AGH
University of Science and Technology in Krakow

Faculty of Computer Science, Electronics and Telecommunications

DEPARTMENT OF COMPUTER SCIENCE

MASTER OF SCIENCE THESIS

PATRYK ŁAWSKI

VALIDATION OF PROTON RECONSTRUCTION ALGORITHM FOR MULTITRACK PATTERN RECOGNITION IN THE TOTEM EXPERIMENT AT CERN

CERN-THESIS-2018-339
07/12/2018

SUPERVISOR:
Leszek Grzanka Ph.D

Kraków 2018

OŚWIADCZENIE AUTORA PRACY

Uprzedzony o odpowiedzialności karnej na podstawie art. 115 ust. 1 i 2 ustawy z dnia 4 lutego 1994 r. o prawie autorskim i prawach pokrewnych (t.j. Dz.U. z 2006 r. Nr 90, poz. 631 z późn. zm.): „Kto przywłaszcza sobie autorstwo albo wprowadza w błąd co do autorstwa całości lub części cudzego utworu albo artystycznego wykonania, podlega grzywnie, karze ograniczenia wolności albo pozbawienia wolności do lat 3. Tej samej karze podlega, kto rozpowszechnia bez podania nazwiska lub pseudonimu twórcy cudzy utwór w wersji oryginalnej albo w postaci opracowania, artystycznego wykonania albo publicznie zniekształca taki utwór, artystyczne wykonanie, fonogram, wideogram lub nadanie.", a także uprzedzony o odpowiedzialności dyscyplinarnej na podstawie art. 211 ust. 1 ustawy z dnia 27 lipca 2005 r. Prawo o szkolnictwie wyższym (t.j. Dz. U. z 2012 r. poz. 572, z późn. zm.): „Za naruszenie przepisów obowiązujących w uczelni oraz za czyny uchybiające godności studenta student ponosi odpowiedzialność dyscyplinarną przed komisją dyscyplinarną albo przed sądem koleżeńskim samorządu studenckiego, zwanym dalej «sądem koleżeńskim».", oświadczam, że niniejszą pracę dyplomową wykonałem(-am) osobiście, samodzielnie i że nie korzystałem(-am) ze źródeł innych niż wymienione w pracy.

Author's statement

I hereby declare, aware of bearing the criminal responsibility, that this thesis is entirely the result of my own work, as stated in the introduction, except where otherwise indicated by references to other authors.

. . . .

PODPIS

I would like to thank my parents, grandparents and sister for their love and support. Thank you to everyone who helped me during my studies to go ahead and become a better person - my friends from the Scouts of Europe deserve a special place here. I would especially like to thank the supervisor Leszek Grzanka for his time and all advices regarding the following research. I also thank my friends for a wonderful time at CERN, where this thesis had its beginning.

Contents

List of Figures

List of Tables

1. Introduction

The CERN organization (from French Conseil Européen pour la Recherche Nucléaire) is the world's largest center for particle-physics research. In the heart of CERN lies the LHC particle accelerator where several experiments are located. To name a few: ALICE (A Large Ion Collider Experiment), ATLAS (A Toroidal Lhc ApparatuS), CMS (theCompact Muon Solenoid), LHCb (the Large Hadron Collider beauty), LHCf (the Large Hadron Colliderforward) and TOTEM (TOTal Elastic and diffractive cross section Measurement)

The TOTEM experiment is dedicated to forward hadronic phenomena. It measures the total proton-proton cross section, elastic scattering differential cross section in a wide kinematic range and a large spectrum of diffractive processes.

Let us also remark that the elastic scattering – a process, where the protons undergo just a glancing collision and stay intact – is intimately connected with the proton's structure. Thus by studying the elastic scattering, one can learn about the structure of the proton.

TOTEM detectors register particles which collide in interaction point 5. There are such detectors as tracking telescopes - T1 and T2 installed in close proximity of IP5 and a system of movable Roman Pots which are designed to detect leading protons at a few millimetres from the beam centre [26].

Each Roman Pot contains 10 silicon detectors which can register protons by one of 512 silicon strip. These detectors are placed perpendicularly - 5 in U direction and 5 in V direction discussed later. That way it is possible to reconstruct track of single proton passing by RP but when there are 2 and more protons we need a new approach to resolve the ambiguity and correctly perform tracks reconstruction.

1.1. Work objectives

The research objective is to validate a new algorithm to reconstruct tracks of multiple protons passing detector at the same time - later we will call these tracks multitracks - and the phenomena "multitracking".

1.2. Work Content

In this thesis we will devote the second chapter to understand the multitracking reconstruction problem. We will present TOTEM architecture (especially Roman Pots system) and CMSSW software framework which is used by CMS and TOTEM experiment for simulation and reconstruction. Then we will introduce current reconstruction algorithm and its limitations in case of multitracking. Finally we will discuss the proposed multitracking reconstruction algorithm.

In the third chapter we will show methodology used to implement proposed algorithm. At first we will focus on data chosen to serve as an input. These data were gathered during LHC fills: 4499, 4505, 4509, 4510, 4511. They are stored in tree like structures in .root files. We will perform analysis how many events are affected by multitracking problem. Then, we will discuss how algorithm was implemented using pandas, numpy and scipy.optimize python packages.

In the fourth chapter, we will analyze our input data. We will focus on the volume and data structures used by TOTEM experiment and our algorithm. Moreover we will discuss how many single and multitracking tracks may be reconstructed from detector readouts from any of Roman Pot groups.

In the fifth chapter we will present the results produced by algorithm. We will check validity on single and multi track cases. In case of single track we will perform comparison to previously calculated tracks, and in case of multi tracks we will compare expected track distribution with these provided by algorithm.

The sixth chapter is dedicated to results discussion. We will analyze whether results were such as expected, how well tracks were fitted and what could be improved in the future.

2. The TOTEM experiment

The TOTEM experiment aims at a deeper understanding of the proton structure. Physics researches are possible thanks to measurements of total proton-proton cross-section, elastic scattering and diffractive processes at the LHC. Those measurements are based on data provided by movable beam-pipe insertions - so-called Roman Pots. Main task of Roman Pot is to register passing proton by its silicon sensors and then CMMSSW software aims to reconstruct the proton track.

For now it is possible to reconstruct track of the single particle going through the detector at a given time. The objective of this thesis is to prove that reconstruction of multitrack is also possible and valid thanks to the new reconstruction algorithm.

2.1. TOTEM experiment architecture

TOTEM has been built at the LHC accelerator at CERN. It is located at interaction point 5 (IP5) together with CMS experiment.

2.1.1. TOTEM detectors

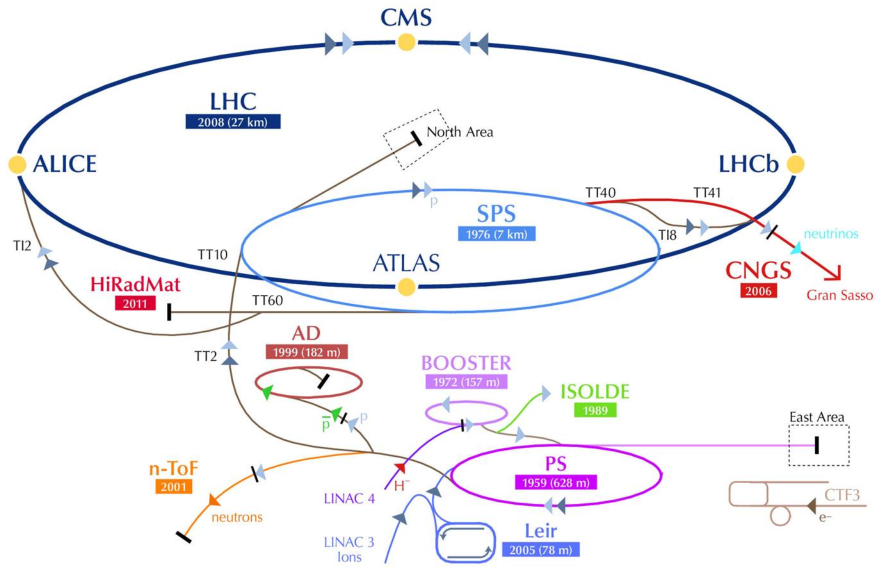

Figure 2.1: TOTEM. Telescope detectors. [19]

In TOTEM there are four telescope detectors - T1 and T2 (accordingly 9m and 13,5m from IP5) as well as 24 so-called Roman Pot detectors (210 - 220m from IP5).

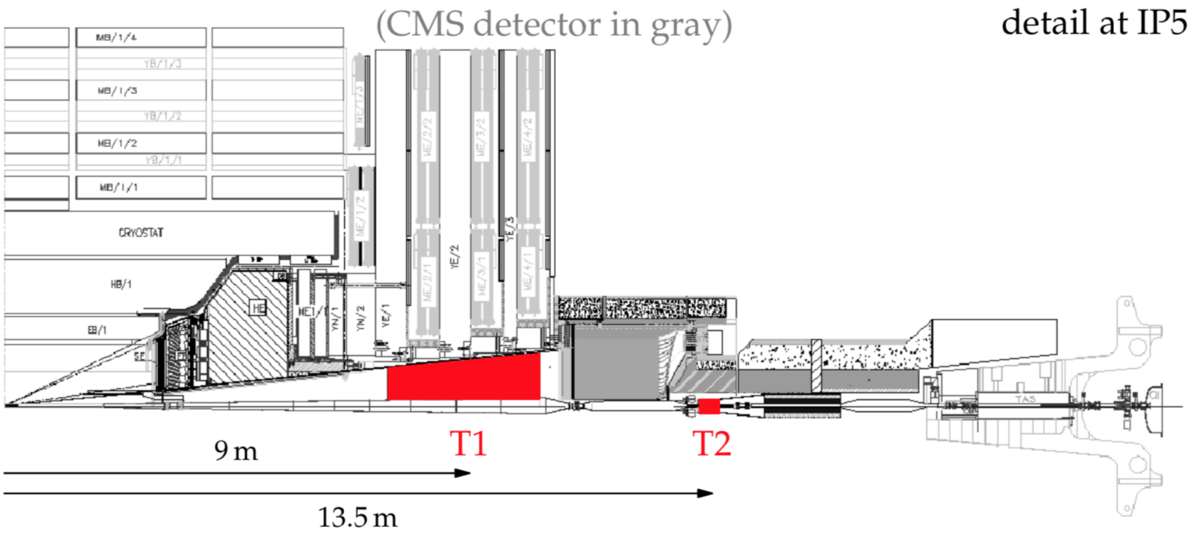

Figure 2.2: TOTEM. Telescope detectors. [21]

In the figure we can see T1 and T2 detectors (marked on red) together with those belonging to CMS experiment. T1 and T2 register particles emerging IP 5 with angles: 18-90 [mrad] and 3-10 [mrad].

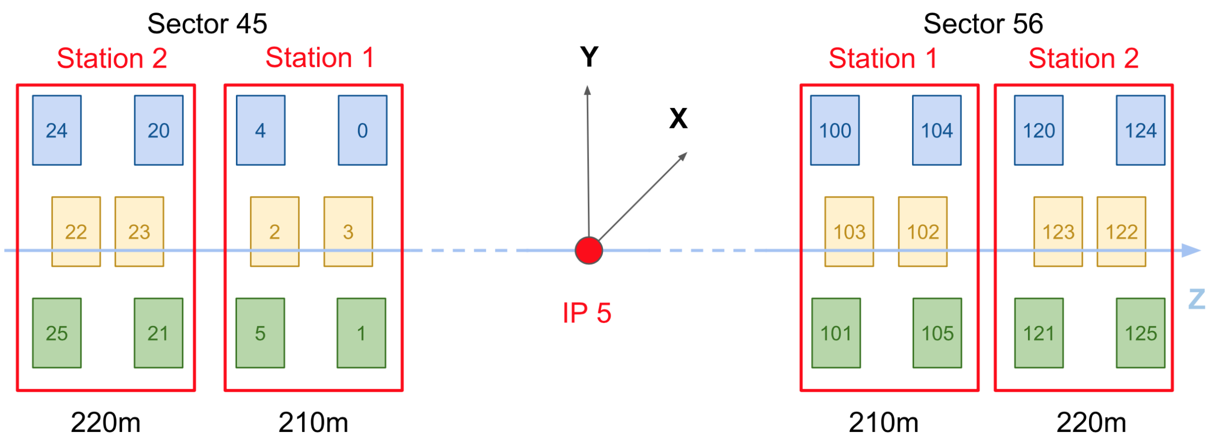

Figure 2.3: TOTEM. The Roman Pot System

The goal of TOTEM is to explore physics of protons emerging from collisions at very small angles (even on the order of tenths of milliradian). This is the reason why Roman Pots are located in stations over 200 meter away from interaction point. On both arms there are 2 stations with 6 Roman Pots. Inside one station there are 4 vertical Roman Pots - 2 top (blue ones), 2 bottom (green ones) and 2 horizontal (yellow ones). As we can see each Roman Pot has its assigned ID.

Each Roman Pot contains 10 silicon detectors with 512 silicon strips in each. They are capable to register (via VFAT controllers) passing proton in its readout direction.

They consists of 512 silicon strips and are oriented in perpendicular U and V direction (as shown below). 5 silicons are oriented in U, and 5 in V direction. The RP sensors are placed at 10 σ + 0,5 [mm] from the beam. [19]

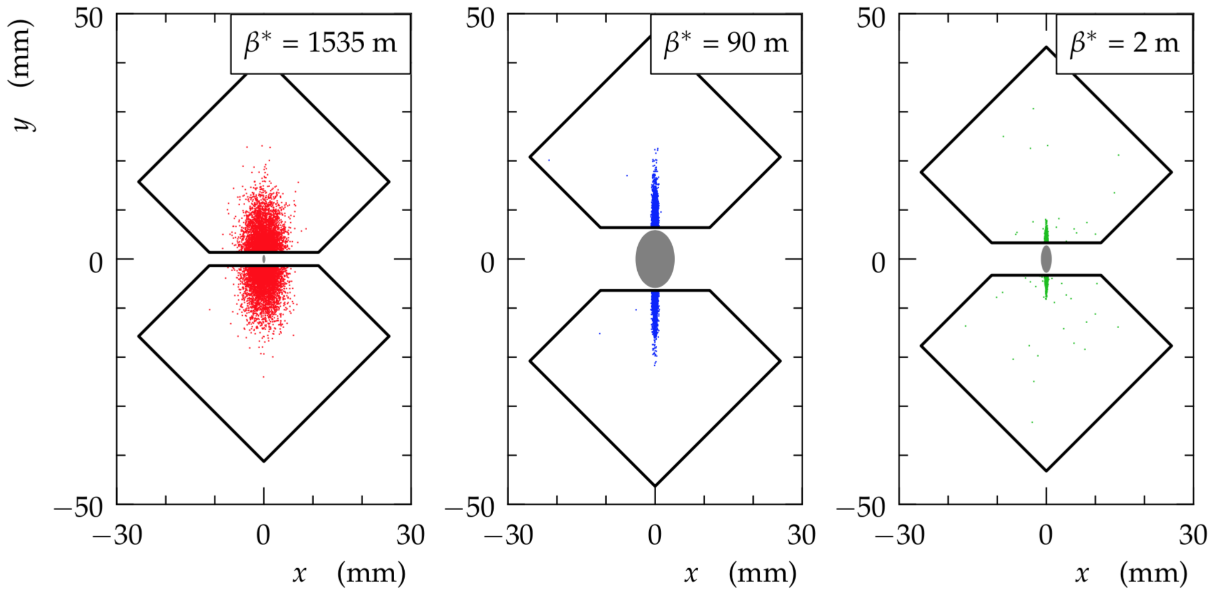

Figure 2.4: Hit distribution depending on β^*[19]

It is worth to mention that between telescope detectors and Roman Pots there are many magnets which keeps the beam inside the beam pipe, and form the beam size. These magnets settings are called the optics of the LHC. Different optics result in different proton tracks and finally different hit distribution registered by RPs.

2.1.2. Coordinate systems

We define xyz (global) coordinate system in the following way. The x axis points outside LHC ring, y axis points upwards (against gravity) and the z axis overlaps with beam axis. Each silicon detector is located on a well defined z coordinate. We have also information about the x and y coordinates of the silicon detector center. Strips in silicon detectors are located in the so-called U and V directions. They correspond to x and y directions rotated by 45 degree. Each silicon has its own readout-direction (it is the perpendicular direction to the strips directions). When the proton hits silicon strip its position is described by the distance (in the readout-direction) between detector's center and the activated strip.

2.2. Framework CMSSW

After detectors register hits we need some software which enable us to perform analysis and tracks reconstruction. Here CMSSW meets our expectations.

2.2.1. CMSSW processing model

CMSSW is a collection of software built around Event Data Model and Services needed by the simulation, calibration and alignment, and reconstruction modules that process event data.

The CMSSW event processing model consists of one executable cmsRun and many configurable modules managed by the Framework. In order to use cmsRun - CMSSW executable - user need to

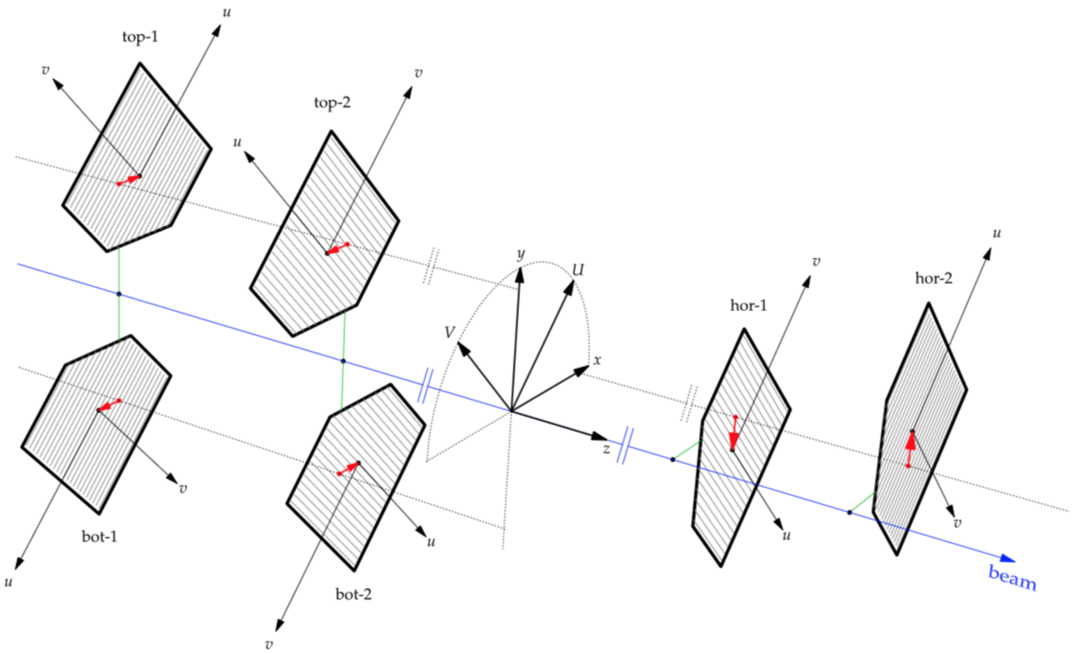

Figure 2.5: Coordinate systems. Source: [19]

provide python configuration file where following things are specified:

- data to use
- which module to execute
- settings for each module
- order of modules execution (path)
- event filters
- output files

Data between modules are passed in Event structure which is a C++ object container. All objects from Event may be individually or collectively stored in ROOT files.

2.2.2. Totem Software architecture

TOTEM uses framework provided by CMSSW with its own modules. Typical flows are shown in the figure 2.6

We can distinguish five main parts of TOTEM software:

- **simulation** - based on Monte Carlo simulations, responsible for generation of physics events, applying smearing effects to simulated bunches (energy, vertex and angular smearing), simulation of transport from IP5 to detectors, and finally simulation of sensor responses
- **reconstruction** - mainly responsible for reconstructing particle tracks and physics based on data provided by sensors

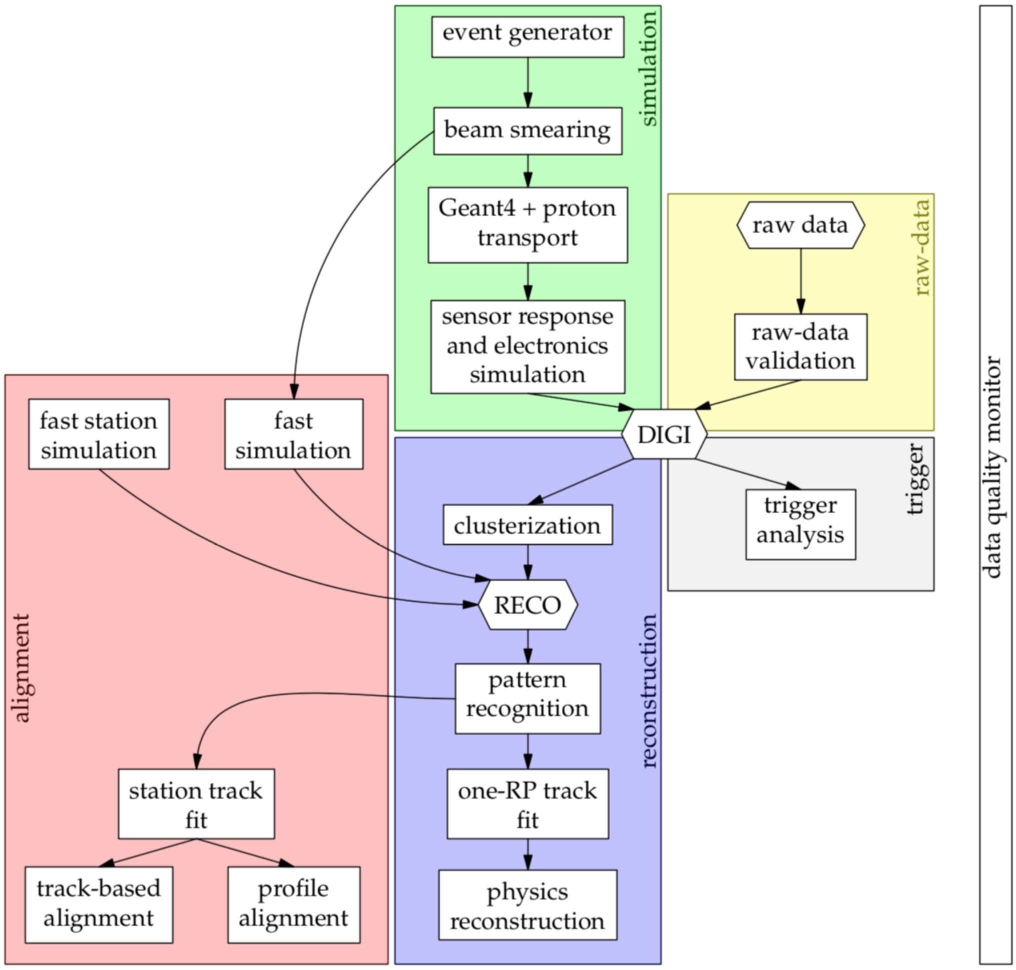

Figure 2.6: TOTEM software overview. [19]

– **alignment** - special reconstruction and analysis with RP alignment applied

– **raw-data** - experimental data preprocessing

– **trigger** - trigger-related analysis.

2.2.3. Reconstruction preprocessing

The objective is to reconstruct proton tracks when multiple particles hit sensor at the same time. Thus the algorithm will take as an input the product of pattern recognition. Let us explore in more details with what kind of data will we begin our reconstruction process.

After detectors are hit, data acquisition system (DAQ) stores VFAT output in **raw data** type. Next these data are validated and - if valid - stored in **DIGI** format. It may happen that several following strips are activated during the hit, then we need to count those strips as a one hit. This is the main idea of **clusterization** after which the **RECO** data type is being created. Inside RECO we will find position of each cluster expressed as distance from the center of the sensor. The next step is to recognize which hits (inside same RP, from silicon detectors with same read-out direction) may form a straight line in Uz or

Vz coordinate system described in 2.1.2. With those "u" and "v" lines, and associated hits we are ready to perform multi track reconstruction.

2.3. State of the art

In this section we would like to examine the nowadays approach to track reconstruction and its limitations when facing multi track problem. Currently reconstruction is done per Roman Pot. Pattern recognition algorithm provide lines in Uz and Vz planes. Those are projections of the reconstructed track.

2.3.1. Pattern recognition

As mentioned in 2.2.3 we will begin with pattern recognition as this step provides data on which we will run our reconstruction algorithm.The aim of pattern recognition is to group reconstructed points of the detectors of a given Roman Pot that belong to the same proton track. [26] As the RP contains 5 pairs of silicon detectors in two readout direction u and v, the search can be performed independently for both directions.

Currently used pattern recognition algorithm is based on an optimization of Hough transformation pattern search. [19]

We can parametrize the line pattern by a slope a and an intercept b

$$q = a(z - z_0) + b,$$

where z_0 refers to the center of the RP. The aim of Hough transform is to assign each point (z, q) (q stands for u or v) a line in the ab space according to following equation:

$$(z, q) \longrightarrow b = -(z - z_0)a + q.$$

The Hough transform is visualized in the figure 2.5. There we can observe that colored points (which are supposed to form a line) are transformed to lines which intersect in a close area. Meanwhile the black point (probable noise) is transformed to a line which does not intersects with others in close proximity. Applying Hough transform reduces our problem to an cluster search.

Cluster search algorithm consists of four major steps.

1. **Assign weights.** We assign each hit a weight $w = \sigma_0/\sigma$, where σ is the uncertainty of the hit and $\sigma_0 = 66/\sqrt{12}\mu m$ is the uncertainty of a 1-strip cluster. The weight of the line crossing is the sum of weights of corresponding hits.

2. **Build clusters.** To build a cluster we iterate over crossings and try to add them to existing clusters. We assume that the cluster position is expressed with a_c and b_c. Where a_c is a weighted sum over all contributing crossings: $a_c = \Sigma a_i w_i / \Sigma w_i$. The b_c is counted accordingly. Given a and b - the coordinates of crossing, we perform following tests:

$$|a - a_c| < clusterSize_a/2$$

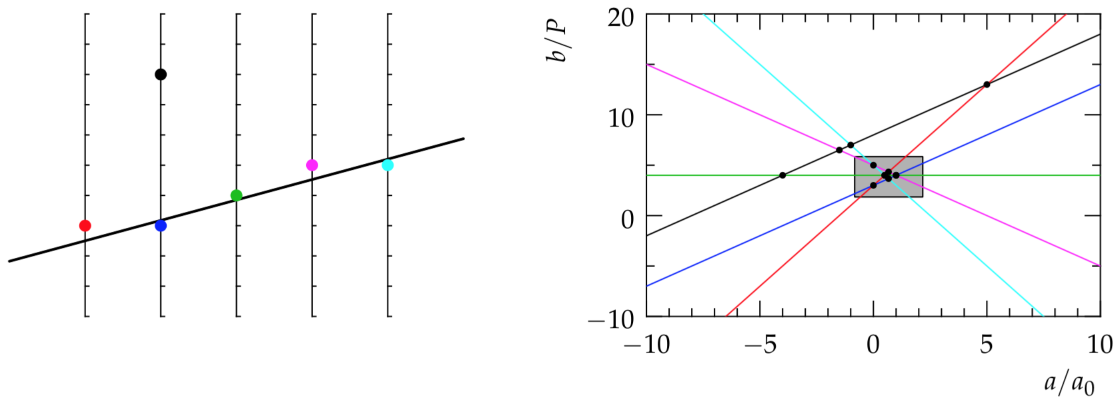

Figure 2.7: Hough transform visualization. On the left we can see five lines representing 5 detectors oriented in same direction. The horizontal ticks represent silicon strips. Dots represent hits. On the right we can see Hough transform. The P unit (pitch) is equal to distance between two strips. The a_0 is equal to P/d where d is the distance between two silicon detectors. [19]

$$|b - b_c| < clusterSize_b/2$$

If both are passed we add crossing to cluster, otherwise a new cluster is created.

3. **Compute clusters weight.** For each cluster we calculate a weight as a sum of the weights of the hits involved in the crossing.

4. **Determine valid clusters.** Take the cluster with highest weight. If the weight is less than $threshold$, stop. Otherwise treat cluster as a recognized pattern and store associated hits.

The constants used in described algorithm were determined empirically and set to following values:

$$clusterSize_a = 3a_0$$

$$clusterSize_b = 4P$$

$$maxHitsPerPlaneToSearch = 4$$

$$minPlanesPerProjectionToSearch = 3$$

$$threshold = 3$$

After the lines patterns are recognized we are ready to move to the reconstruction algorithm.

2.3.2. Reconstruction of elastic events

TOTEM, as the name indicates (TOTal Elastic and diffractive cross section Measurement) run researches on elastic and diffractive interactions. We will describe **only** reconstruction of the elastic events. This reconstruction is based on linear track fit.

In [2.3.1] we described u, v lines pattern recognition which easily leads us to track fit on one RP-level. Unfortunately due to limited RP thickness (3 cm) and little angle of the track (with respect to the beam) only hit position (x and y) may be reconstructed with acceptable precision. At a distance of 3 centimeters track deviation is negligible. Thus silicon strips - distant from each other by $66 \mu m$ - cannot properly measure v_x and v_y angles.

Having fitted tracks per each RP, the global track reconstruction algorithm comprises with three steps:

1. **Hit selection.** The objective of this phase is to select only hits associated with the actual track. That is to suppress background and tracks from secondary interactions. We calculate $\zeta(s)/L(s)$ for each hit and apply the road search algorithm [18]. Then we choose the road with the highest weight and proceed only if there is at least one hit on both sides.

2. **Fitting.** In this phase we perform three fittings: left-arm fit, right-arm fit and global fit with corresponding hits.

3. **Cut application.** In the last phase we want to distinguish elastic scattering events by exploiting symmetry of angles and positions on both arms. We investigate left-right differences and apply cut on vertex and angular tolerances.

This way we have reconstructed proton tracks from elastic scattering events.

2.3.3. Multitracking problem

Having recognized one u and one v line we have two projections of track line in perpendicular planes. Thus it is easy to reconstruct the line of a track using basic algebra. Let us ask a question: **What happens when more than one u and v lines are recognized?** Such situation leads us to an ambiguity, hence we do not know which u projection matches v projection. This problem is referred as multitracking problem, as there are many particles passing passing RP at a same time, and there exist more than one possible hit point. The pictures below will help us to visualize the problem.

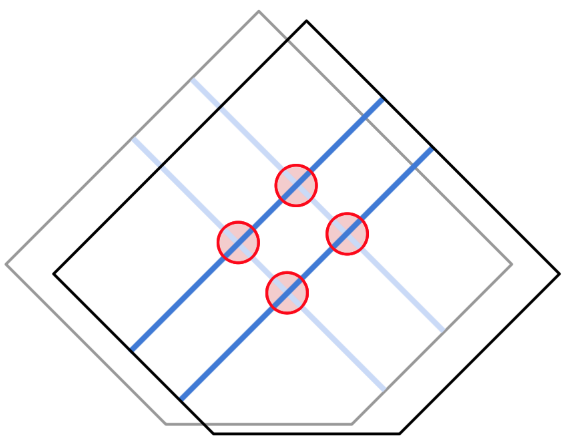

Figure 2.8: Multitracking problem in one Roman Pot. Two silicon detectors register two hits each. Possible tracks may lead through each of diagonal pairs of points.

Traditional reconstruction algorithm can be used only when no ambiguity occurs. The multitracking phenomena requires new approach which will be discussed in details in section 2.4.

2.4. Multitracking Algorithm

As far we got to know the TOTEM architecture, the currently used reconstruction algorithm and we were introduced to multitracking problem. Let us investigate further this phenomena.

2.4.1. Problem description

We have seen that many u and v lines recognized on one RP-level results in the ambiguity. This is not the only challenge we have to face during multitracking reconstruction. Situation is similar when we consider reconstruction between Roman Pots. Following picture will introduce the so-called "ghost tracks".

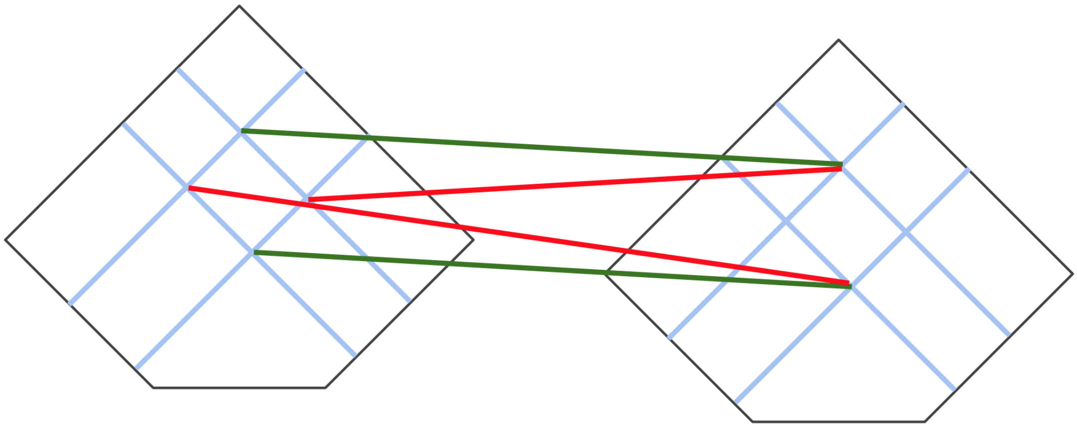

Figure 2.9: Multitracking in two Roman Pots - ghost tracks. The green lines represent real tracks, whereas the red lines show *ghost tracks*.

The *ghost tracks* are the track that satisfy detectors readouts but do not show the real way of proton. In order to reject false tracks we need an additional validation. The solution was introduced in Totem upgrade proposal.

According to technical report [9] one station (more precisely far unit of first station) is rotated by 8 degrees. The enhancement enables us to validate tracks and reject the ghost ones. Pictures below show the enhancement and its effect.

Rotated unit results in different orientation of silicon detectors and hence we are able to perform validation as shown below.

The ghost tracks, although satisfy hits in two Roman Pots, fail to satisfy the hits from tilted one. We will use that fact in the last phase of proposed multitracking algorithm.

2.4.2. Algorithm description

The aim of this algorithm is to provide track line as a set of following parameters:

$$(x, y, z, dx, dy, dz)$$

Where:

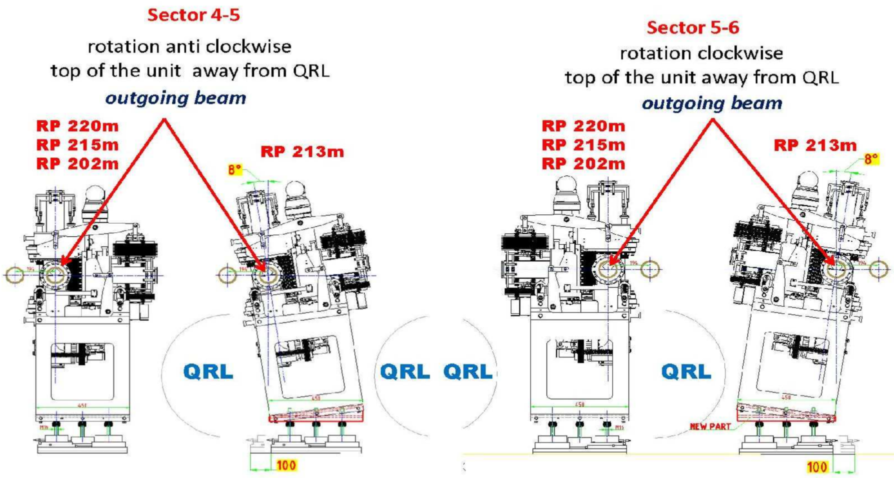

Figure 2.10: Drawings of the rotated RP units. The view is directed along the outgoing beam. [9]

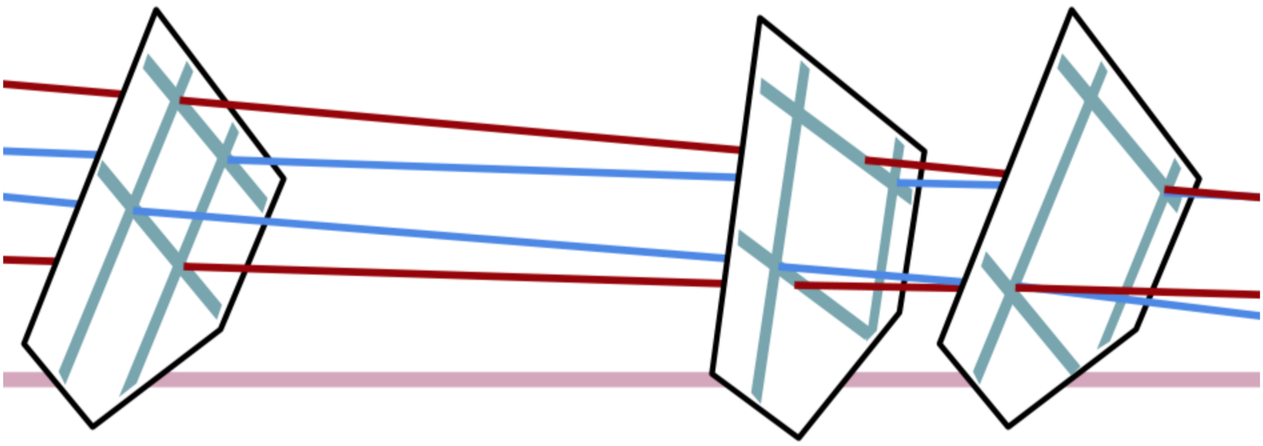

Figure 2.11: Reconstruction with rotated silicon detectors.

– x, y, z are the coordinates of the track's starting point in the global coordination system

– dx, dy, dz are the components of the unit vector which indicates the track direction.

Algorithm will take as an input the recognized u and v lines as well as the associated hit positions. Before we move on with algorithm's steps description we need to define following terminology:

– **Event** - collision of proton bunches in IP5 and associated detector readouts

– **Roman Pot group** - set of Roman Pots located on the same arm (left or right) and in the same orientation (horizontal, top or bottom). There are 6 groups in total.

– **UV combination** - pair of u and v lines from the same Roman Pot in the given event

– **Fishing Roman Pot** - the Roman Pot with the most number of *UV combinations* in a given event and Roman Pot group

– **Fitting Roman Pots** - all Roman Pots from Roman Pot group except Fishing Roman Pot.

Data preparation

In order to start algorithm we need to divide our data set on chunks. In each chunk there are u and v lines from specified event and Roman Pot group. Then we can distinguish three main algorithm phases:

1. **Combinatorics**
2. **Fitting**
3. **Fishing**

The following picture will help us understand algorithm flow, its input and output.

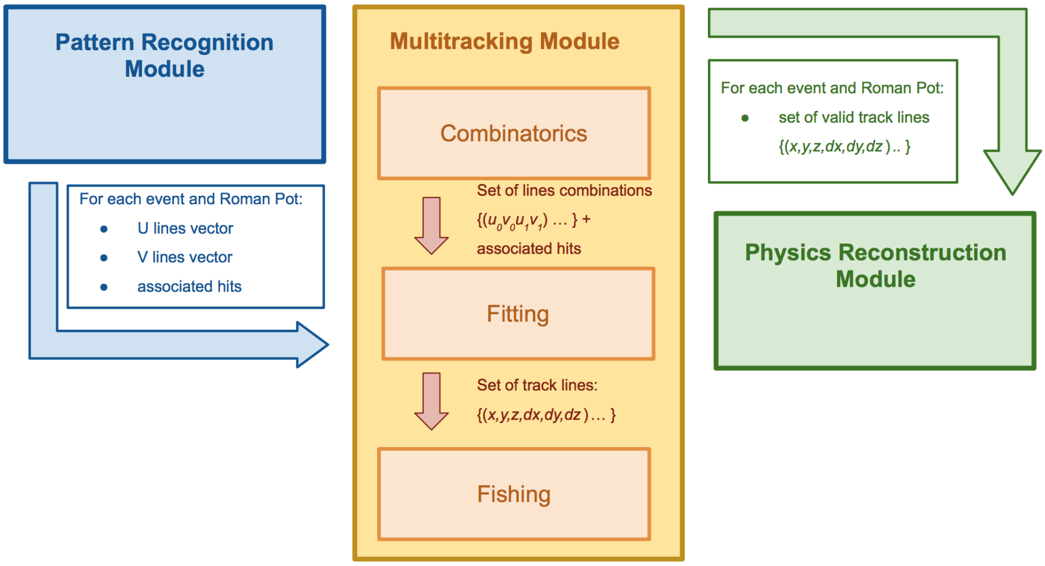

Figure 2.12: Multitracking Module data flow

Combinatorics phase

This phase has two steps and affects only data from fitting Roman Pots. In first step we work only on one Roman Pot level. In each Roman Pot we combine all u and v lines to create UV combinations (every possible pair of u and v). The auxiliary drawing is shown below in Figure 2.11.

After the one RP level combinations are ready we start to combine them between Fitting Roman Pots.

Let us count the total number of combinations. If we assume that:

- U_0 - number of u lines in first of Fitting Roman Pot, assume $U_0 = 2$
- V_0 - number of v lines in first of Fitting Roman Pot, assume $V_0 = 2$
- U_1 - number of u lines in second of Fitting Roman Pot, assume $U_1 = 2$
- V_1 - number of v lines in second Roman Pot, assume $V_1 = 2$

The number of all combinations (N_c) is expressed with equation:

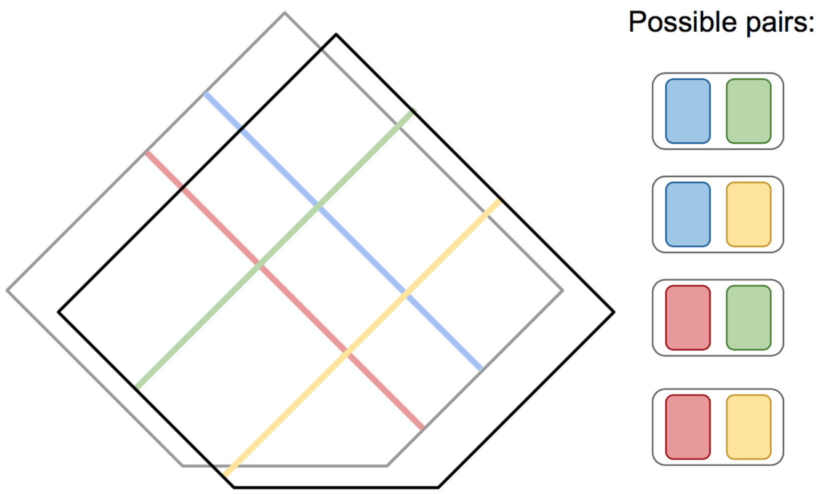

Figure 2.13: Combinatorics phase inside one Roman Pot

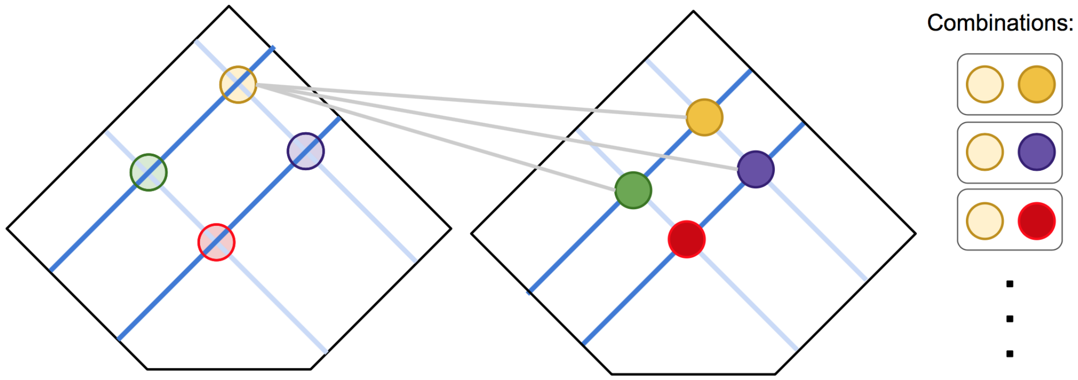

Figure 2.14: Combinatorics phase between Fitting Roman Pots

$$N_c = U_0 * V_0 * U_1 * V_1$$

With our assumption there would be:

$$2 * 2 * 2 * 2 = 16$$

combinations and it is only the most simple example of multitracking problem. The output of this phase is set of (u_0, v_0, u_1, v_1) combinations.

Fitting phase

Having determined UV combinations it is time for fitting phase. The aim of which is to determine a track line. The figure 2.13 depict the process. We want to represent each hit - related to any of u or v line from combination - as a line in global coordinate system, which would represent the hit silicon strip. Then we want to find a best fit line for those silicon strip lines.

The goal is achieved by least square method which simply minimizes the sum of squared residuals (differences between silicon strip lines and a fitted track line).

Fishing phase

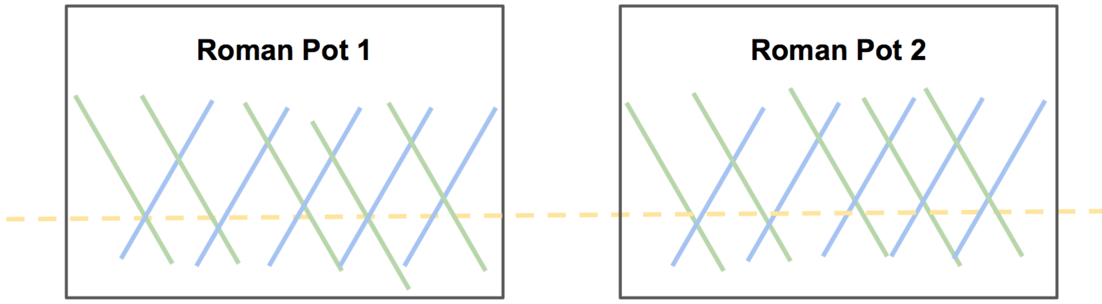

Figure 2.15: Fitting Phase. The solid green lines represent hit strips in u direction and solid blue lines - hit strips in v direction. The dashed yellow line represent fitted track line

In this phase we want to validate tracks obtained by fitting. To do so we use hits related to u and v lines from fishing Roman Pot. For each track we measure the weight which says how good does it fit this third Roman Pot. Tracks with lowest weights are considered as the real ones and passed further as the multitracking module output.

We have described multitracking problem, discussed connected experiment architecture update and outlined algorithm concept. In the following chapters we will study implementation process.

3. Methodology

In order to prove the correctness of the multitracking algorithm, we have implemented it and performed the validation process using data collected from the TOTEM experiment detectors in October 2015. Due to the size of the data, we were forced to parallel our computations. For this purpose, we used the Prometheus supercomputer from Academic Computer Center Cyfronet AGH in Cracow.

During the implementation, we used such tools as SWAN and CMSSW 2.2. We have also utilized python libraries for big data analysis, processing and visualization such as numpy, pandas and matplotlib.

We paid particular attention to the following problems:

– scalability of the algorithm with respect to the data size

– transformation of tree structured data provided by CMSSW framework into the records used by our algorithm

– reconstruction of the correct geometry of the detectors.

3.1. Data to analyze

The analyzed data origin from data taking process performed in October of 2015. During such data taking there are many *fills* where bunches of protons are injected into LHC and accelerated to almost the speed of light. Each fill is assigned with id. Our data where gathered during fills with following ids: 4499, 4505, 4509, 4510, 4511. At the time of those fills the optics of LHC was set to $\beta* = 90m$

It is worth emphasizing the size of the input data, namely:

– 6230 .root files with a total size of 6,59 TB

– Each file contains data from $2 * 10^5$ events

– In total, there were $1, 2 * 10^{10}$ recorded events.

To store such a big amount of data the EOS [1] (open source distributed filesystem (DFS) developed at CERN) was used. This file system is accessible from LXPLUS - a CERN cluster of PCs running Linux.

In order to investigate what C++ data types are stored in .root files we used a CMMSW tool named *edmDumpEventContent*. Results are shown below.

This tool provided us with information about data type as well as the origin module and process which produced each data.

```
Type                                    Module                        Label              Process
--------------------------------------------------------------------------------------------------------
RPFittedTrackCollection                 "RPSingleTrackCandCollFit"    ""                 "rpReconstruction"
RPRecognizedPatternsCollection          "NonParallelTrackFinder"      ""                 "rpReconstruction"
RPReconstructedProtonCollection         "RP220Reconst"                ""                 "rpReconstruction"
RPTrackCandidateCollection              "NonParallelTrackFinder"      ""                 "rpReconstruction"
Totem::RawEvent                         "source"                      ""                 "rpReconstruction"
TotemDigiCollection<T1DetId,T1DigiVfat>     "Raw2DigiProducer"            "t1DataOutput"     "rpReconstruction"
TotemDigiCollection<T1DetId,T1DigiWire>     "Raw2DigiProducer"            "t1DataOutput"     "rpReconstruction"
TotemDigiCollection<T2DetId,T2DigiVfat>     "Raw2DigiProducer"            "T2Digivfat"       "rpReconstruction"
TotemDigiCollection<T2DetId,T2PadDigi>    "Raw2DigiProducer"            "T2PadDigi"        "rpReconstruction"
TotemDigiCollection<T2DetId,T2StripDigi>    "Raw2DigiProducer"            "T2StripDigi"      "rpReconstruction"
edm::DetSetVector<RPDigCluster>         "RPClustProd"                 ""                 "rpReconstruction"
edm::DetSetVector<RPRecoHit>            "RPHecoHitProd"               ""                 "rpReconstruction"
edm::DetSetVector<RPStripDigi>          "Raw2DigiProducer"            "rpDataOutput"     "rpReconstruction"
edm::TriggerResults                     "TriggerResults"              ""                 "rpReconstruction"
map<Totem::FramePosition,VFATStatus>      "Raw2DigiProducer"            "conversionStatus"   "rpReconstruction"
map<unsigned int,unsigned int>          "Raw2DigiProducer"            "t2DataOutput"     "rpReconstruction"
vector<RPCCBits>                        "Raw2DigiProducer"            "rpCCOutput"       "rpReconstruction"
```

Figure 3.1: Dumping .root file content

```
struct Line {
  typedef std::vector<RPRecoHit> HitCollection;

  double a;                        ///< slope in rad
  double b;                        ///< intercept in mm
  double w;                        ///< weight
  HitCollection hits;              ///< corresponding hits

  Line(double _a = 0., double _b = 0., double _w = 1.) : a(_a), b(_b), w(_w) {}
};

std::vector<Line> uLines, vLines; ///< the recognized lines
```

Figure 3.2: RPRecognizedPatterns data structure

The type of our interest was RPRecognizedPatternsCollection, where for each Roman Pot in a given event we could find vectors of u and v lines together with associated hits. Each RPRecotHit contains detector ID, position (represented by one *double* with value of hit distance from the silicon detector center) and its uncertainty.

3.2. Geometry of detectors

Silicon detectors become damaged over the time. Of course some the most exposed to the beam radiation parts become damaged more quickly. If they become unable to register particles we may shift detector inside Roman Pot and use unaffected regions to continue readouts. Thus during each fill detectors may have different alignment. We need to remember about detectors placement in reconstruction algorithm.

Silicon geometry is described in XML files based on Detector Description Language (DDL) [8]. We can distinguish two kinds of geometry:

- **ideal** as declared in a design phase
- **real** it is the ideal geometry + alignment corrections. Alignment takes into account such things like metrology and motors. Also remember that Roman Pots are movable devices.

There are separate XML files with detector geometry for each fill. We will access geometry information via *Geometry/TotemRPGeometryBuilder*.

3.3. CMSSW

The data origins from CMSSW framework. To access it we implemented module - the EDAnalyzer - which took as an input the .root file. Then we processed event by event with use of a handle - namely *Handle<RPRecognizedPatternsCollection>* - by which we were able to access vector of u and v of each Roman Pot and store related hits to a flat .root file. We used that file as an input for our python multitracking algorithm.

3.4. SWAN and Python Algorithm

Each lxplus user has access to SWAN, where one can create jupyter notebooks and use installed Python libraries. We have used the following libraries:

- **root_numpy** to read content of .root files
- **pandas** to create dataframes
- **scipy.optimize** to perform algorithm's fitting phase

We have created dataframes from .root data, based on which we have analyzed how many events are affected by multitracking.

In order to begin reconstruction we need to know the exact location of each of the silicon sensors. This kind of data we got from *Geometry/TotemRPGeometryBuilder* module. Configuration of this module required .xml file with distances from the beam of each Roman Pot as well as file containing alignment of every silicon detector.

During implementation we have tried a new approach, reconstructing proton tracks based on 3RP data at once. Roman Pots were divided into the following groups:

- **top**
- **bottom**
- **horizontal**.

To make a proof of concept we have determined a tracks that are a simple lines in 3D space. The lines were determined using the scipy python library with the minimization function **optimize.minimize** We were trying to minimize the chi squared function based on sum of distances from hit lines to the track:

$$\chi^2(x_0, y_0, z_0, v_x, v_y, v_z) = \sum_{i=1}^{n} = \frac{d^2(l_0, l_i)}{e^2(l_i)}$$

We consider track as a well fitted when *normalized* χ^2 function is lower than 1, that is:

$$\frac{\chi^2}{n - n_{dof}} < 1$$

Where:

– n - number of hit lines taken to fitting

– n_{dof} - number of degrees of freedom - basically it is the number of parameters of χ^2 function

This comply with intuition - the more hits taken to fitting, the better results we should get.

The results were satisfactory (the tracks passed through the detectors, with the accuracy of the distance between the strips). However, due to the amount of data, and the performance of the original reconstruction algorithm, the time (> 1s) to determine a single track was unacceptable.

Then we tried to find tracks using *scipy.optimize.least_squares* . The results were obtained much faster - one order of magnitude smaller execution time per track.

3.4.1. Pseudocode

The psuedocode presented in the following page give us an insight what happens during each of algorithm phases. At the beginning we load our data to pandas dataframe. We will perform reconstruction process within distinct pairs of eventIds and groupIds if and only if there will be hit registered in all of three Roman Pots from a given group. Then we are choosing the fitting Roman Pots and the fishing one. For fitting Roman Pots we set the all possible combinations of *u* and *v* lines in the combinatorics phase.

During fitting phase for each combination we compute track line using python scipy.optimize library.

After the all track lines are set. We perform fishing phase during which we reject the false pairs of tracks and choose the only one which is correct.

3.4.2. Code structure

In this subsection we will discuss the structure of the implemented software.

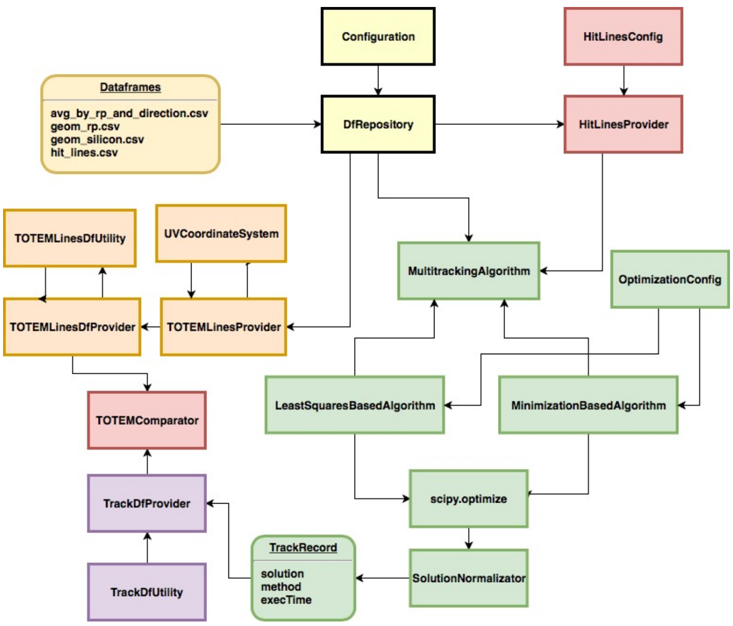

Figure 3.3: Code structure overview

Algorithm 1 Multitracking Algorithm

```
hits_df ← pandas.Dataframe(eventId, groupId, rpId, u_line_no, v_line_no
                           siliconId, direction, line_no, position)
event_group_distinct ← distinct(pair(eventId, groupId)) in hits_df
for all event_group_pair in event_group_distinct do
    if distinct(rpId) < 3 then
        continue
    end if
    fishing_rp ← rpId with max(u_line_no ∗ v_line_no)
    fitting_rps ← distinct(rpId)\{fishing_rp}
    RP_a, RP_b ← fitting_rps

    for all rpId in distinct(rpId) do                                   ▷ Combinatorics phase
        rp_u_lines_set ← distinct(line_no) for u direction
        rp_v_lines_set ← distinct(line_no) for v direction
        UVcombinations ← rp_u_lines_set × rp_v_lines_set
    end for
    fitting_combinations ← []
    for all UVpairs in UVcombinations(RP_a) do
        fitting_combinations ≪ UV_pairs × UVcombinations(RP_b)
    end for

    fitted_tracks ← []
    for all combination in fitting_combinations do                      ▷ Fitting phase
        strip_lines ← {Line(x, y, z, dx, dy, dz)} representing hit strip from combination
        seed_track_line ← Line(x, y, z, dx, dy, dz) from optimization configuration
        fitted_tracks ≪ scipy.optimize.least_squares(DistanceSum(strip_lines, seed_track_line))
    end for

    for all track_line in fitted_tracks do                              ▷ Fishing phase
        min_w ← ∞
        for all combination in UVcombination(fishing_rp) do
            strip_lines ← {Line(x, y, z, dx, dy, dz)} representing hit strip from combination
            update min_w with DistanceSum(strip_lines, track_line)
        end for
        track_line.w ← min_w
    end for
    Pick fitted_tracks with minimal weights that cover all u and v lines from fitting_rps
end for
```

The entry point to our program is a repository of pandas dataframes, where according to the configuration we load data from the appropriate .csv files. From these data, HitLineProvider creates lines representing activated silicon strips. This line collection is passed to MultitrackingAlgorithm where the combinatorics, fitting and fishing phases are being run. Then we perform results normalization and forward them for comparison with the expected ones. The comparison is made by the TOTEMComparator module, which reads the track reproduced by our algorithm and compares them with those determined on the basis of the projections u and v provided by TOTEM. The reconstructed tracks and result of comparison are saved to the corresponding .csv files.

4. Data Analysis

Let us introduce the data sample on which we will validate multitracking algorithm.

Data acquisition is performed during fills when LHC is loaded with protons. During each fill there may be several recording periods called runs. Our data sample comes from fill with id 4499 and run with id 9920. According to *https://totem-runlog.web.cern.ch/totem-runlog* this run was recorded on 16th of October 2015 and due to the information about high pileup, we know that the multitracking phenomenon was highly probable. We place run statistics in the list below:

– $310,8 * 10^6 *$ events were registered, which is 25% of all events from this data taking process

– the data volume is 1,64 TB, stored in 1555 .root files

– the energy of the beam was 6,5 TeV

– the LHC optics was $\beta * = 90m$

– there were approximately $5,0 * 10^{13}$ protons in 671 bunches

The data originating from CWSSW has the tree-like structure. To effectively process given data we decided to flatten it into the array of fixed size records. The list below describes record fields as well as the size and possible values.

– eventID - 4B integer - $[0; 310.8 * 10^6]$

– rootFileID - 2B short - $[0; 1555]$

– armID - 1B char - $[0; 1]$

– groupID - 1B char - $[0; 5]$

– rpID - 1B char - $\{3, 4, 5, 20, 21, 22, 23, 24, 25, 103, 104, 105, 120, 121, 122, 123, 124, 125\}$

– uLineSize - 1B char - $[0; 30]$

– vLineSize - 1B char - $[0; 30]$

– uvLineID - 1B char - $[0; 30]$

– direction - 1B char - $\{'u',' v'\}$

– hitPosition - 8B double - hit position in direction U or V

– lineA - 8B double - track projection line *a* coefficient

– lineB - 8B double - track projection line *b* coefficient

– lineW - 8B double - track projection weight (the greater value, the more reliable projection is)

Each of our records has volume of 48B. The data from 1 .root file - usually of size 1TB - is flatten into one binary file with around $1,1 * 10^7$ records of total size above 500MB. Note that .root file contains not only *RPRecognizedPatterns* but also many other data types (as we have shown in Figure 3.1). Thus, even though there is high redundancy in our records they have still much smaller volume.

4.1. Events by number of recognized track projections

Having such big number of events we may be curious how many of them were affected by multitracking. A prerequisite for an event to be classified as multitracking is to recognize more than one line *u* or *v* in a Roman Pot. Thus, the interesting statistic would be the number of events where the maximum number of recognized lines *u* or *v* per RP is greater then one.

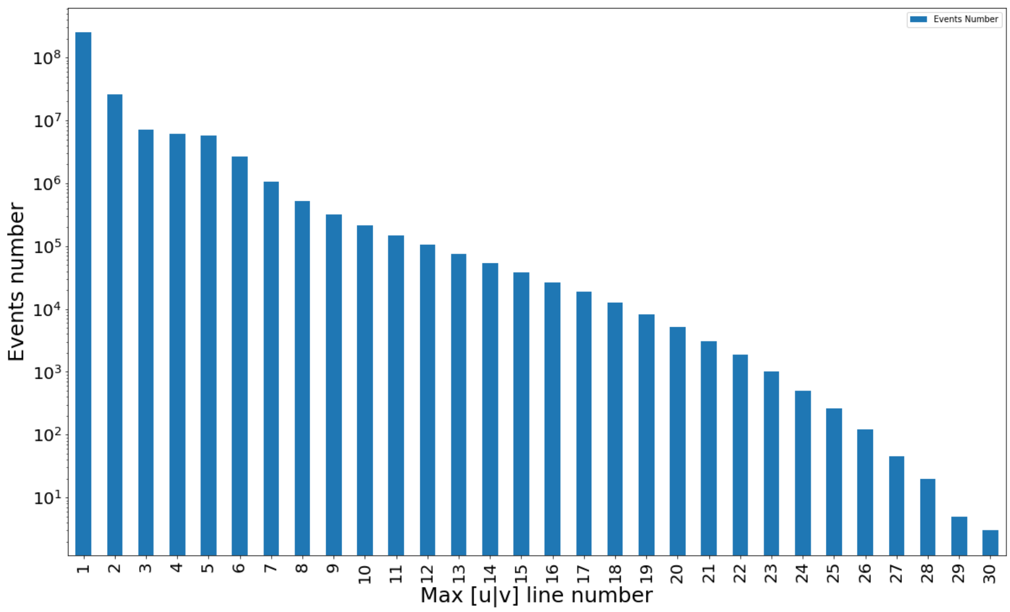

Figure 4.1: Events number for given max [u|v] lines number in any hit Roman Pot

The number of recognized lines u and v depends on whether the following took place during the event:

– single tracking

– multitracking

– noise readouts.

During the collision of beams, the most common case is an elastic collision of one pair of protons, i.e. single tracking. As many as 83.2% of events meet the criterion of single tracking (in each Roman Pot there is at most one line u and v). In the remaining 16.8% of events, multitracking for 2, 3, 4 and more simultaneous tracks could be recorded. However, it is very unlikely that the number of registered protons would be as many as 30 (which is the highest number of recognized track projection lines u and v in a single Roman Pot). There are several reasons why the number of recognized u and v is so large. First, the proton that passes through the detector can react with the atoms that the detector consists of. As a result, we register the so-called ˝showerþhenomenon. The second reason is the electronic noise. Finally, it is worth mentioning that some u and v lines can be the result of the type II error (false negative) of the pattern recognition algorithm.

Max [u|v] line number	1	2	3	4	5+
Events number [mln]	250,112	26,158	7,196	6,176	10,969
Percent [%]	83,201	8,701	2,394	2,054	3,649

Table 4.1: Events number for given maximal number of u or v line in any RP hit during event

About 300.6 million events were registered in the analyzed sample. This is less than reported in the runlog, but it should be remembered that some events could have been filtered out, for example, those during which protons were recorded in Roman Pots, but were not recorded in the T2 detector, or the readouts were on both arms but only in the lower or only in upper Roman Pots. What's more, there are also events during which no hits are recorded in any of the detectors.

4.2. Roman Pot Groups by number of recognized track projections

We have learned the number of analyzed events. However, to reconstruct the track of the proton we will process the data collected within a given group of Roman Pots. For this purpose, we will also analyze hit groups in each event. Let us remind that by group we understand the set of three Roman Pots located on the same arm, and in the same orientation (top, bottom or horizontal).

For each of the events we expect protons passing through detectors in the right and left arm. This is because we should have elastic scattering events. For inelastic scattering one of the protons may be destroyed. Therefore two Roman Pot groups should provide some readouts. Consequently number of registered groups should be twice greater than the number of events, that is around 601.2 millions. However, due to the so-called detectors overlapping and multitracking phenomena, readouts from more than two groups are possible and the total number of registered groups is as many as 639 millions.

We would like to predict how many traces we can reconstruct. We know that 88.9% groups contain at most one track projection in U and V direction. From them we may reconstruct at most 568.3 millions of tracks.

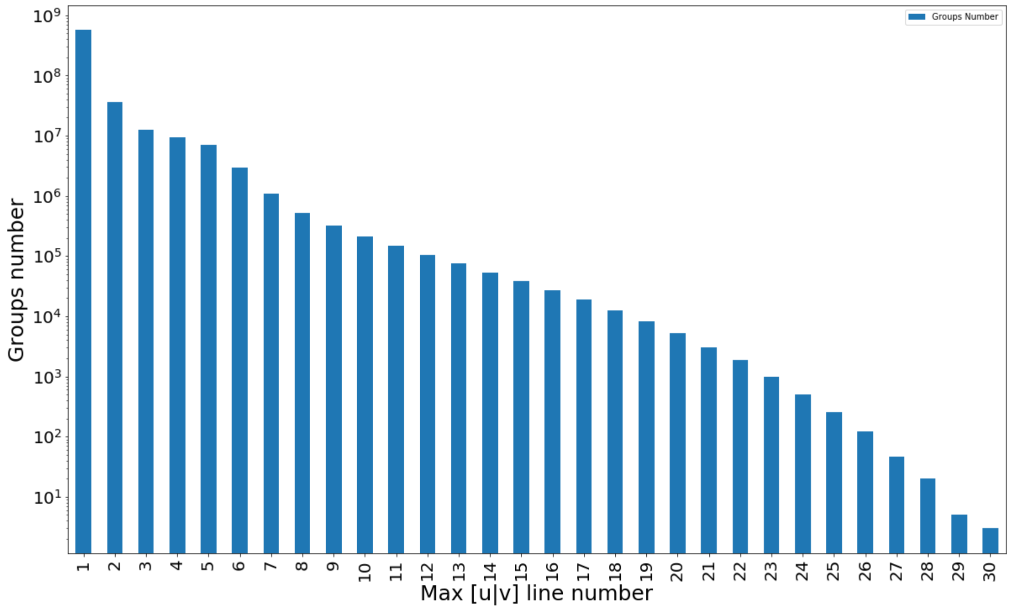

Figure 4.2: Groups number for given max [u|v] lines number in any hit Roman Pot

Max [u|v] line number	1	2	3	4	5+
Groups number [mln]	568,324	36,159	12,434	9,481	12,641
Groups percent [%]	88,934	5,658	1,946	1,484	1,978
Potential tracks [mln]	568,324	72,318	37,302	0	0
Potential tracks percent [%]	83,83	10,667	5,502	0	0

Table 4.2: Groups number for given maximal number of u or v line in any RP hit during event

In the case of a larger number of projections, it is possible to reproduce more traces. We must remember though, that not all of them may belong to real traces.

Let's analyze the distribution of groups. With the increase in the number of projections, the number of groups in which the number of projections was recorded is decreasing. However, the number of groups with 3 and 4 projections is comparable. Therefore, it can be assumed that groups with 4 or more projections contain mainly the noise readouts. Let us assume that only groups with 2 and 3 registered projections correspond to real traces.

In this case, groups with 2 projections should provide about 72.3 million traces, and those with three about 37.3 million. Together they may provide about 16% of all traces. Such a result would significantly improve the quality of the reconstruction algorithm.

4.3. Roman Pot hits in the arms

For further analysis of our data sample we will determine the number of events in which exactly 1, 2 etc. Roman Pot detectors were hit. We expect that the most common case will be one in which exactly one group registers proton. It means exactly 3 hit RP per arm. There are some cases when the angle of proton is too small or too big to hit all of detectors from group, thus it is possible that only one or two Roman Pots would have some readouts.

On the other hand, if the single proton passes through overlapping area it can be register by both vertical and horizontal detectors. The more readouts are possible only when multitracking occurs or proton collide with molecules from detector material providing lot of noise readouts from many different directions. In such cases even all Roman Pots may be hit.

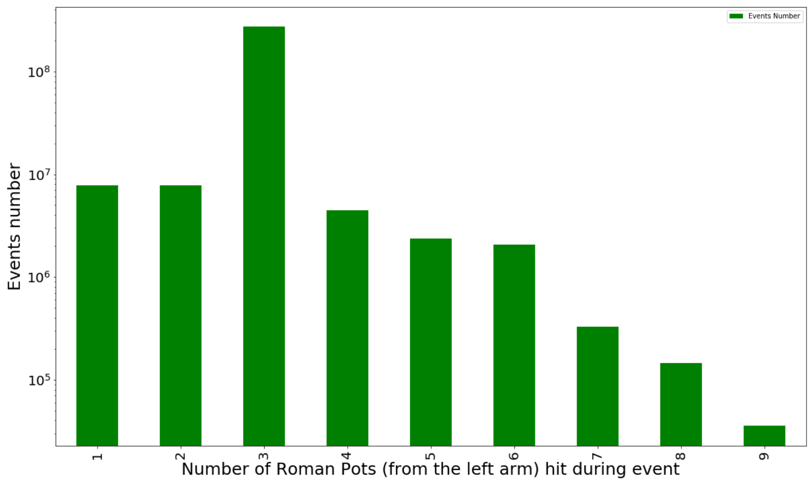

Figure 4.3: Events number for given RP hit number from the left arm

Hit RP	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
Events [mln]	7,827	7,808	273,06	4,442	2,361	2,075	0,328	0,145	0,036
Percent [%]	2,626	2,62	91,606	1,49	0,792	0,696	0,11	0,049	0,012

Table 4.3: Data for all events where given number of RP was hit (from left arm)

As we expected most events were detected by exactly 3 Roman Pots. There are about 2,6% of events where proton angle was too small or too big to hit all of RP from group. We may also observe events with 4-6 detectors hit (single or many protons passing through overlapping area) and finally there were some rarest cases when 7-9 RP registered some particles.

On the right arm we can observe similar distribution as on the left one. However we may notice that right arm registers about 40mln more events where 2 RP were hit and 40mln less where 3 RP were hit. To find the reason of such difference we will investigate Roman Pots geometries, especially how far from the beam are detector edges.

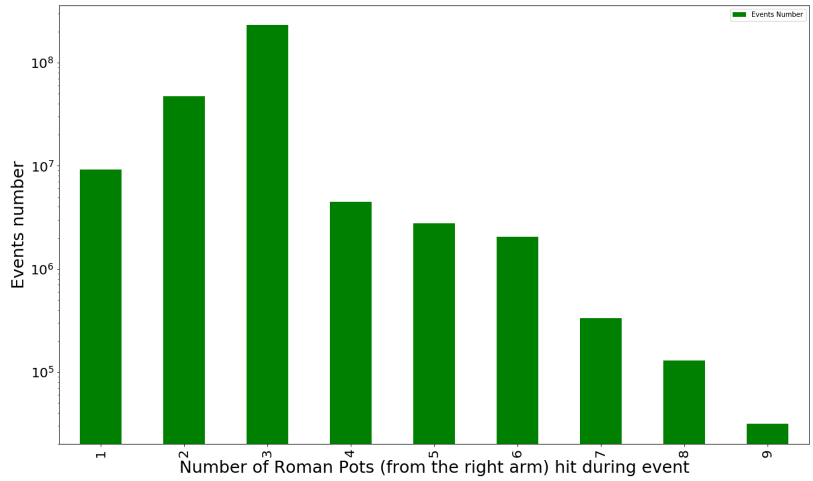

Figure 4.4: Events number for given RP hit number from the right arm.

Hit RP	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
Events [mln]	9,24	47,482	231,585	4,487	2,769	2,041	0,332	0,13	0,031
Percent [%]	3,1	15,928	77,688	1,505	0,929	0,685	0,111	0,044	0,01

Table 4.4: Data for all events where given number of RP was hit (from right arm).

In the figure 3.5. we can observe that during events mainly 6 RP where hit (72,4% of all events). Some protons were missing several RP from groups (those events where 1-5 RP provide some readouts). Then we can see that there were cases where in one arm protons were passing through overlapping area (7-9 detectors hit). Less common case was when in both arms we registered protons in overlapping area (10-12 RP registered particles). Finally, most rarely there were noises in one arm (13-15) or both arms(16-18).

In the Figure 3.6 we compared edge distances from the beam in vertical Roman Pots between corresponding detectors from left and right arm. Most noticeable difference is between RP 20 (left) and RP 120 (right). Right detector is about 1,6 mm (27,1%) more distant than the left one.

In the following chapters, we will show that most hits are recorded in vertical detectors (up to 25 times more). In addition, the most hits fall on the first millimeter from the edge of the detector, so the offset of the order of mm significantly affects the number of readouts.

We may also compare distances in horizontal detectors but the overall number of protons registered by those Roman Pots is rather insignificant.

4.4. Roman Pot groups by hit distribution

In this section we would analyze how many events are registered by horizontal and vertical Roman Pots. Theoretical background provided by TOTEM experiment guides that when LHC optics is set to

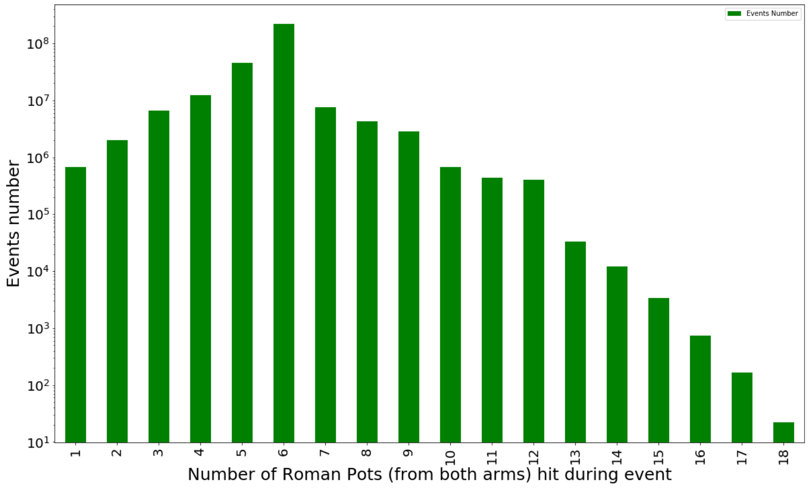

Figure 4.5: Events number for given RP hit number from both arms

Hit RP	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
Events [mln]	0,678	2,007	6,585	12,232	45,265	217,668	7,478	4,262	2,864
Percent [%]	0,226	0,668	2,191	4,069	15,058	72,409	2,488	1,418	0,953
Hit RP	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**
Events [mln]	0,678	0,437	0,407	0,033	0,012	0,003	0,000746	0,000168	0,000022
Percent [%]	0,225	0,145	0,135	0,011	0,004	0,001	>0,001	>0,001	>0,001

Table 4.5: Data for all events where given number of RP was hit (from both arms)

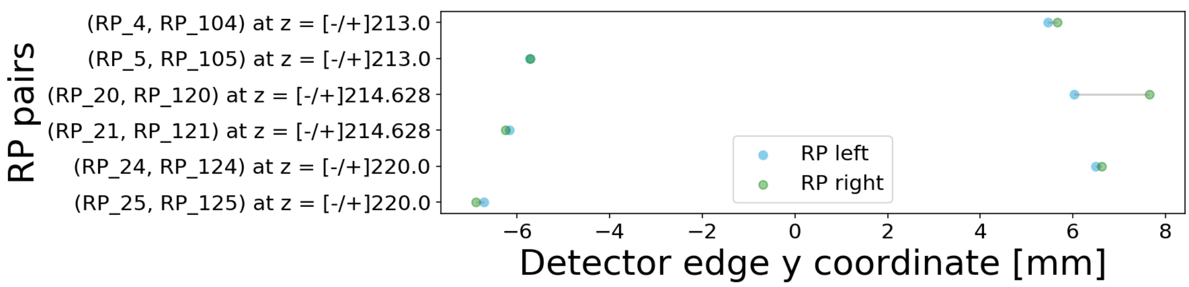

Figure 4.6: Detector edge to beam distance comparison for vertical Roman Pots

$\beta* = 90m$ we should expect definitely more readouts from vertical than from horizontal groups.

Those results are consistent with predictions. The number of events registered in vertical groups are by average 12,5 time greater than in the horizontal ones. Additionally number readouts from corresponding groups from left and right arms do not differ significantly.

RP pairs	**(4, 104)**	**(5, 105)**	**(20, 120)**	**(21, 121)**	**(24, 124)**	**(25, 125)**
RP left dist [mm]	5,459	-5,729	6,02	-6,169	6,487	-6,715
RP right dist [mm]	5,66	-5,709	7,654	-6,247	6,617	-6,883
Dist diff [mm]	0,201	0,02	1,634	0,078	0,13	0,168

Table 4.6: Detector edge to beam distance comparison for vertical Roman Pots

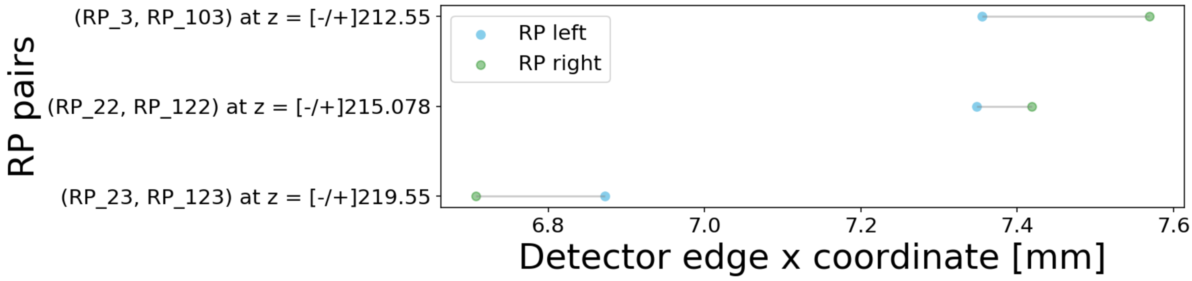

Figure 4.7: Detector edge to beam distance comparison for horizontal Roman Pots

RP pairs	**(23, 123)**	**(22, 122)**	**(3, 103)**
RP left dist [mm]	6,872	7,348 7	7,355
RP right dist [mm]	6,708	7,419 7	7,569
Dist diff [mm]	0,164	0,071	0,214

Table 4.7: Detector edge to beam distance comparison for horizontal Roman Pots

Group	**1 - Top Left**	**2 - Bot Left**	**3 - Hor Left**	**4 - Top Right**	**5 - Bot Right**	**6 - Hor Right**
Events [mln]	153,3	153,7	12,2	156,8	150,9	12,1
Percent [%]	23,99	24,05	1,92	24,54	23,61	1,9

Table 4.8: Number of events where given group of RP was hit

4.5. Number of UV track projections

When reconstructing many traces in a single detector, the first step is to determine their number. For this we will use data about projections in the directions U and V. After filtering out projections, which we consider false, we can predict that the number of recorded traces is the minimum of filtered U and V projections number.

The heat maps in figure 4.9 present for how many Roman Pots the given number of projections has been identified in U and V directions. We can observe three things. Firstly, there are visible diagonals. This is due to the fact that if we are considering the real traces, for each projection u there should be a projection v. Second, there are Roman Pots, in which the number u is different from v, these are the detectors for which pattern recognition produced false results, or noise has been recorded. In the end,

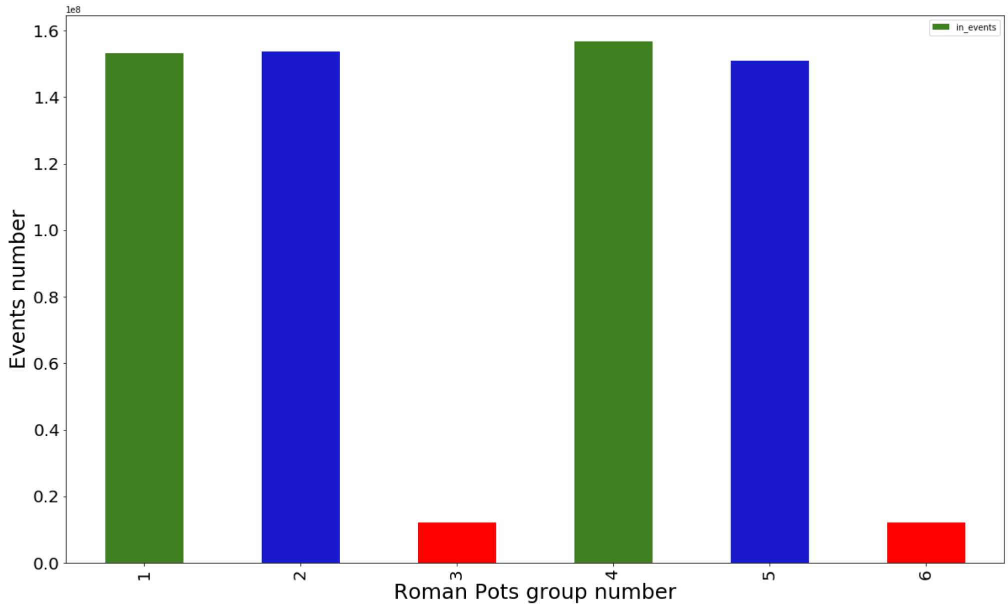

Figure 4.8: Number of events where given group of RP was hit

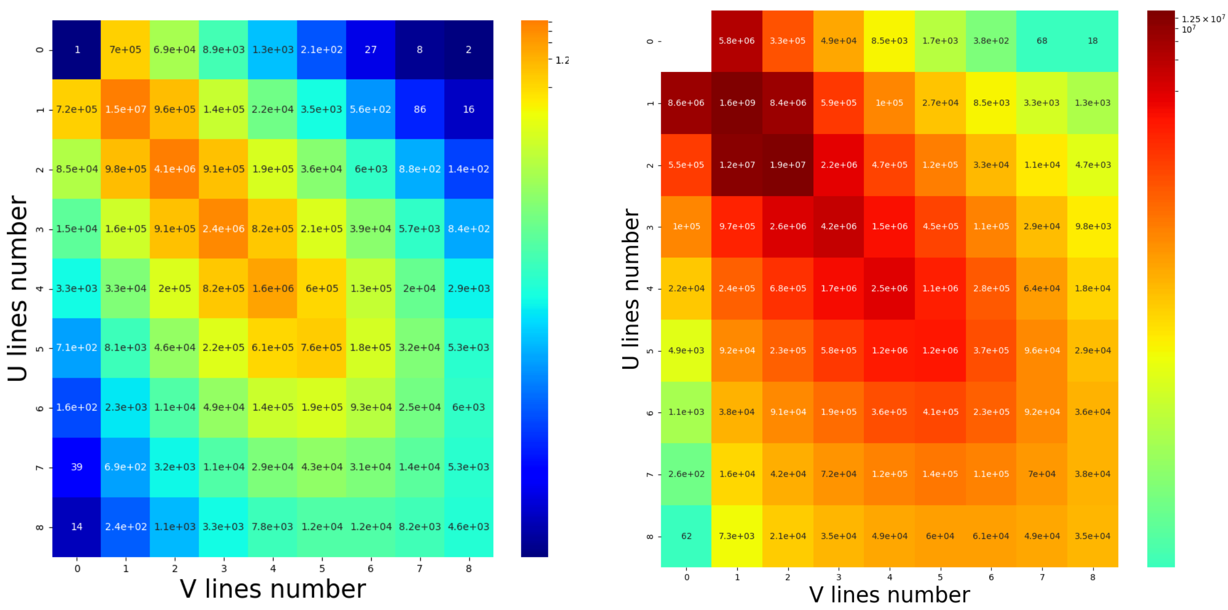

Figure 4.9: Comparison of the number of recognized UV lines between horizontal (on the left) and vertical (on the right) Roman Pots.

the number of all Roman Pots hit is approximately 2 billion, this is the expected result, as the number of recorded groups is almost 640 million, and each group consists of 3 detectors.

4.6. Single tracking analysis

In this section we want to analyze how many traces are there that pass through all, two or one detector from a given group, provided that only a single trace was registered in that group.

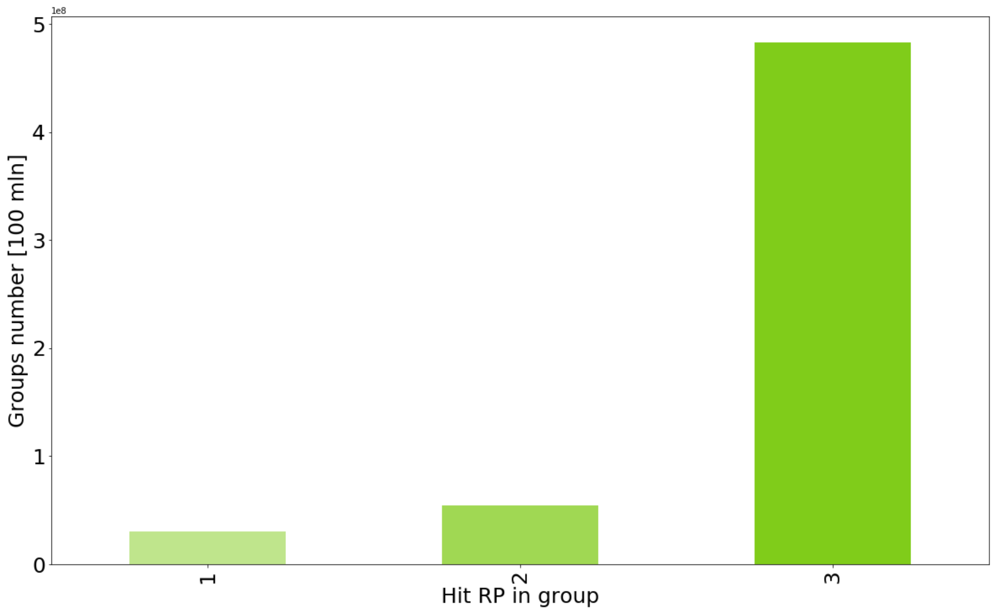

Figure 4.10: Number of groups where given number of RP were hit

Hit RP in group	**1**	**2**	**3**
Groups [mln]	30,3	54,5	483,4
Percent [%]	5,34	9,6	85,06

Table 4.9: Number of groups with given number of Roman Pots hit during single tracking

We may observe that vast majority of protons passes through all detectors which is the standard behavior. Two protons are register probably when the angle is too small to hit the first detector. And the case when only one RP provide some readouts is probably when angle is too small to hit first two ones, or too big to hit the last two, or may be registered only by rotated one. We will discuss it with Table 3.10.

In the above chart we can find the answer to the following question *If in a given group only one detector was hit, which one was it?* Results for top Roman Pots are green, for bottom - blue, and for horizontal - red.

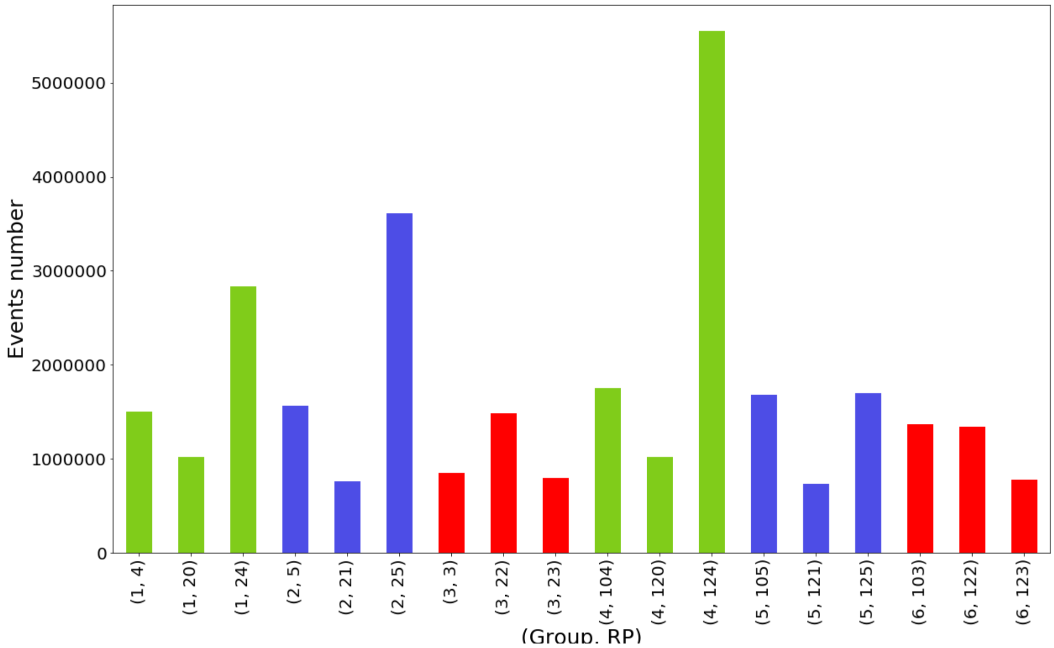

Figure 4.11: Events where only one RP from group was hit

According to predictions, the most often hit detector is the most remote one, and the least often protons are recorded only in the central detector - below twenty percent of cases. This behavior is observed for vertical Roman Pots, for the horizontal distribution of strokes looks different. This may be related to their location, as well as the type of collisions recorded.

(Group, RP)	**(1, 4)**	**(1, 20)**	**(1, 24)**	**(2, 5)**	**(2, 21)**	**(2, 25)**	**(3, 3)**	**(3, 22)**	**(3, 23)**
Events [mln]	1,504	1,016	2,837	1,568	0,763	3,612	0,846	1,484	0,797
Percent [%]	28,07	18,97	52,96	26,38	12,84	60,78	27,07	47,45	25,48
(Group, RP)	**(4, 104)**	**(4, 120)**	**(4, 124)**	**(5, 105)**	**(5, 121)**	**(5, 125)**	**(6, 103)**	**(6, 122)**	**(6, 123)**
Events [mln]	1,753	1,015	5,555	1,677	0,733	1,696	1,372	1,343	0,776
Percent [%]	21,07	12,2	66,74	40,85	17,84	41,3	39,3	38,48	22,22

Table 4.10: Events where only one RP from group was hit

Table 3.10 shows the expected results. The only surprise is the outstanding result for the detector number 124. However, considering its position relative to the beam (Figure 3.6), one can expect the range of proton angles he registers to be much larger.

4.7. Multi tracking analysis

In this section, let us analyze the distribution of hits in the detectors where the phenomenon of multitracking was detected. At the beginning we will look at how many Roman Pots from a given group were hit, and then we will analyze which of them are most often hit as the only one from the group.

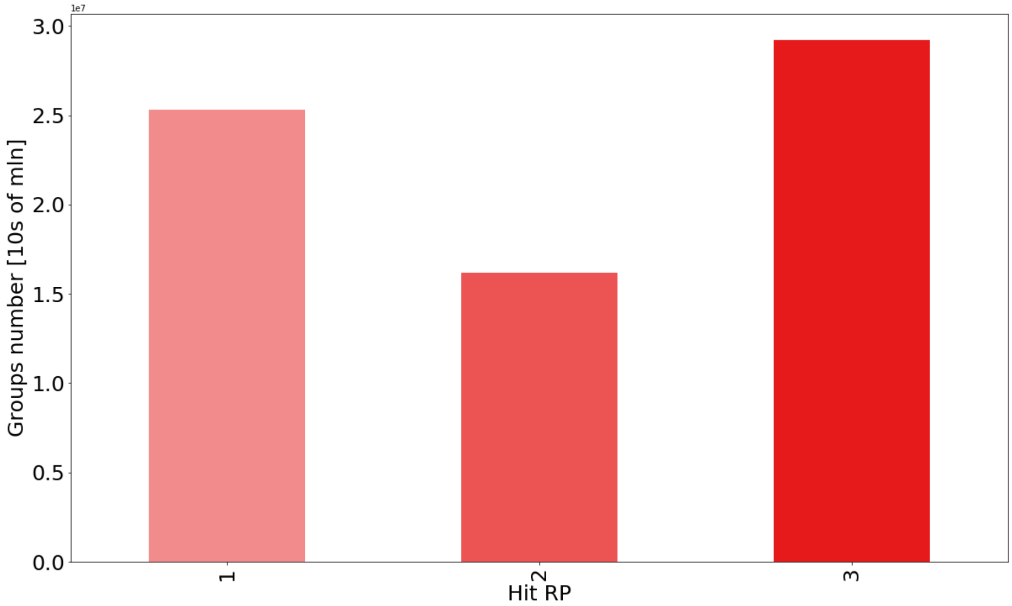

Figure 4.12: Number of groups where given number of RP from group where hit during multitracking events

Hit RP in group	**1**	**2**	**3**
Groups [mln]	25,321	16,17	29,225
Percent [%]	35,81	22,87	41,33

Table 4.11: Number of groups where given number of RP from group where hit during multi tracking events

The distribution from Figure 3.14 differs significantly from Figure 3.12 (for single tracking). Definitely more often (over 35 percent of cases) only one detector from the group provides readouts. These are probably recorded noise or the so-called shower phenomenon. Most frequently, however, protons are recorded by all Roman Pots from the group (over 41 percent of cases), these readings most likely correspond to real proton tracks.

As the vast majority of events are recorded by vertical detectors, we will only analyze these cases. Among the Roman Pots from Top and Bottom groups, the detector more distant from the collision point, the more often it is the only one providing readouts for multitracking. Such a distribution is strongly related to the distance of individual detectors from the beam.

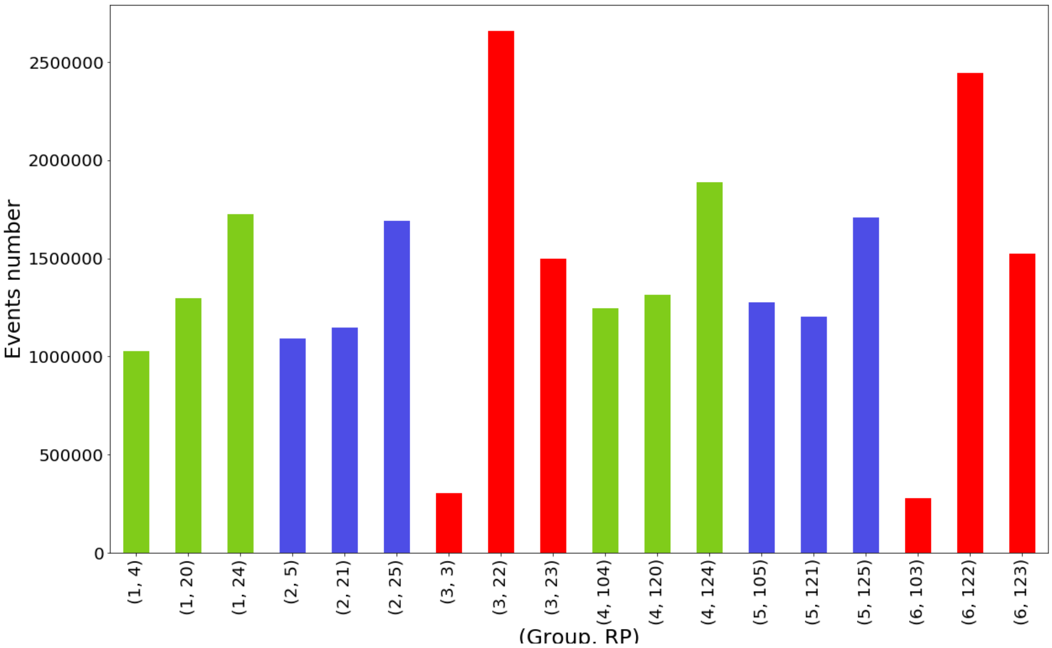

Figure 4.13: Number of groups where given number of RP from group where hit during multitracking events

(Group, RP)	**(1, 4)**	**(1, 20)**	**(1, 24)**	**(2, 5)**	**(2, 21)**	**(2, 25)**	**(3, 3)**	**(3, 22)**	**(3, 23)**
Events [mln]	1,029	1,296	1,726	1,093	1,145	1,689	0,302	2,66	1,499
Percent [%]	25,4	31,99	42,61	27,84	29,16	43	6,78	59,62	33,6
(Group, RP)	**(4, 104)**	**(4, 120)**	**(4, 124)**	**(5, 105)**	**(5, 121)**	**(5, 125)**	**(6, 103)**	**(6, 122)**	**(6, 123)**
Events [mln]	1,248	1,315	1,888	1,274	1,202	1,707	0,277	2,444	1,524
Percent [%]	28,03	29,55	42,41	30,46	28,72	40,81	6,53	57,56	35,9

Table 4.12: Multitracking events where only one RP from group was hit

5. Results

In the research we have determined proton tracks for data registered during data taking in LHC in October 2015. The algorithm has been analyzed for complexity and the average time of execution per event.

The quality of tracks was examined in two cases:

– For the single tracking phenomenon

– For the multi tracking phenomenon

5.1. Validation

This section will give us the answer for the following questions: Does the algorithm work correctly? What criteria were used to verify the correctness of the algorithm?

The validation process is divided into two steps - the first for single tracking and the second for multitracking. In the first one, we compare the results of our algorithm with those provided by the TOTEM experiment. The main subject of our analysis will be the distribution of hits in horizontal and vertical detectors for both algorithms. In the case of multitracking, we do not have an access to the validation dataset. Therefore our assessments should be based on the predicted distribution of hits for the optics of the detector $\beta* = 90m$

We have applied the following validation criteria:

– reconstructed tracks are consistent with LHC physics

– for single tracking the distribution of detector hits is consistent tracks reconstructed by the TOTEM experiment

– for multitracking the distribution of detector hits satisfy predictions for $\beta* = 90m$ optics

– normalized χ^2 statistics of reconstructed tracks is lower or comparable with 1.0

– average track distance from activated silicon strips is of the order of 1σ.

One of metrics used to assess the algorithm quality is χ^2 statistics. To consider the reconstruction as reliable this statistics should be less than 1.0. However, the track for which χ^2 is slightly higher may also be a good approximation of the actual proton track. The other statistics we may want to investigate is the average distance between the track and activated silicon strips. The silicon detector has 512 silicon strips and the distances between them is constant:

$$\sigma = 0.0659mm$$

We will use σ in following sections.

5.2. Single tracking results

The results presented in this section come from FILL 4499, RUN 9920, in which $310.8 * 10^6$ events were recorded. We will show distribution of projections of tracks on the center plane of the RP 20 detector (from the left arm, Top group) located at $z = -214.628m$ and from the center plane of RP 22 (from the left arm, Horizontal group) located at $z = -215.078m$. We can compare the distributions of hits from tracks reconstructed with the new multitracking algorithm and those that correspond to TOTEM's recognized U and V projections.

5.2.1. Single tracking in vertical detectors

During the reconstruction process for data from fill 4499, more than $1.3 * 10^8$ tracks were reconstructed in the top group on the left detector arm.

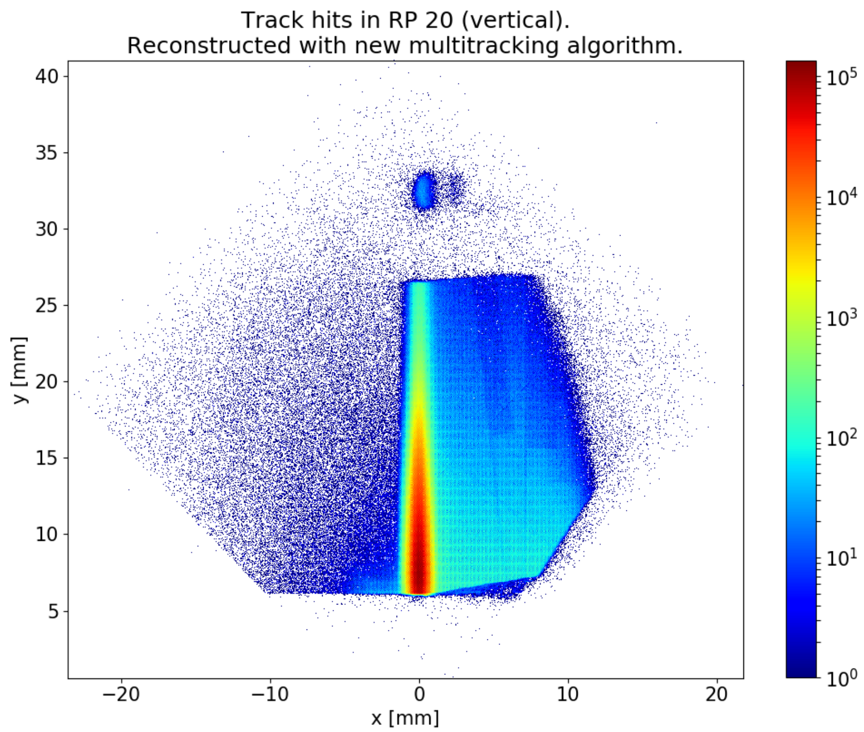

Figure 5.1: Single tracking hit distribution in RP 20 (vertical). Track reconstructed by multitracking fitting phase. Binning 800x800. Each pixel corresponds to a square of the side of 0.05mm.

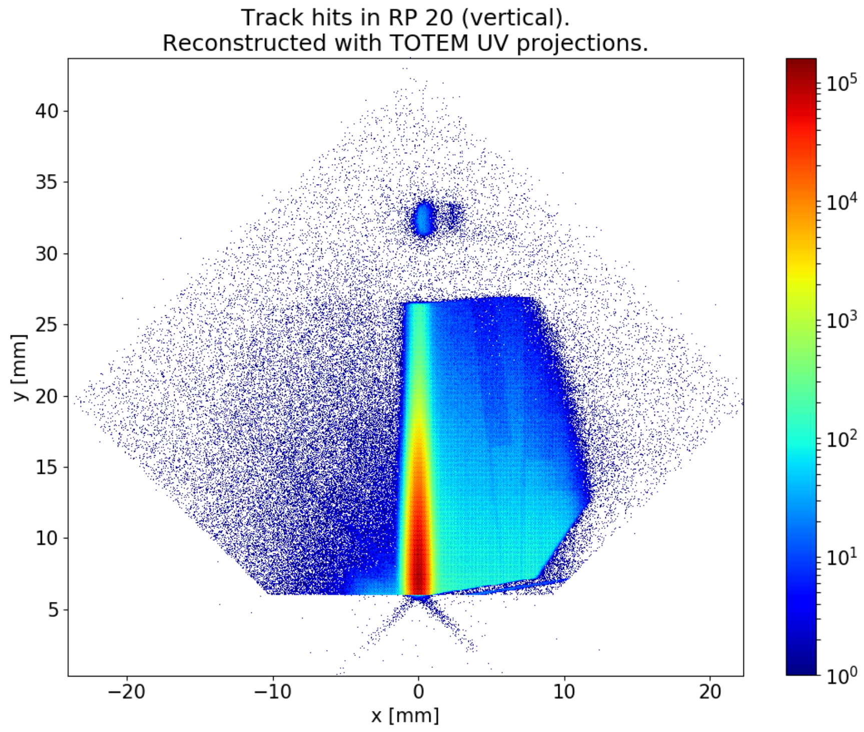

Figure 5.2: Single tracking hit distribution in RP 20 (vertical). Tracks correspond to TOTEM's recognized U and V projections. Binning 800x800. Each pixel corresponds to a square of the side of 0.05mm.

The vast majority of tracks are shown in the figures 5.1 and 5.2. It is worth to mention that the 82 tracks recreated by our algorithm and 8 tracks reproduced directly from the U and V projections were outside the plot boundaries.

The distribution of the reconstructed tracks in both figures is very similar, on both of them we can see a characteristic flame for the optics $\beta* = 90m$ (corresponding to the elastic protons) and higher intensity of hits on the right side (originating mainly from diffractive protons). Due to 800x800 binning we are able to notice the bands. This is the effect of the distance between the silicone strips ($\sigma = 0.0659$), which is greater than the size of one pixel in the figure ($0.05mm$).

In the picture with the distribution of hits reconstructed with the help of the lines *u* and *v*, we can see a clear outline of the detector, in the case of our algorithm, the boundary of the upper edges is more fuzzy. In addition, in both cases we can observe the outline of the station rotated by 7 degrees in the form of a raised right part of the lower edge of the detector.

In both cases we can observe non-physical tracks lying outside the detector area. In the case of our algorithm, erroneous tracks may appear when the stop condition of fitting phase is met in non optimal solution. In the case of non-physical tracks reconstructed with projections *u* and *v*, one of the possibilities is that the TOTEM module responsible for pattern recognition selected incorrect projections.

5.2.2. Single tracking - Horizontal detectors

In the horizontal detector, we have recreated approximately 464 thousands of tracks - almost 300 times less than in the vertical detector. The reason of such disproportion is the LHC optics $\beta* = 90m$ for which the vast majority of registered protons are the elastic ones which mainly hit vertical detectors. In the horizontal detector, we can observe mainly diffractive protons.

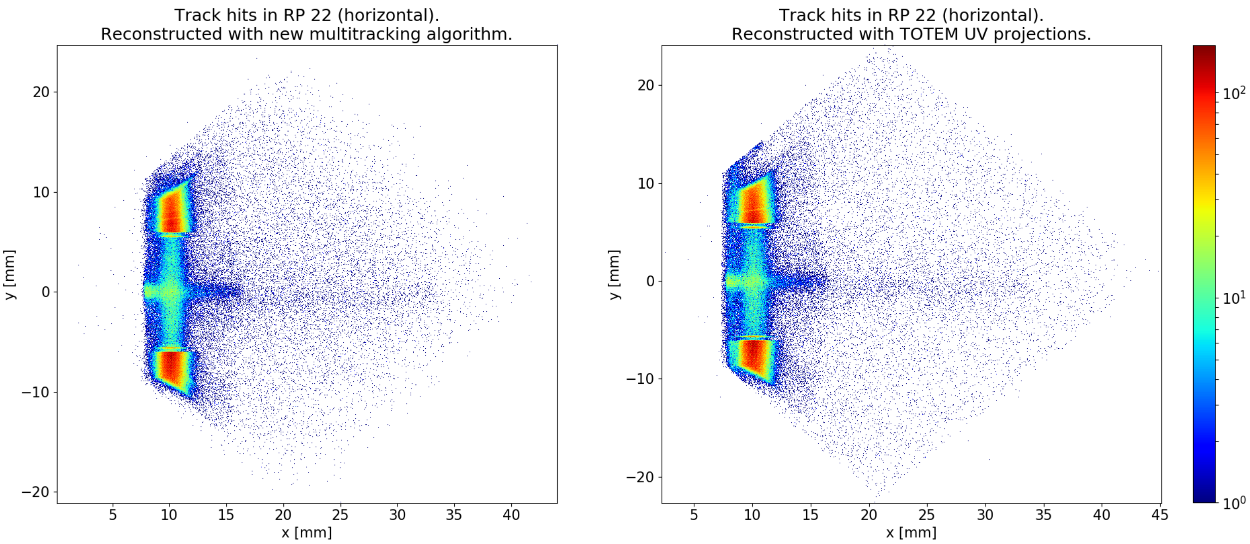

Figure 5.3: Comparison of single tracking hit distribution in RP 22 (horizontal) for tracks reconstructed by multitracking fitting phase (on the left) and those which correspond to TOTEM's recognized U and V projections (on the right). Binning 800x800. Each pixel corresponds to a square of the side of 0.05mm.

Comparing the results of our algorithm with those reconstructed with use of the *u* and *v* projections, we note that the distribution of hits is very similar and, most importantly, consistent with the predicted one. In both cases we observe two clusters in the overlapping area. The outline of the detector is also visible.

In the above figure there are sporadic cases of tracks outside the detector's outline. Similarly as in the case of vertical detectors, the most likely cause may be incorrect selection of *u* and *v* projections or a wrong stop condition for the fitting phase.

5.2.3. Single tracking - statistics

In this section, we will analyze the following statistics of the reconstructed tracks: χ^2, average distance between tracks and activated silicon strips as well as the track fitting time.

Among the aforementioned statistics, we will particularly focus on the results of χ^2. Let us remind that one of our correctness criteria for reconstruction is χ^2 of the order of 1.0 or less.

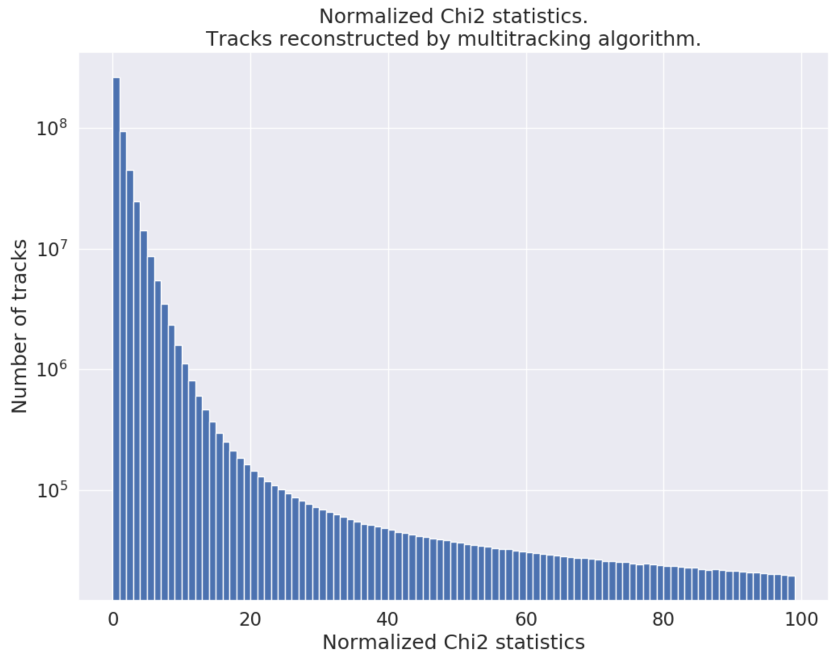

Figure 5.4: Normalized χ^2 statistics of all reconstructed single tracks. 100 bins of width = 1.0.

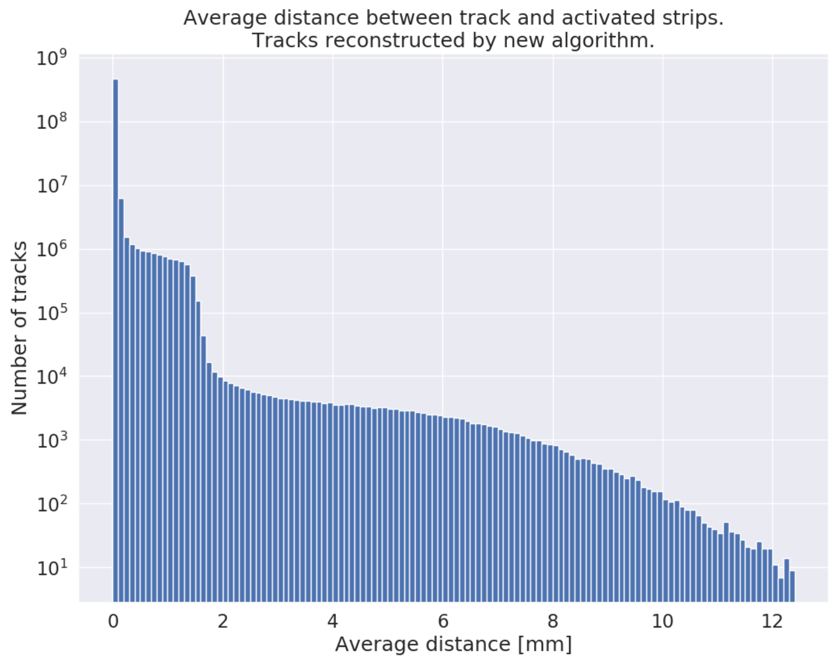

Figure 5.5: Average distance between track and activated silicon strips for all of reconstructed single tracks. 125 bins of width = 0.1mm.

The charts depicts statistics for over $4.83 * 10^8$ tracks. While we analyze the presented statistics, we can make the following observations:

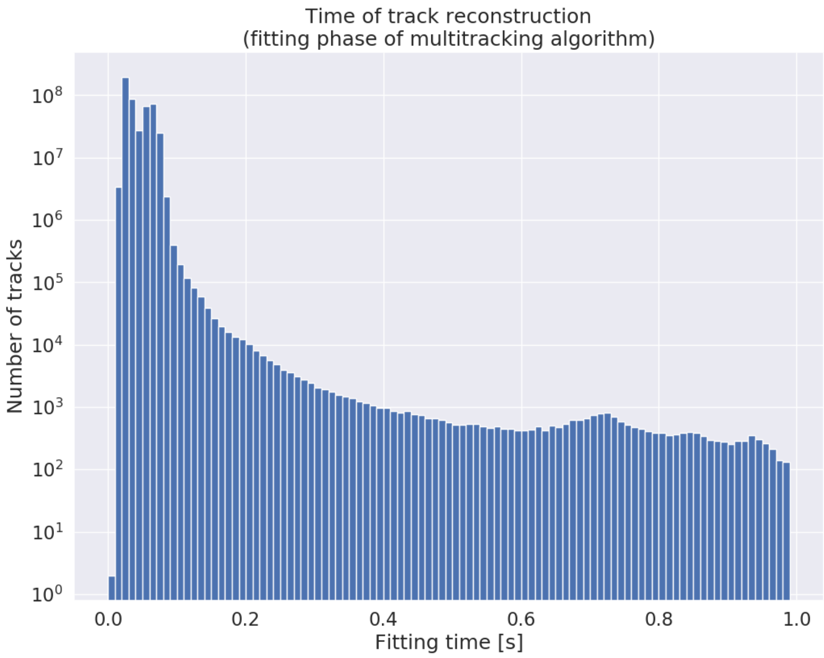

Figure 5.6: Fitting execution time for all of reconstructed single tracks. 100 bins of width = 0.01s.

		X bins contain N% of results				N% tracks have lower statistic than x			
Statistics	**bin_size**	**1_bin**	**2_bin**	**5_bin**	**10_bin**	**50%**	**90%**	**95%**	**99%**
χ^2	1.0	74.48%	83.86%	93.71%	96.61%	1	5	8	None
Avg. distance	0.1	97.65%	97.96%	98.61%	99.44%	0.1 mm	0.1 mm	0.1 mm	0.9 mm
Fitting time	0.01	0.69%	41.26%	78.82%	99.90%	0.04 s	0.07 s	0.08 s	0.08 s

Table 5.1: Single tracking statistics. χ^2, average distance between reconstructed track and activated silicon strips and fitting time.

– For χ^2 and average distance between track and activated strips the most tracks fall into the first bucket

– χ^2 for 90% of tracks has values lower than 5.0 which proves high quality of reconstruction

– For over 97% of tracks, the average distance between the track and activated strips is $0.1mm$ that is $1.5 * \sigma$

– The time of fitting of 50% tracks is lower than $0.04s$, which in theory gives more than 25 fits per second which is considered to be a good result, additionally as much as 99.9% of tracks are reconstructed in less than $0.1s$. Thanks to this information, we can estimate the computation time for any data sample size.

– There are tracks with considerably higher time of fitting than $0.1s$. These tracks could be fitted to incorrectly determined projections *u* and *v*.

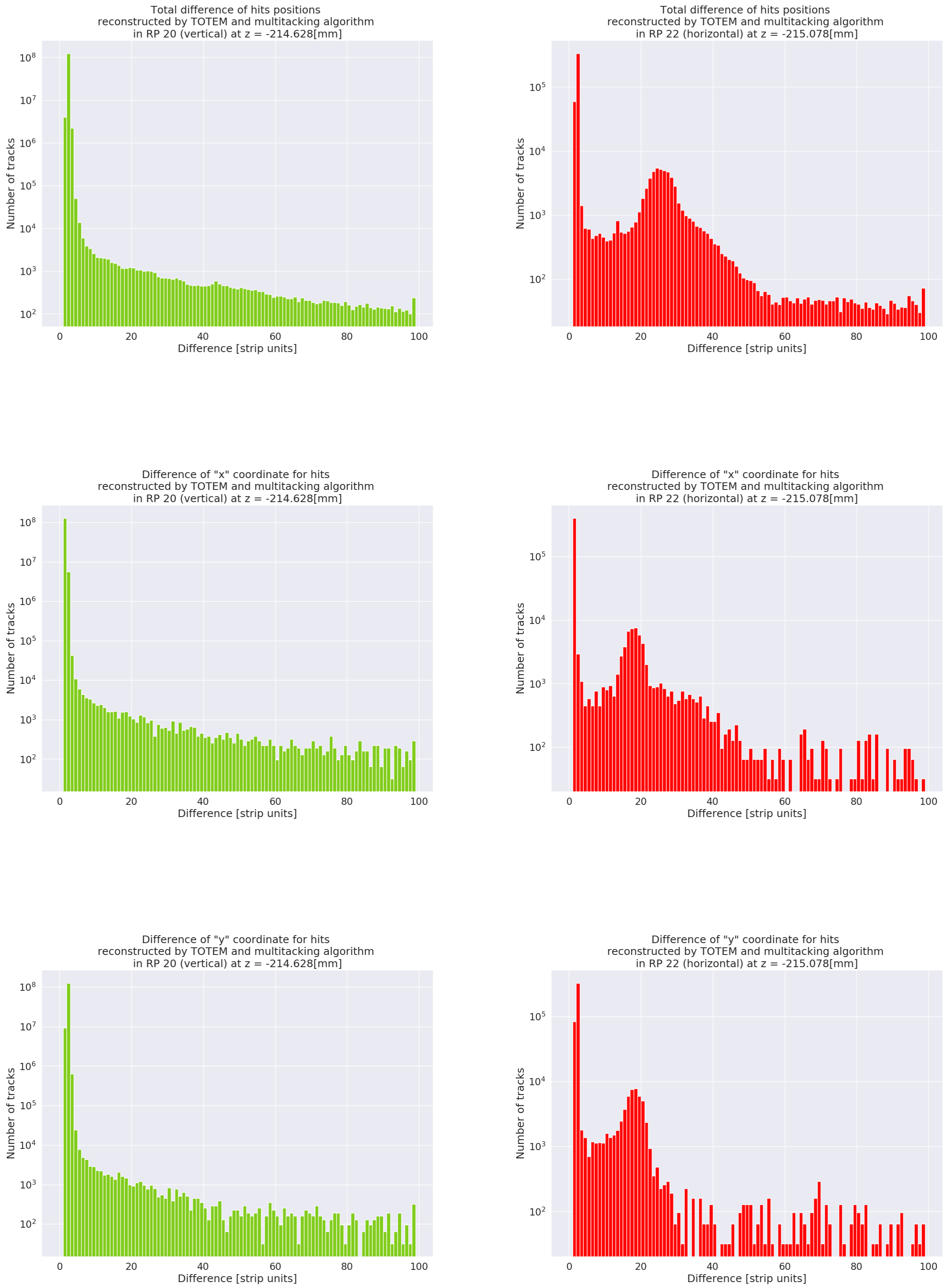

Figure 5.7: Comparison of coordinates and position differences of hits reconstructed by TOTEM and our algorithm in the vertical (RP 20 - green charts) and horizontal (RP 22 - red charts) detectors.

Figure 5.7 depicts differences between reconstructed hit positions computed by our algorithm and TOTEM's reconstruction process, based on recognized *u* and *v* track projections.

The largest percentage of hit points differs by no more than 1σ in "x" coordinate and no more than 2σ in "y" coordinate. This results in a general difference of hit positions being no more than 2σ for the majority of reconstructed tracks. We may ask why do we notice larger differences for the "y" coordinate than of the "x" one? We can presume that this difference refers to the larger spread of "y" coordinate than "x" coordinate for all of the reconstructed hits.

		X bins contain N% of results				**Difference for first N% results**			
Difference of	**Bin size**	**1 bin**	**2 bin**	**5 bin**	**10 bin**	**50%**	**99%**	**99.9%**	**99.99%**
Total	σ	3,06%	98.18%	99.94%	99.95%	3σ	4σ	5σ	None
X	σ	95.72%	99.90%	99.95%	99.96%	2σ	3σ	3σ	97σ
Y	σ	6.79%	99.45%	99.95%	99.96%	3σ	3σ	4σ	None

Table 5.2: Difference of hit positions reconstructed by TOTEM and multitracking algorithm in Roman Pot 20 (vertical). We assume σ = 0.0659mm.

		X bins contain N% of results				**Difference for first N% tracks**			
Difference of	**Bin size**	**1_bin**	**2_bin**	**5_bin**	**10_bin**	**50%**	**90%**	**95%**	**99%**
Total	σ	12.72%	84.60%	85.17%	85.66%	3σ	25σ	30σ	None
X	σ	84.61%	85.22%	85.67%	86.38%	2σ	18σ	21σ	None
Y	σ	17.42%	85.72%	86.53%	87.83%	3σ	16σ	20σ	None

Table 5.3: Difference of hit positions reconstructed by TOTEM and multitracking algorithm in Roman Pot 20 (horizontal). We assume σ = 0.0659mm.

Analyzing the data from the Table 5.2 and Table 5.3 above tables, we can observe that the smaller differences in the reconstructed hits were recorded for the vertical detector. However, the difference of hit position at the level of maximum 2σ for 98.2% hits in the vertical detector and 84.6% in the horizontal detector indicates on the high quality of the reconstructed tracks in both types of detectors.

5.3. Multitracking

Let us remind that the process of track reconstruction is based on the recognized lines *u* and *v*. Each such line is associated with weight and a set of hit positions. The weight informs how well the line has been recognized (the higher weight, the higher possibility that line represents the real track projection). During the tests, we noticed that the higher the weight, the lower the standard deviation of hit positions. In addition, projections with a weight less than 3.0 are often associated with an incomplete number of detectors. According to presented observations, we decided to consider as a noise those lines which weight is less than 3.0 or the standard deviation of the hit positions is greater than 1.0.

5.3.1. Multitrack types

After removing the noise projections, we can get different distributions of the numbers of lines *u* and *v* in detectors from a given group. Depending on the distribution, it is possible to reconstruct a different number of tracks. For example, the multitracking problem may be reduced to the reconstruction of one track. We decided to classify the tracks depending on the distribution of projections *u* and *v* within the group.

The list below presents the classification of tracks.

– **Type 0**: There is a detector in the group that does not have projection *u* or *v*.

– **Type 1A**: All three detectors in the group have exactly 1 *u* and *v* line. The problem has been reduced to the reconstruction of a single track.

– **Type 1B**: Two detectors from the group have exactly 1 *u* and *v* line. The third one has more projections. The problem is reduced to the reconstruction of a single track with an additional phase of fishing in the 3rd Roman Pot.

– **Type 2**: Two detectors from the group have two lines *u* and *v*. The third have more or exactly 2 projections in each direction. It is the pure multitracking case. During the algorithm, all three phases are present (combinatorics, fitting, fishing).

– **Type 2B**: Two detectors from the group have two lines *u* and *v*. The third has less than 2 projection in any direction.

– **Type 3**: Other types. Probably it is possible to reconstruct more tracks, but at a much higher cost due to the complexity of the combinatorics phase.

Group type	**Type 0**	**Type 1A**	**Type 1B**	**Type 2**	**Type 2B**	**Type 3**
Group number [mln]	3,157	16,736	5,506	1,846	0,275	1,704
Group percent [%]	10,804	57,268	18,839	6,317	0,941	5,831

Table 5.4: Track types by number and percent from all events in Fill 4499, Run 9920.

The total number of groups that qualified to multitracking reconstruction was 29.2 million. Our algorithm has reconstructed tracks for groups "1A", "1B" and "2", which together gives 24.1 million groups (82.4% of all). We presume that only groups "2B" and "3" contain data sufficient for reconstruction other real tracks. However, these are only 6.8% of all groups.

5.3.2. Vertical detector

In this subsection we will show hit distributions on the plane located in the center of Roman Pot 20 from Top group, on the left arm. The exact position of plane is $z = -214.628m$. Hits registered by vertical detectors belong mainly to elastic protons.

In order to reduce noise hits we applied two cuts. At first we want to reject tracks which miss any detector from the group. Then we will focus on the average distance between track and registered strip hits. In ideal case we would assume that tracks should not miss the hit strips by more than σ on average. However, we need to consider the clustering that aggregates the activated strips, reducing the resolution. Therefore, we will reject only those tracks that passes strips by more than 10 σ on average. We must also remember that the fitting algorithm is based on optimization methods that work with finite accuracy and may interrupt the search without reaching the global optimum.

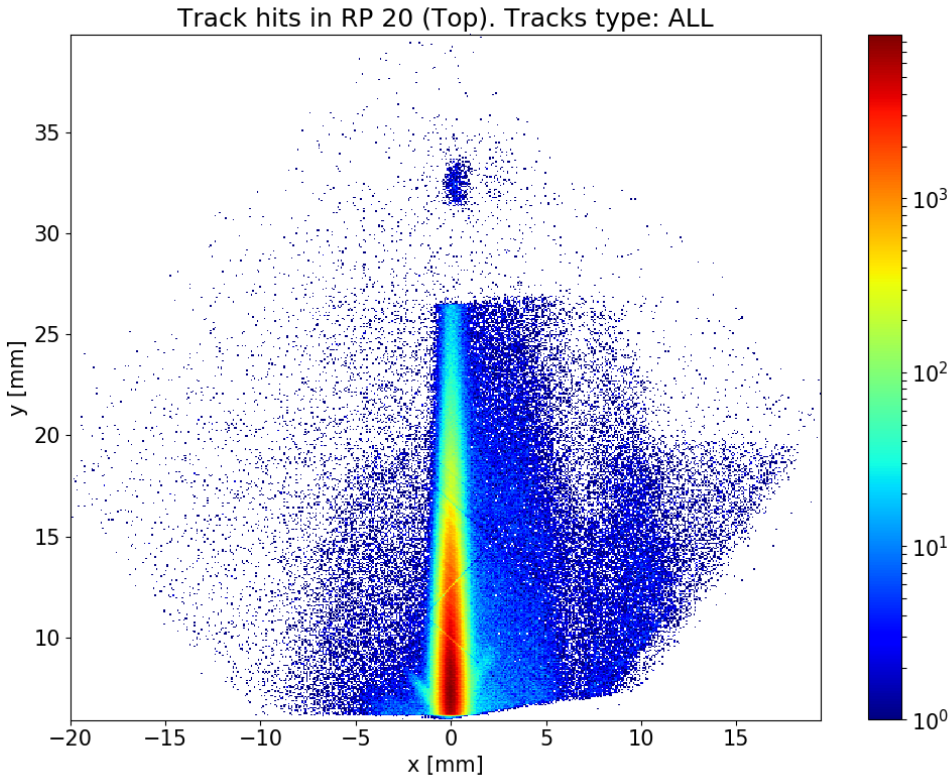

Figure 5.8: Hit distribution in RP 20 - all of tracks

Over 5.5 million track projections are visible in the plot (for all three track types - 1A, 1B, 2). Plotting surface is 40 mm by 40 mm. It is divided by 500 by 500 grid. Therefore, each pixel corresponds to a square with 0.08 mm side.

The distribution of hits is consistent with the predictions presented in [19]. The outline of the detector is also visible (a square with a cut corner). Vertical detectors register mainly elastic protons, therefore the majority of them are focused on the axis $x = 0$ creating a distribution resembling a flame shape. Some of the protons may come from diffractive reactions which may be seen as the intensity on the right side of flame.

Note that as the x-coordinate increases, starting with $x = 0$, the detector's edge rises. The reason is the location of one of the stations, which has been deviated by an angle of 7 degrees. However, there are few visible hits below the edge, for x between 5mm and 10mm. Further we will see that, their source are tracks of type 1B.

There are also few characteristic slanting lines passing through the flame. They represent places where fewer hits were recorded, those slanting lines indicate damaged silicon strips.

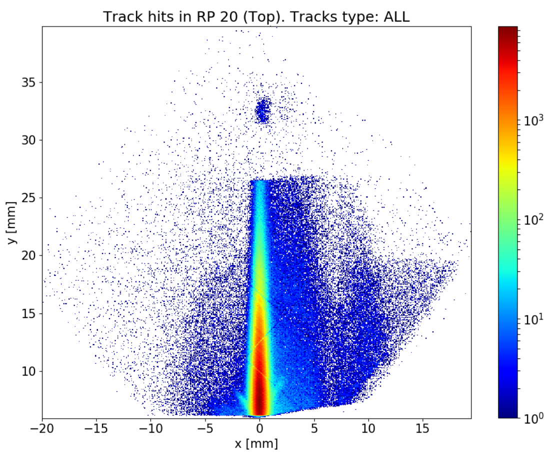

Figure 5.9: RP 20 ALL hits

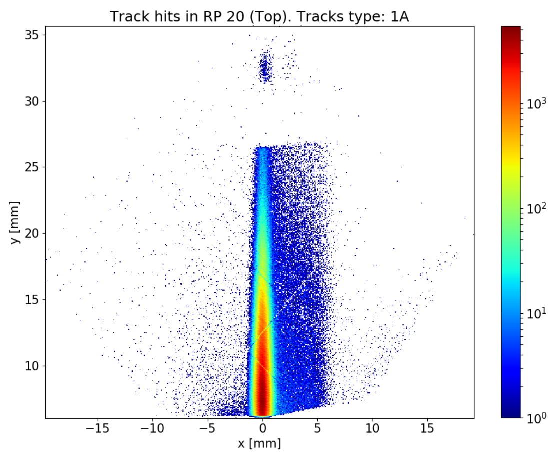

Figure 5.10: RP 20 1A hits

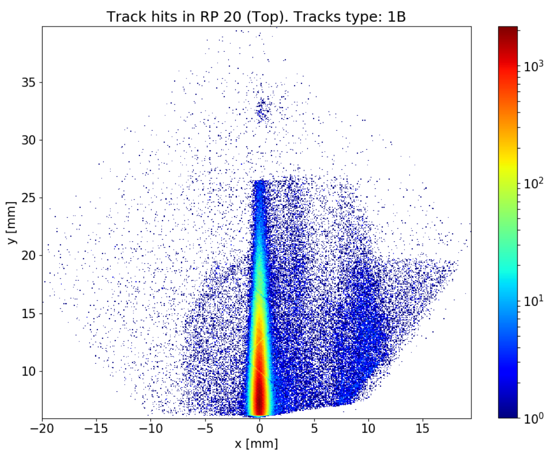

Figure 5.11: RP 20 1B hits

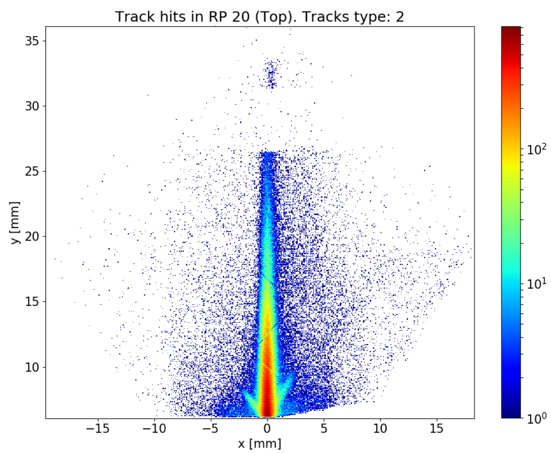

Figure 5.12: RP 20 2 hits

Comparing distributions for all tracks, as well as for individual types, we may notice that each of them contain a characteristic - for $\beta* = 90m$ optics - flame. Each plot reveals the influence of the tilted station (visible as a tilted edge). Moreover there are visible slanting lines where the silicon strips have been damaged. Note that one distribution (type 1B) contains nonphysical tracks (registered outside the detector). Probably there is some error in code responsible for fishing the correct tracks for this type.

When generating charts, some tracks were rejected. Cut 1 is responsible for removing tracks that do not pass through all detectors from the group. Cut 2 is responsible for those whose average distance from

	Tracks [mln]	**Cut 1 [%]**	**Cut 2 [%]**	**Plotted tracks [mln]**	**Plotted tracks [%]**
All	5,517	0,205	3,489	5,313	96,306
Type 1A	3,622	0,04	1,299	3,574	98,661
Type 1B	1,294	0,236	4,393	1,234	95,372
Type 2	0,601	1,129	14,744	0,506	84,128

Table 5.5: Track type statistics for RP 20 (from Top group on the left arm)

hit silicone strip was more than $10 * \sigma$

It is worth noting that the most numerous group of tracks are those that reduce to single tracking. This is over 65.65% of all cases. 23.45% of tracks were reduced to single tracking with additional fishing phase. Only 10.89% of tracks were reconstructed by original multitracking algorithm. Further analysis shows that those tracks are registered only in 0.39% of all events, and still more than 15% those reconstructions are rejected.

Please note that the most tracks are filtered out due to the average distance from the activated silicon strips. The reason of this may lay in not optimal way of matching track pairs in the original multitracking algorithm or in limitations related to the optimization method used in the fitting phase.

Nevertheless, we may consider reconstructing 96.3% of tracks with an accuracy of $10 * \sigma$ (or better) to be a satisfying result.

5.4. Horizontal detector

Horizontal Roman Pots mainly register particles coming from diffractive reactions. Let us analyze track projection on plane located in Roman Pot 22 in the horizontal group, on the left arm. Plane is located in the detector center, on $z = -215.078m$.

The following figures show tracks that pass through all detectors and their average distance from activated silicone strips is no more than $20 * \sigma$. A 300 x 300 binning has been used for each drawing.

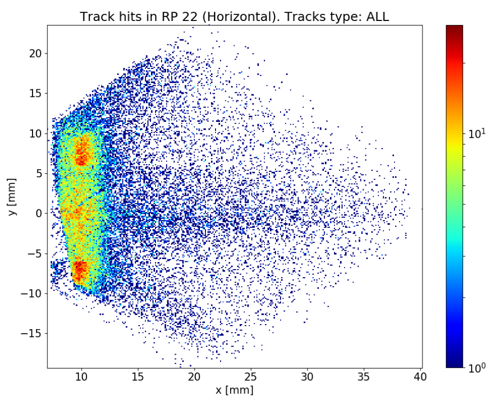

Figure 5.13: Hit distribution in RP 22 - all of tracks

Plotted hits, generated mainly by diffractive protons, usually tend to focus along horizontal axis near $y = 0$. We may observe two clusters near the vertical edge of the detector as well as intensified distribution along side edges. Note that these regions lies in the overlapping area of horizontal and vertical detectors.

Moreover, according to our predictions, a slight rotation of the detector is visible, as well as a significant part of the tracks outside the detector area. This are most probably tracks reconstructed by the single tracking algorithm with additional fishing. We will verify it with following figures.

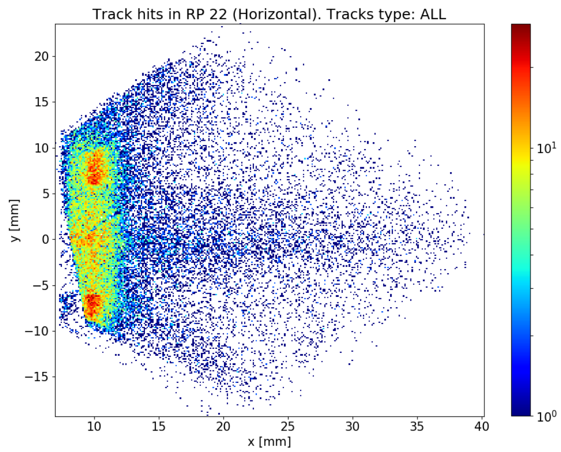

Figure 5.14: RP 22 hits. Track type ALL.

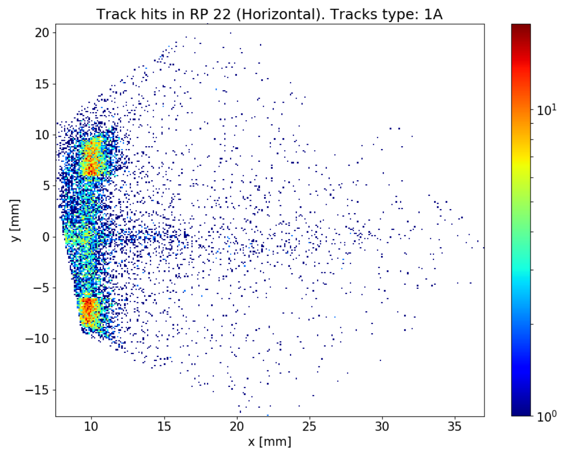

Figure 5.15: RP 22 hits. Track type 1A

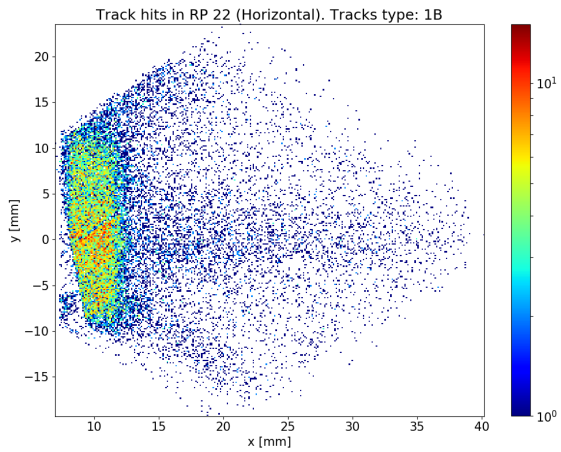

Figure 5.16: RP 22 hits. Track type 1B

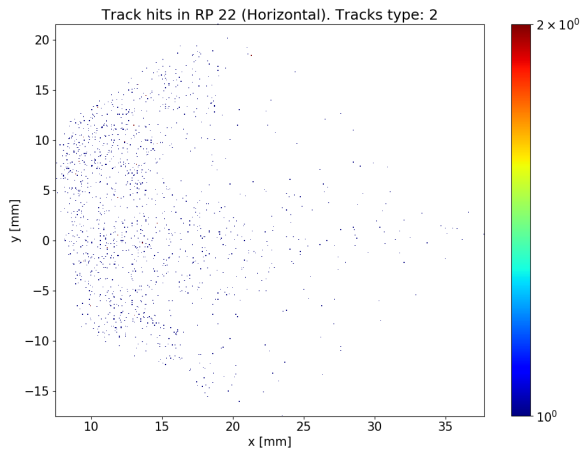

Figure 5.17: RP 22 hits. Track type 2

For all types of tracks, there are visible three most dense areas, first at the axis $y = 0$, and other two on the side edges of the detectors. This behaviour is characteristic for diffraction protons. There are also visible undesired tracks outside the detector for the type 1B, which indicates a problem in the code that generates it.

	Tracks [thous.]	Cut 1 [%]	Cut 2 [%]	Plotted tracks [thous.]	Plotted tracks [%]
All	132,3	6,251	56,285	49,6	37,465
Type 1A	32,1	3,507	52,338	14,2	44,155
Type 1B	71,7	7,709	49,957	30,4	42,334
Type 2	28,5	5,674	76,626	5,1	17,7

Table 5.6: Track type statistics for RP 22 (from Horizontal group on the right arm)

As we may have predicted the number of registered hits is significantly lower comparing to the vertical detectors. Remember that horizontal groups register about 12 times less hits than the vertical one. There are only 132 thousand of them.

Results are less accurate than in the vertical groups. Although higher limit for average distance from activated strips, more than half reconstructed tracks are rejected for type 1A and 1B. For type 2 only 17,7% tracks meet the correctness criteria. Probably the reason lies in the nature of the registered particles that originate from diffractive collisions and more often react with the material of the detectors, changing the path or hitting other particles.

Although the percent of valid tracks is only 37,5% we may easily verify them and get a distribution consistent with predictions.

5.5. Multitracking Statistics

In this section, we will analyze statistics for $2.59 * 10^7$ of reconstructed multiple tracks from fill 4499 run 9920. From the previous sections, we know that the algorithm reconstructed tracks consistent with detector physics and with predicted distribution for LHC optics $\beta* = 90m$ which were the first of the validation criteria.

Another criterion was to evaluate the reconstructed tracks by the χ^2 statistic defined in the section **??**

In addition to this statistics, we are also interested in what is the average distance between track and associated activated silicon strips in the detector.

At the beginning, we will analyze the cumulative results for all multitracking types. Then we will discuss each of them.

5.5.1. Overall multitracking statistics

The following figures depict χ^2, average distance between track and silicon strips and time of execution statistics for all types. Some statistics can be disturbed by different types of tracks. However, it is important to assess the general performance of the algorithm. What results can we expect and whether these results are satisfactory? Let us analyze it.

All histograms show results for $2.59 * 10^7$ reconstructed tracks. For the χ^2 histogram, we have applied 100 bins of the size = 1.0, for average distance between track and silicon strips, the bucket size was $\sigma = 0.0659mm$ and for the time of execution we used 100 bins of size $0.01s$. In the histograms of χ^2 and the average distance between track and activated silicon strips we can observe that the first bucket

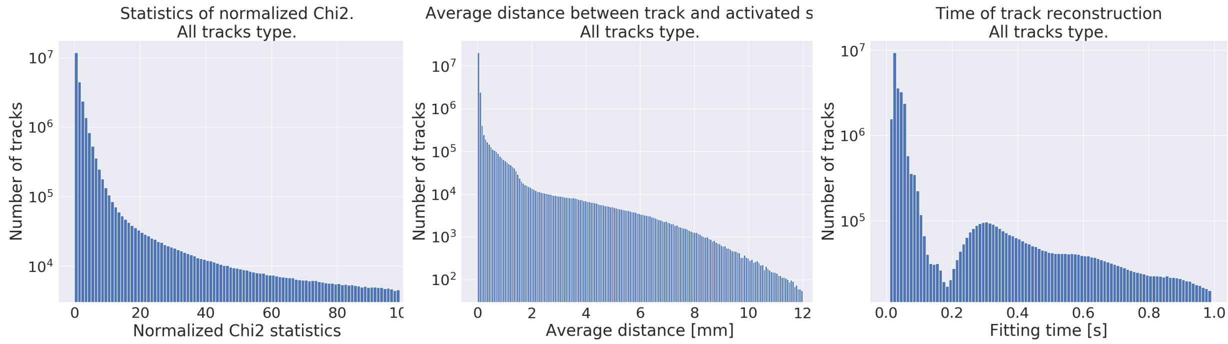

Figure 5.18: Cumulative statistics of normalized χ^2, average distance between track and silicon strips and time of execution statistics for all track types.

is the most numerous bucket, which proves the high quality of reconstructed tracks. We may notice that the histogram with the time of fitting phase is split into two parts. For time from $0.0s$ to $0.2s$ we register mainly tracks of type 1A and 1B with the number of performed fitting phase is equal respectively to 1 and 2. In contrast, the time $0.2s$ and larger, is mainly associated with the tracks of type 2, where the number of performed fitting is equal to 18 for one pair of tracks.

5.5.2. Multitracking statistics by track type

Finally we would like to compare which class of tracks is fitted with highest reliability. There is an intuition that the more complex case, the more difficult it is to perform a valid reconstruction. Let us investigate statistics for all of reconstructed track types.

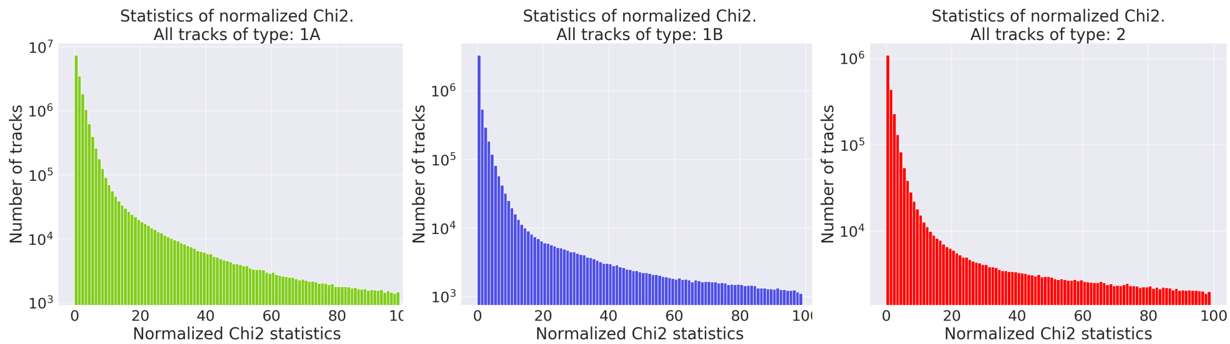

Figure 5.19: Compared track types by statistics of normalized χ^2.

In the Figure 5.19 we can see that the χ^2 statistic for each type adopts a visually similar distribution with particular emphasis on the first bucket, which for all types contains the highest percentage of reconstructed tracks. To find out the exact distribution of values, we analyze Table 5.7 from which we learn that over half of all tracks have $\chi^2 < 2.0$. For tracks of type 1A, 1B and 2 the χ^2 takes values less

			Tracks in first N bins		X bins contain N% of tracks			
Type	**Tracks [mln]**	**Bin size**	**1_bin**	**10_bin**	**50%**	**90%**	**95%**	**99%**
ALL	25,9	1	45,55%	85,98%	2	48	None	None
1A	16,7	1	44,24%	92,48%	2	7	24	None
1B	5,5	1	60,09%	85,03%	1	79	None	None
2	3,7	1	29,76%	57,93%	4	None	None	None

Table 5.7: Multitracking results. Comparison of χ^2 statistics for all track types.

than 10.0 respectively for 92.5%, 85.03% and 57.9% of all tracks of given type. Such a distribution of values indicates a good quality of reconstruction algorithm for all types. Although, as we expected, the weakest quality is observed for the reconstruction of tracks of type 2, but it is the most difficult case to reconstruct.

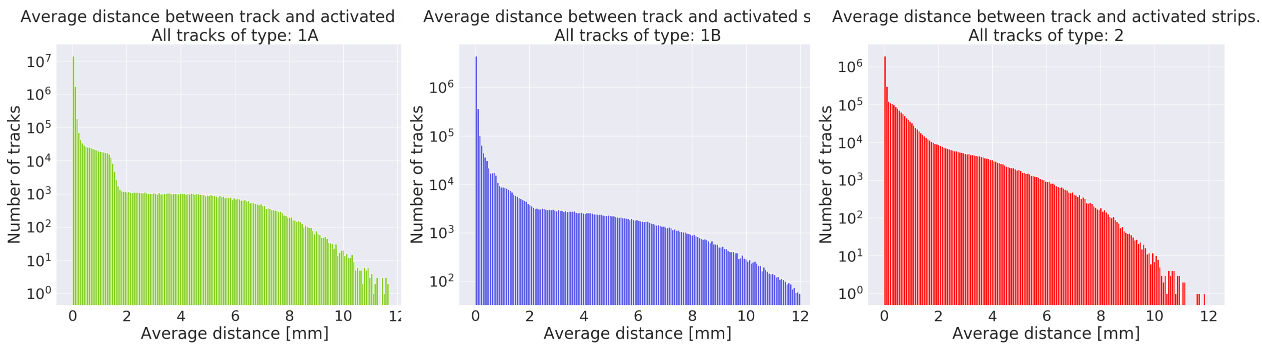

Figure 5.20: Compared track types by average distance between track and associated activated silicon strips.

			Tracks in first N bins		X bins contain N% of results			
Type	**Tracks [mln]**	**Bin size**	**1 bin**	**10 bins**	**50%**	**90%**	**95%**	**99%**
ALL	25,9	σ	79,41 %	94,5 %	1	3	12	60
1A	16,7	σ	85,02 %	98,05 %	1	2	2	18
1B	5,5	σ	80,47 %	93,15 %	1	4	20	96
2	3,7	σ	52,36 %	80,44 %	1	20	38	76

Table 5.8: Multitracking results. Comparison of statistics of average distance between track and activated silicon strips for all track types. Bucket size = $sigma = 0.0659mm$

Analyzing the average distance between tracks and activated silicon strips, it is worth noting that for each type the first bucket (of size $\sigma = 0.0659mm$) is the most numerous. This is an expected behavior. In the case of the reconstruction of type *2*, we can also observe that the first bucket contains

52.4% of all tracks. Let us remind that during this type of reconstruction we get a pair of tracks. It is a common situation where one track from a pair has a much better quality. However, to counteract high quality differences we used root mean square to choose the correct pair of tracks from all reconstructed combinations. Root mean square prefers a set of values with smaller deviations.

For types 1A and 1B, 85% and 80.5% of all tracks have been reconstructed with the average distance from activated strips equal σ. This is a very good result.

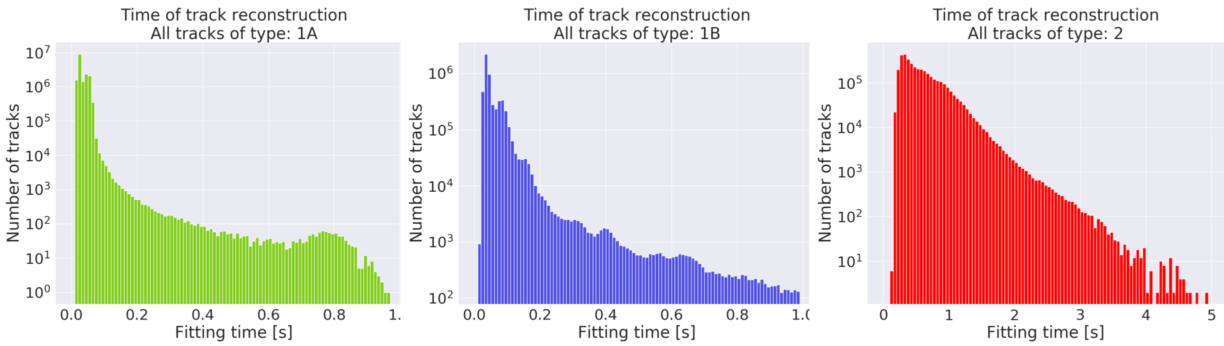

Figure 5.21: Compared track types by time of reconstruction.

				Fitting time of first N% tracks			
Track type	**Tracks [mln]**	**Bin size**	**In first 10 bins**	**50%**	**90%**	**95%**	**99%**
ALL	25,9	0.01	83.90 %	0.04 [s]	0.37 [s]	0.63 [s]	None
1A	16,7	0.01	99.86 %	0.03 [s]	0.06 [s]	0.06 [s]	0.07 [s]
1B	5,5	0.01	91.65 %	0.05 [s]	0.1 [s]	0.13 [s]	0.33 [s]
2	3,7	0.05	51.77 %	0.5 [s]	1.05 [s]	1.2 [s]	1.65 [s]

Table 5.9: Multitracking results. Comparison of fitting time for all track types.

Reconstruction time is closely related to the number of performed fitting phases. For subsequent types 1A, 1B, 2, we run 1, 2 and 18 fitting phases respectively. That is the reason why tracks with type 1A are reconstructed in 90% cases with the time less than $0.06s$, whereas those of type 2 are reconstructed in 90% with time less than $1.05s$.

It is also worth noting that the choice of the *u* and *v* lines has a big influence on the reconstruction time, for non-physical combinations the fitting phase of the algorithm can take considerably much more time.

Finally we would like to compare the average results for each of statistics per each type. To do this we will use values from the Table 5.10. The most numerous type of reconstructed tracks were those of type 1A - near $2,6 * 10^7$. They have the best statistics. Comparing reconstruction times we can observe that the reconstruction of tracks of type 1A lasts twice less than 1B and 18 times less than those with type 2. This reflects into the number of required fittings during reconstruction. For 1A - there is one fitting, type 1B - two fittings (one additional after the fishing phase) and for type 2 there are 18 fittings (for each

		χ^2		Average distance [mm]		Fitting time [s]	
Track Type	**Tracks[mln]**	**mean**	**stdev**	**mean [mm]**	**stdev**	**mean [s]**	**stdev**
ALL	25,9	891,36	83470,1	0,179	0,789	0,116	0,228
1A	16,7	173,7	8094,9	0,081	0,359	0,032	0,016
1B	5,5	2415,7	180137,2	0,28	1,345	0,061	0,084
2	3,7	1871,05	15727,4	0,473	0,976	0,577	0,325

Table 5.10: Multitracking statistics comparison for all track types.

u and *v* combination - 16, and after fishing - 2). An interesting observation is the highest mean value of χ^2 statistic for track with type 1B, which is most likely related to the error in the fishing phase.

Comparing the average track distances from the activated strips, we note that for tracks of type 1A the average value is $0.081mm = 1.22\sigma$, for tracks of type 1B $0.28mm = 4.25\sigma$, and for tracks of type 2 $0.473mm = 7.18\sigma$. Such distribution of values is consistent with predictions.

The results observed for the χ^2 statistics may be contrary to expectations. We would rather expects the mean value of the order of 10. However, let us note that some of the reconstructed tracks can give values of the order of 10^5 and larger, which significantly increases the average value of this statistics. Remember that according to Table 5.7 86% of all tracks have $\chi^2 < 10.0$, and 45.5% tracks have $\chi^2 < 1.0$. This is an order of magnitude for which the results can be considered, with high probability, to be correct.

6. Discussion

During the research we proved the posed thesis that the validated algorithm is not only useful for reconstruction of individual proton tracks but, most importantly, it is also effective and provides reliable results for multitracking phenomena.

In order to validate the algorithm we have implemented it, and compared single tracking results with those obtained by TOTEM experiment. Moreover, the tracks reconstructed in the case of multitracking provide us with an expected hit distribution.

We have applied following validation criteria:

– reconstructed tracks are consistent with LHC physics

– for single tracking the distribution of detector hits is consistent with the TOTEM experiment results

– for multitracking the distribution of detector hits satisfy predictions for $\beta* = 90m$ optics

– normalized χ^2 statistics of reconstructed tracks is lower or comparable with 1.0.

– average track distance from activated silicon strips is of the order of $1 * \sigma$

6.1. Result discussion

In this section we will answer the following questions. What has been achieved? What was expected? How our results differ from previous attempts to reconstruct multitracking tracks?

6.1.1. Achieved goals

We have implemented multitracking algorithm in python language using libraries for big data analysis, processing and visualization such as numpy, pandas and matplotlib. Moreover to perform reconstruction we have used scipy.optimize package. The implemented algorithm was tested on the data provided by TOTEM experiment from fills 4499, 4505, 4509, 4510, 4511.

We have analyzed over 6,5TB of data collected by detectors and partially processed by CMSSW modules. The multitracking algorithm was tested for data from fill 4499 of volume of 1,62TB.

According to Table 4.6 on page **??** there were over $4,8 * 10^8$ groups of Roman Pots for which we performed single tracking reconstruction. Furthermore, according to Table 4.7 on page **??** there were over $2,9 * 10^7$ groups of detectors where we reconstructed multitracking tracks. To process 1,62TB of data, stored in 1555 .root files, we have scaled our algorithm using Prometheus supercomputer. To process one .root file on one node we needed approximately 30 hours of computation time. During this time we performed full analysis, single and multitracking tracks reconstruction as well as the results comparison for single tracks between our results and those received using TOTEM projections *u* and *v*. The overall computation time was over $5*10^4$ hours. Most computations were performed simultaneously on 1000 nodes. In this thesis we showed results of multitracking reconstruction for data from fill 4499 where over $3,1*10^8$ events were registered.

It is worth to mention that we have discovered that many cases with multitracking phenomena may be easily simplified to single tracking by rejecting noises from recognized patterns of u and v lines. Hence we have specified three different types of multitracking presented in section 5.3.1

6.1.2. Expectations and results

We expected that reconstructed tracks in case of single tracking would be comparable with those computed by TOTEM software.

Indeed we have obtained similar hit distributions, consistent with LHC physics and satisfying predictions for $\beta* = 90m$ optics. According to Table 5.2.3 on page 51 3.06% of hit positions from vertical detector, and 12.72% from horizontal one, differed by no more than $1*\sigma(0.0659mm)$. Whereas, vast majority of tracks, namely 98.2% from vertical detector, and 84.6% from horizontal, differed by no more then 2σ. Over 74.4% of tracks had χ^2 statistics lower than 1, furthermore over 97.6% percent of tracks passed activated strips in distance lower than $0.1mm$. Such results proves that single tracking cases are reliably reconstructed.

Considering multitracking we will focus on hit distribution, statistics of normalized χ^2 and average distance between reconstructed tracks and activated silicon strips.

As we presented in section 5.3 the hit distribution is consistent with $\beta* = 90m$ optics for all multitracking track types. However some of reconstructed tracks were rejected. According to the Table 5.3.2 on page 55 and Table 5.4 on page 57 there were $3,6\%$ and $62,5\%$ of rejected tracks in vertical and horizontal detectors respectively. Such result is the effect of different interactions types registered in both detectors. The horizontal one registers mainly diffractive protons, hence they are more difficult to properly reconstruct. Though, we must remember that only 2.3% of all multitracking tracks were registered in horizontal Roman Pots. Therefore those tracks had little impact on the final results.

Let us remind that we have performed the reconstruction process for three different types of multitracking tracks. For tracks of the first type (*1A* - 64.5% of all tracks), the reconstruction problem was reduced to the reconstruction of a single track wiht only one fitting phase. Reconstruction of the tracks of type *1B* - 21.1% of all tracks - was reduced to the reconstruction of a single track with an additional fishing phase. Only tracks of type *2* - 14.3% of all tracks - were reconstructed using all three phases of the validated algorithm (combinatorics, fishing and fitting).

We have reconstructed 45.5% tracks with statistics of $\chi^2 < 1.0$ and 86% with $\chi^2 < 10.0$. The average distance between the track and the associated silicon strips was less than σ for over 79.4% cases. The average reconstruction time per track with the type *1A*, *1B* and *2* was $0.032s$, $0.061s$, $0.557s$

respectively, such proportions are consistent with the predictions and depend on the number of performed fitting phases (the most complex one).

The presented results prove that our algorithm is valid also for multitracking reconstructions.

6.1.3. Comparison with other algorithms

The TOTEM multitracking software is based on physical properties of elastic collisions utilizing principle of conservation of momentum. This limitation significantly narrows the class of tracks possible to reconstruct. Our algorithm relies only on detectors readouts and recognized u and v patterns, hence it does not have such constraint and makes it possible to reconstruct both of the proton types - elastic and diffractive as well.

6.2. Future work

After proving the correctness of the algorithm, the next logical step would be to implement it in C++ as a module in TOTEM offline software. It will increase efficiency of reconstruction process and will enable physicists to perform research on new class of registered proton-proton collisions.

In addition, in the next implementation, it is worth taking into account readouts from both vertical and horizontal detectors for protons passing the overlapping area. Such improvement could increase the accuracy of the reconstruction, as it would decrease the number of degree of freedoms.

Furthermore we could extend the algorithm to reconstruct more than 2 tracks per group. However we must remember about computational complexity of the combinatorics algorithm phase. Thus we would need a mechanism that would allow us to reject false combinations as soon as possible.

Bibliography

[1] Geoffray Adde, Belinda Chan, Dirk Duellmann, Xavier Espinal, Alessandro Fiorot, Jan Iven, Lukasz Janyst, Massimo Lamanna, Luca Mascetti, Joaquim M Pereira Rocha, et al. Latest evolution of eos filesystem. In *Journal of Physics: Conference Series*, volume 608, page 012009. IOP Publishing, 2015.

[2] G Antchev, P Aspell, I Atanassov, V Avati, J Baechler, V Berardi, M Berretti, M Bozzo, E Brücken, A Buzzo, et al. Proton-proton elastic scattering at the lhc energy of. *EPL (Europhysics Letters)*, 95(4):41001, 2011.

[3] Ilka Antcheva, Maarten Ballintijn, Bertrand Bellenot, Marek Biskup, Rene Brun, Nenad Buncic, Ph Canal, Diego Casadei, Olivier Couet, Valery Fine, et al. Root—a c++ framework for petabyte data storage, statistical analysis and visualization. *Computer Physics Communications*, 180(12):2499–2512, 2009.

[4] J Balcas, B Bockelman, D Hufnagel, K Hurtado Anampa, F Aftab Khan, K Larson, J Letts, J Marra da Silva, M Mascheroni, D Mason, et al. Stability and scalability of the cms global pool: Pushing htcondor and glideinwms to new limits. In *Journal of Physics: Conference Series*, volume 898, page 052031. IOP Publishing, 2017.

[5] B Bellenot, Ph Canal, O Couet, G Ganis, P Mato, L Moneta, A Naumann, and D Piparo. Root 6 and beyond: Tobject, c++ 14 and many cores. In *Journal of Physics: Conference Series*, volume 664, page 062006. IOP Publishing, 2015.

[6] Rene Brun and Fons Rademakers. Root—an object oriented data analysis framework. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 389(1-2):81–86, 1997.

[7] A-S Chen and P-F Shen. Computational complexity analysis of least-squares monte carlo (lsm) for pricing us derivatives. *Applied Economics Letters*, 10(4):223–229, 2003.

[8] Michele Barone Andrea Gaddi Larry Price Randal Ruchti Claude Leroy, Pier-Giorgio Rancoita. *Astroparticle, Particle And Space Physics, Detectors And Medical Physics Applications-Proceedings Of The 11th Conference On Icatpp-11*, volume 5. World Scientific, 2010.

[9] The TOTEM Collaboration. *Totem upgrade proposal, CERN-LHC-2013-009 / LHC-p-007.* Technical report, TOTEM, CERN, 2013.

[10] C Damerell. Applications of silicon detectors in high energy physics and astrophysics. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 226(1):26–33, 1984.

[11] EM Fajardo, JM Dost, B Holzman, T Tannenbaum, J Letts, A Tiradani, B Bockelman, J Frey, and D Mason. How much higher can htcondor fly? In *Journal of Physics: Conference Series*, volume 664, page 062014. IOP Publishing, 2015.

[12] H Grote. Pattern recognition in high-energy physics. *Reports on Progress in Physics*, 50(4):473, 1987.

[13] C Höppner, S Neubert, B Ketzer, and S Paul. A novel generic framework for track fitting in complex detector systems. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 620(2-3):518–525, 2010.

[14] Vincenzo Innocente, L Silvestris, D Stickland, et al. Cms software architecture: Software framework, services and persistency in high level trigger, reconstruction and analysis. *Computer Physics Communications*, 140(1-2):31–44, 2001.

[15] Frederick James. *Statistical methods in experimental physics*. World Scientific Publishing Company, 2006.

[16] Eric Jones, Travis Oliphant, and Pearu Peterson. {SciPy}: open source scientific tools for {Python}. 2014.

[17] Janusz Kacprzyk and Witold Pedrycz. *Springer handbook of computational intelligence*. Springer, 2015.

[18] Jan Kaspar. *Reconstruction of elastic events*. CERN-TOTEM-NOTE-2007-007 https://cds.cern.ch/record/1111157, Jun 2008.

[19] Jan Kaspar. *Elastic scattering at the LHC.* PhD thesis, Institute of Particle and Nuclear Physics, Charles University in Prague, 2011.

[20] Jan Kašpar. *Multi-track inefficiency of Si strip sensors used in Precision Proton Spectrometer*. CMS-TOTEM Precision Proton Spectrometer, CERN, 2018.

[21] C. Lefevre. *The CERN Accelerator Complex. Complexe des accelerateurs du CERN.* https://cds.cern.ch/record/1260465, Dec 2008.

[22] Rainer Mankel. *A Concurrent track evolution algorithm for pattern recognition in the HERA-B main tracking system*, volume 395. North-Holland, 1997.

[23] Ivan Markovsky and Sabine Van Huffel. Overview of total least-squares methods. *Signal processing*, 87(10):2283–2302, 2007.

[24] Wes McKinney. *Python for data analysis: Data wrangling with Pandas, NumPy, and IPython.* " O'Reilly Media, Inc.", 2012.

[25] Douglas C Montgomery, Elizabeth A Peck, and G Geoffrey Vining. *Introduction to linear regression analysis*, volume 821. John Wiley & Sons, 2012.

[26] Hubert Niewiadomski. *Reconstruction of protons in the TOTEM Roman Pot detectors at the LHC.* PhD thesis, The University of Manchester, 2008. Chapter 8.

[27] Daniel A Reed and Jack Dongarra. Exascale computing and big data. *Communications of the ACM*, 58(7):56–68, 2015.

[28] Jakub Sawicki and Jan Wójcik. *Proton reconstruction algorithm for track pattern recognition and fitting in the TOTEM experiment at CERN.* PhD thesis, AGH-UST, Cracow, 2015.

[29] Thomas Speer, W Adam, R Frühwirth, A Strandlie, T Todorov, and M Winkler. *Track reconstruction in the CMS tracker*, volume 559. Elsevier, 2006.

[30] Julian R Ullmann. *Pattern recognition techniques.* Butterworths, 1973.

[31] Stéfan van der Walt, S Chris Colbert, and Gael Varoquaux. The numpy array: a structure for efficient numerical computation. *Computing in Science & Engineering*, 13(2):22–30, 2011.

[32] Tzay Y Young and Thomas W Calvert. *Classification, estimation and pattern recognition.* North-Holland, 1974.