from this edge. Next it compare this amount of charge with the expected charge distribution to find out where the mean position was. This is repeated for each row in cluster, and the final x position is the mean value of positions for each row (where most prominent rows are most charged rows).

Such procedure is implement using inverse error function.

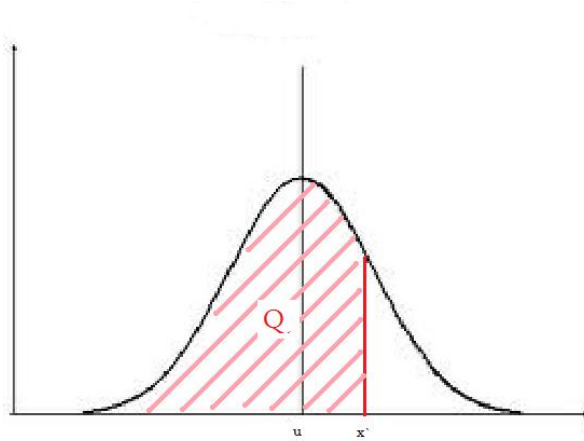


Figure 7: Comparing amount of charge to find out the mean value.



Here are the strict equation which are use in the processor:

$$x = \frac{1}{Q_{total}} \cdot \sum_{n=1}^k [x_{0n} + \frac{1}{2} \cdot x_{width} - \sqrt{2} \cdot D \cdot Erf^{-1}(\frac{Q_{leftn} - Q_{rightn}}{Q_{leftn} + Q_{rightn}})] \cdot (Q_{leftn} + Q_{rightn})$$

$$y = y_0 - \sqrt{2} \cdot D \cdot Erf^{-1}(\frac{Q_{down} - Q_{up}}{Q_{total}})$$

where:

$D$  - diffusion

$x_{width}$  - width of a pad

$x_0, y_0$  - edge position

This algorithm works with an assumption that the charge is spited at least between two rows. Whereas in data which I analyzed it was not like that for the Y coordinate. Then the reconstructed y position is just the center of a row.

### 3.4 Electron drift simulation

Whole electron drift simulation is perform by calling method: `drift_through_cylinder` from `ElectronCloudDrifterProcessor`. Using this function you can set the point of releasing electron, and the redout position. Inside it the iterative simulation (according to 3.4) of movement of electron, in E and B fields, is made using a Class Library for High Energy Physics (CLHEP). It starts from the photodot position and last till the electron reaches the redout plane.

$$\vec{v} = \frac{\mu E}{1 + \omega^2 \tau^2} \cdot [\hat{E} + \omega \tau (\hat{E} \times \hat{B}) + \omega^2 \tau^2 (\hat{E} \cdot \hat{B}) \hat{B}]$$

Where:

$\vec{v}$  - drift velocity

$\mu$  - electron mobility

$\omega \tau$  - gas parameters

I added some functionality inside this function like for expample a possibility to setting a constant E field inside whole TPC volume without providing a field map. I also introduce a shift along Z axis because I'm convinced that magnetic field map which I used is centered at (0,0,150 mm).

Unfortunately the expected position from the simulation do not matches the real image position at all. Now it works only for small inhomogeneities, whereas for a large inhomogeneities, when the effect image rotation appears, it seems that the rotation of expected image is in opposite direction to the real image rotation (see Figure 8 ). I know about some problems with mirrored images and I assumed that it is the problem of mirrored

pattern at the endplate. But now it seems to me that the photodots position on the endplate are right. And this mirror effect appear because padplane mapping is made looking at the module from the inside of the TPC whereas the gear returns the position looking from the outside of the TPC. Of course I am not sure of this but for me it looks like that.

Here you can see well matching images for a small field distortions and opposite rotation in a large field distortions.

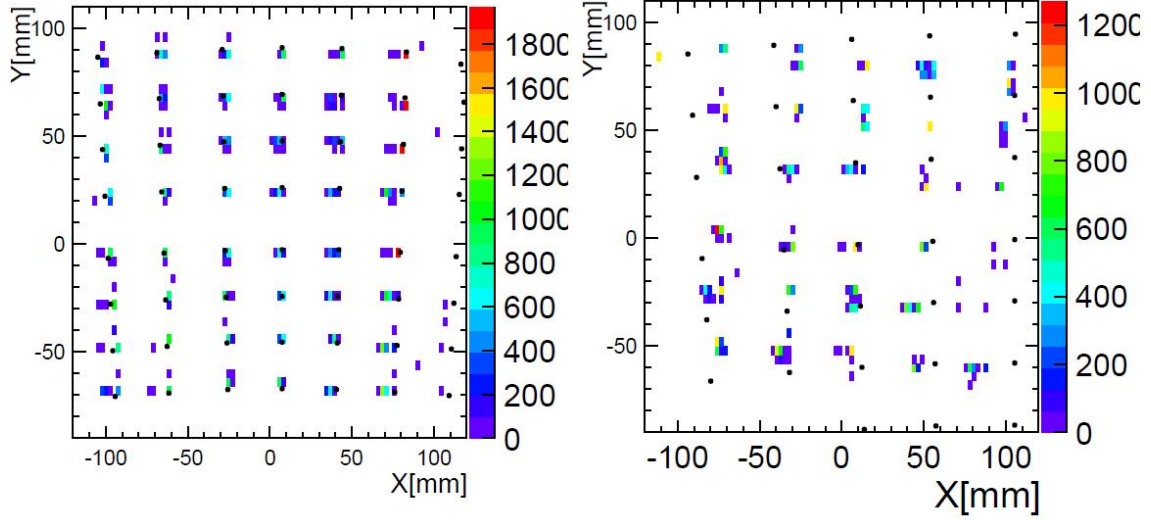


Figure 8: Comparison of expected position(black dots) and real image (colz plot) for Z=15 (left) and Z=50 (right).

### 3.5 Output file

The output file contains information about photodot position, expected image and image position for each photodot. The information about expected image comes from a simulation of the electron drift through TPC, whereas the image position is the reconstructed position.

Here is the example of an output file:

```
# Cartesian coordinates with origin at the center of Large Prototype (x = 0 ; y = 1503.62)
# Photodot position # Expected image # Image position
#( x, y) #( x, y) #( x, y)
103.886 -67.4633 103.304 -67.1752 103.527 -69.3266
72.4265 -65.6485 70.7634 -64.4849 71.9259 -67.3951
38.139 -64.3776 37.1405 -63.0963 39.7891 -66.1457
6.76367 -63.996 6.43815 -62.6235 7.09522 -65.6127
-24.6407 -64.272 -24.2832 -62.8024 -24.3843 -65.8019
-58.9621 -65.3413 -57.8946 -63.8063 -60.4252 -66.8653
.
.
.
```

## 4 Results

### 4.1 Photoline Influence

Now I will show some first results. As we can't see ( Figure 9 - left ) the image contains also some pulses from photoline. To get rid of them I introduce a set of cuts, and on the Figure 9 (right) there is the same image but after cuts. They are implemented after a reconstruction process in my analyse macro.

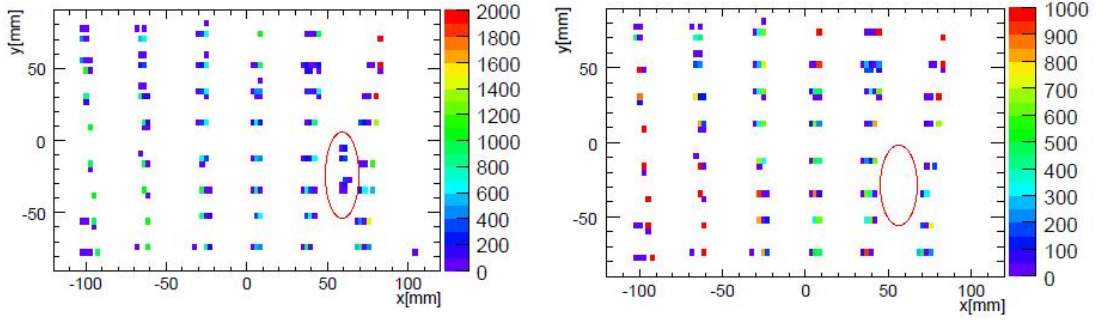


Figure 9: Plots before(left) and after(right) setting a photoline cuts.

### 4.2 B field inhomogeneities

In this section I will show a set of plots which shows how the B field inhomogeneities influence the results. The black dots are the positions of the photodots at the cathode and a plot shows a image for whole run.

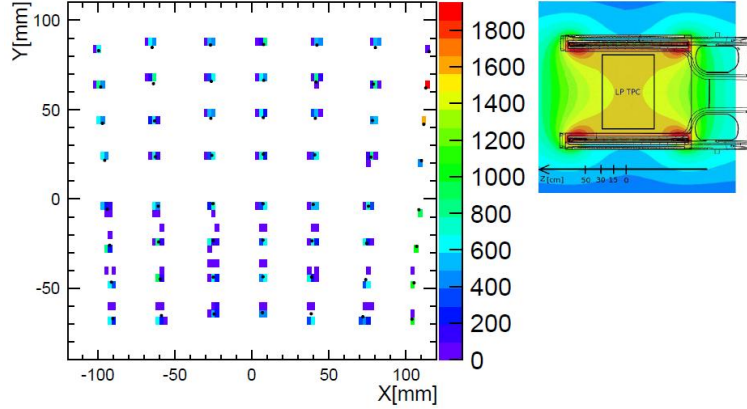


Figure 10: Image at Z=0(left) and TPC position inside magnet (right).

We see that the more inhomogeneous field is, the more image spread outside. And that is a behavior which we should expected. We know that movement of electron is describe

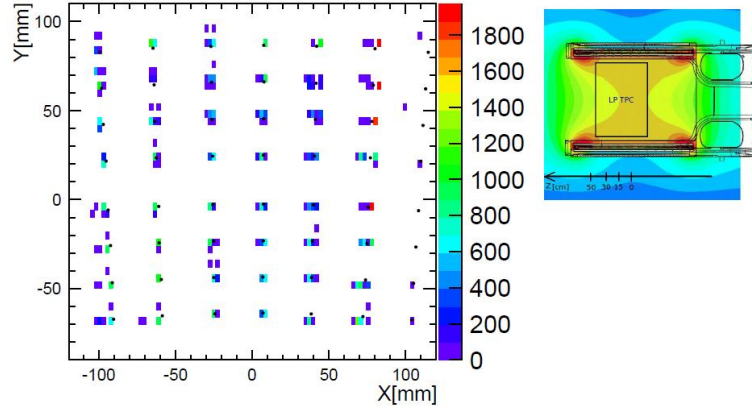


Figure 11: Image at Z=15(left) and TPC position inside magnet (right).

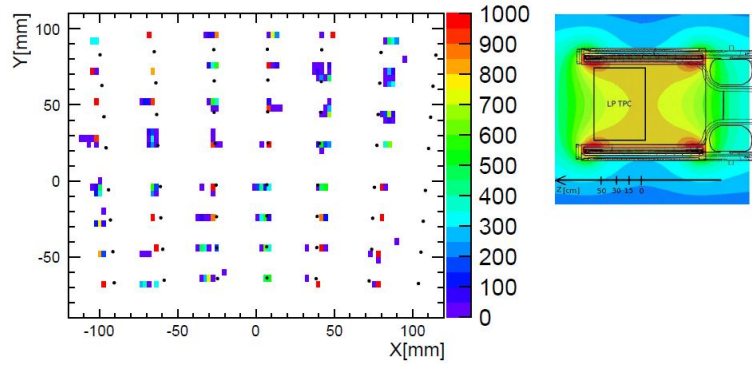


Figure 12: Image at Z=30(left) and TPC position inside magnet (right).

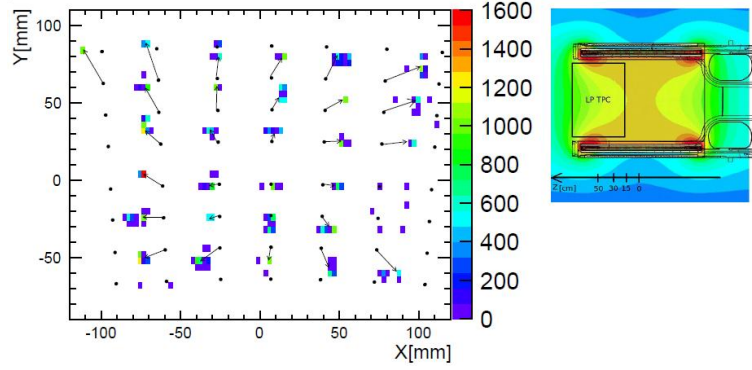


Figure 13: Image at Z=50(left) and TPC position inside magnet (right).

by equation 3.4, where for electrons  $\omega\tau$  is in order of 10 so the last term dominates. So we expect, that electrons follow B field lines and that is exactly what we can see.

### 4.3 Different E fields

These two plots (Figure 14 shows that increasing the drift field value (potential between endplates) don't influence the image even for the high magnetic field inhomogeneities. This is one more the result of high  $\omega\tau$  in formula 3.4.

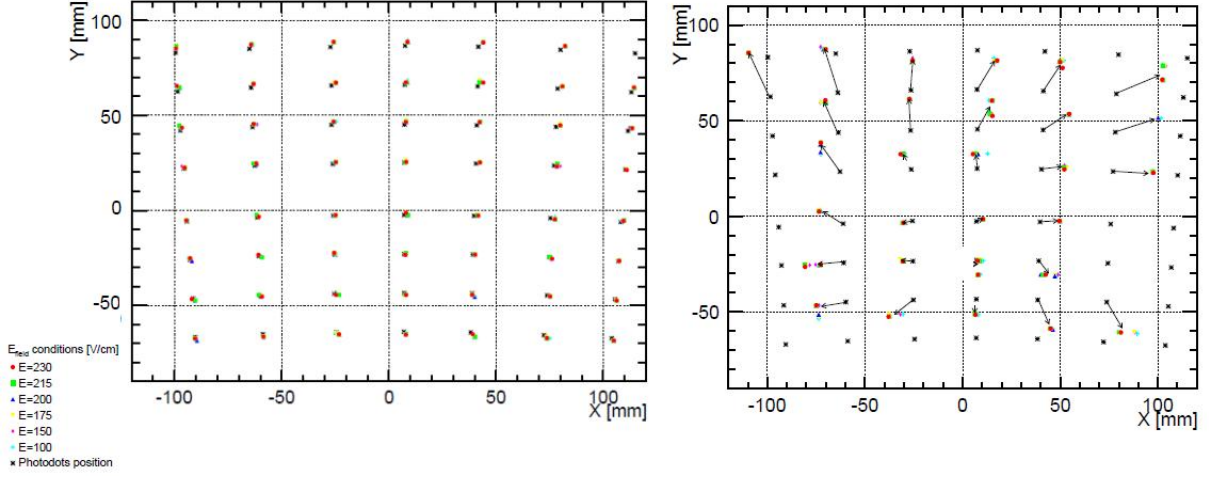


Figure 14: Mean image values for different E fields at Z=0(left) and Z=50 (right).

## 4.4 Different B field values

Measurements provide data for two different B field: 0.5 and 1 T. What we can see(Figure 15 that for a lower field the photodot image is broaden.

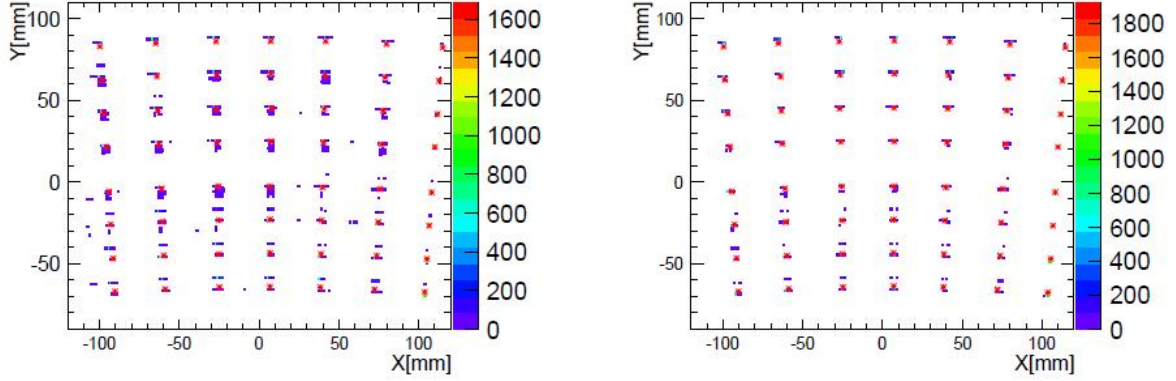


Figure 15: Images for B=0.5T (left) and B=1T (right) at Z=0. (red stars are photodots positions on the cathode)

## 5 Geometry note

while I was trying to adapt the PhotodotReconstructionProcessor to new multi-module standards I have to merge a few different coordinate systems. Here I will try to describe relations between them. To simplify it I will treat them as the 2-dimensional  $(r, \phi)$  or  $(x, y)$  system, because the z coordinate is always centered in the middle of the cylinder.

At first I will define these coordinate systems:

**Global coordinate system** - a polar system of the final TPC with the origin in the center of TPC,

**Magnetic field system** - system with the origin in the center of the magnet coil,

**Large Prototype system** - similar to Magnetic field system, but the origin could be shifted along z axis, by moving LP inside the magnet. Using this Cartesian system output file from the PhotodotReconstructionProcessor is created ,

**Middle module system** - system with the origin in the center (middle row and middle column) of the central module.

In global coordinate system the  $\phi$  axis ( $\phi = 0$ ) is directed to the top. So now this system is well defined and I will use it to describe the origins of the other systems. The polar coordinates of the origin of the magnetic field and the Large Prototype system are : ( $r = 1503.616mm, \phi = 0rad$ ). Whereas the middle module is a little bit shifted ( $r = 1519.755mm, \phi = -0.003055244rad$ ). See Figure 16 where I also put the module ID's from the gear file for LP endplate (gear\_LP\_endplate\_7MICROMEGAS\_modules.xml). Here I will also list the centers of each module:

**No 0.** : ( $r = 1692.255mm, \phi = -0.088401845rad$ ),

**No 1.** : ( $r = 1692.255mm, \phi = 0.05803128rad$ ),

**No 2.** : ( $r = 1519.755mm, \phi = -0.149488369rad$ ),

**No 3.** : ( $r = 1519.755mm, \phi = -0.003055244rad$ ),

**No 4.** : ( $r = 1519.755mm, \phi = 0.143203347rad$ ),

**No 5.** : ( $r = 1348.14mm, \phi = -0.05349526rad$ ),

**No 6.** : ( $r = 1348.145mm, \phi = 0.092937865rad$ ).

I should also mention that the  $\phi$  shift of the central row of the photodots is -0.001643 and this not correspond to the shift of the central module. Moreover, I introduce inside the processor the  $\phi$  shift equals the shift of the central module (-0.003055244). Only after that the photodot matches the image, but I'm quite worried if it is not already taken to account.

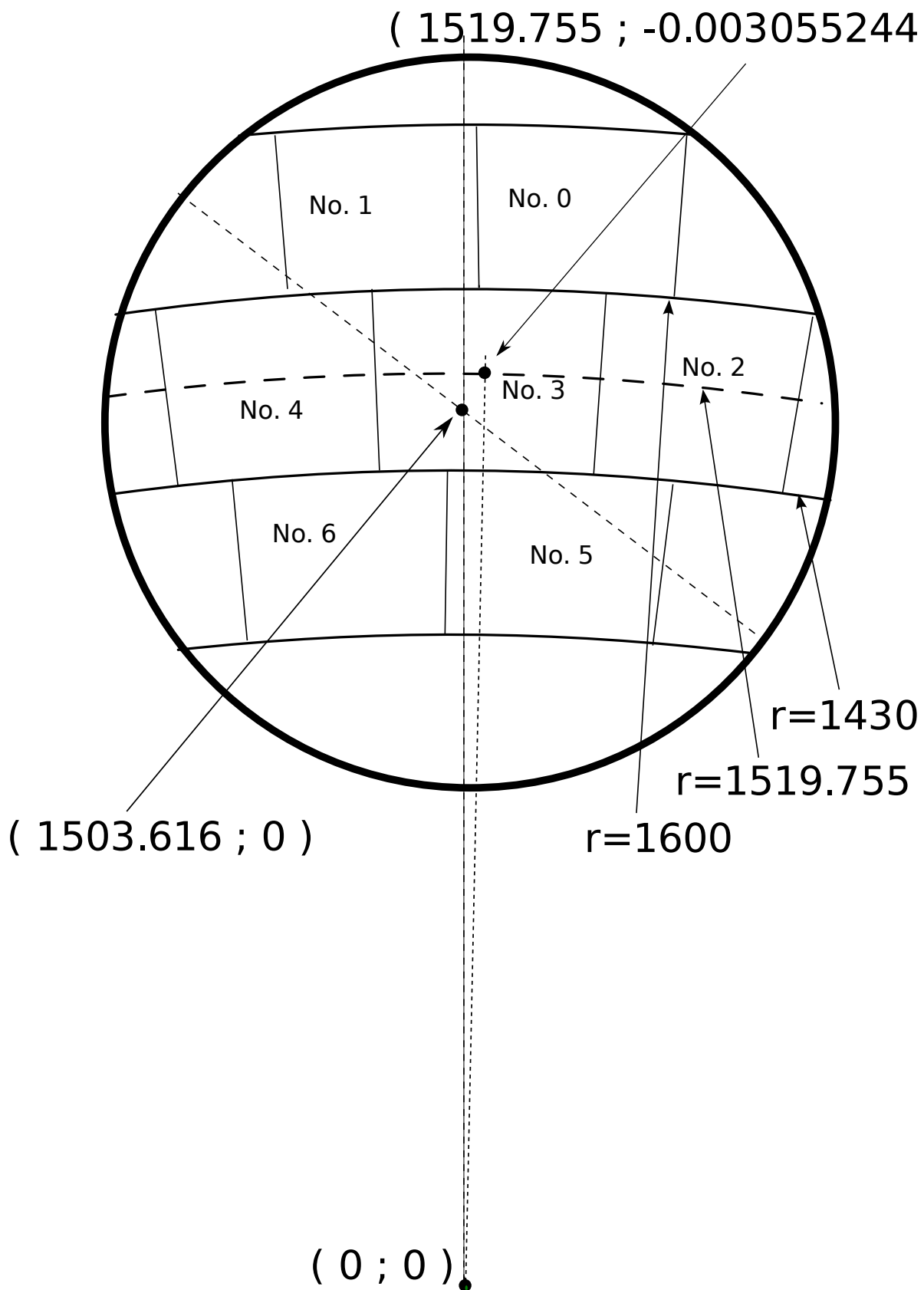


Figure 16: Images for  $B=0.5T$  (left) and  $B=1T$  (right) at  $Z=0$ . (red stars are photodots positions on the cathode)



## 6 Conclusions

During my my stay in DESY I have made a few upgrades to the PhotodontReconstructionProcess which allows to run it with the old data and new multimodule geometry. But it still needs some changes which will allows dealing also with multimodule data. I also added some functionality like possibility of using external pad plane mapping, or setting constant E field without including field map. I have introduce some changes which allows setting the start and end point for the electron drift simulation. Analysis which I perform allows for checking if the reconstruction works fine and can show some general behavior. But still further analysis is needed to get the exact dependence's.

## References

- [1] Aluminum target pattern for LP1 cathode (in attachments) *D. Karlen*