



UNIVERSITY OF BERGEN



Utrecht University



Optimization of the Track Reconstruction Algorithm in a Pixel Based Range Telescope for Proton Computed Tomography

Author: Alba García Santos

Delft, 15th of July 2019.

MSc thesis

In partial fulfillment of
the requirements for the

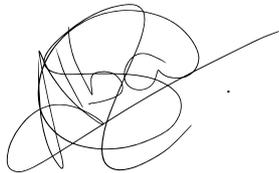
*Master of Science in
Biomedical Engineering*

Supervisors:

Danny Lathouwers
Helge Egil Seime Pettersen
Thomas Peitzmann

This Master Thesis project was completed in the Department of Medical Physics & Technology at the Delft University of Technology (TU Delft) in order to fulfill the degree requirements for a Master of Science in Biomedical Engineering in collaboration with Utrecht University and the University of Bergen. The project was conducted under the supervision of Danny Lathouwers, Dr., Helge Egil Seime Pettersen, Dr., and Thomas Peitzmann, Prof. Dr.

AUTHOR: ALBA GARCÍA SANTOS

A handwritten signature in black ink, consisting of several overlapping loops and a long horizontal stroke extending to the right. The signature is positioned above a solid horizontal line.

Acknowledgements

I would like to thank in these lines the help that many people and colleagues have given me during the research and drafting process of this work.

First of all I want to thank Thomas for accepting me to do this thesis in his department. I would also like to thank your willingness to attend to my questions despite your multiple occupations and your contributions and suggestions throughout this project.

My special thanks go also to Helge. Despite the distance, your support and confidence in my work and your ability to guide my ideas have been an invaluable contribution to the development of the thesis. I thank you also for your attentive and quick responses to the different concerns raised in the course of this work.

In a special way I am also grateful to Danny for his important contribution and active participation in the development of this thesis. There is no doubt that your participation has enriched the work done. The new ideas, always under your guidance and rigor, have been the key to good work.

I like to offer my warmest thanks to Naomi, for answering any question that came up and for your valuable contributions in the team meetings.

And, of course, thanks to Rene, for all those cafes, either inside or outside (when weather permitted) that allowed me to disconnect for a few minutes. Without being a direct part of my thesis, you have been a great help. Telling you about my issues during our breaks was not only good to blow off steam, but on many occasions you offered me valuable and useful solutions to continue when I was stuck. Thank you very much, you are a unique person, I doubt that regardless where I go or work, I find someone like you.

I also want to thank all the new friends in Delft that this Master has given me the opportunity to meet. Nor can I forget my friends from Madrid, Utrecht, Vienna, and those spread around the world. That either in person or by the amazing and useful recordings of WhatsApp, they supported me in this interesting, but sometimes complicated work. I cannot thank them enough for that.

And without a doubt, the best thing I get from the Master is a person who in a very short time has become someone fundamental and special to me. Silas, you are the person who supported me the most. Having nothing to do with my project, your help

has been fundamental, not only for your technical contributions and corrections, but for the emotional support that you have been giving me day after day. Thank you for advising me and suggesting solutions when I had very negative views of the project. I cannot wait to move to exotic Denmark and continue living amazing moments with you. Jeg elsker dig.

Y, por supuesto, el agradecimiento más profundo y sentido va para mi familia. Sin su apoyo, colaboración e inspiración habría sido imposible llevar a cabo este proyecto. A mis pares, Angela y Luis, por su ejemplo de lucha y superación. Muchas gracias por vuestro apoyo, por haber confiado en mí y por ayudarme a llegar a donde estoy hoy. Me siento la persona más afortunada por teneros y nada de esto habría sido posible sin vosotros. Y a mi hermana Lorena, por su personalidad, su manera de vivir, de hacerme reír como nadie. Eres la persona que más echo de menos, cada vez que voy a España no veo el momento de verte, abrazarte y tumbarnos en la cama para ponernos al día. Eres única. Os quiero.

Abstract

The use of protons to treat cancer has expanded rapidly in the past two decades. For safe and effective proton therapy, the proton range in a patient's body must be accurately determined. Current treatment planning is based on X-ray computed tomography images, which might cause uncertainty because of the different behaviour between protons and X-rays. As an alternative, proton Computed Tomography (pCT) has been proposed to directly measure the Relative Stopping Power (RSP) map in the patient and reduce this uncertainty. During a proton CT scan, a high-energy proton beam is directed at the patient. Then, the proton's residual energy and position are measured with a detector placed behind the patient. This information is used to calculate the volumetric RSP. In the case of using a pixel based detector, a tracking algorithm is required in order to increase the proton intensity capacity of the detector.

A proton track reconstruction system has been already developed by Pettersen [1], however, it has some limitations on the track density that can be reconstructed correctly. The algorithm is based on the track-following scheme, in which a growing track searches for deeper-lying activated pixels.

This thesis introduces proton therapy and the advantages of pCT and proton radiography for treatment planning. Then, the main track reconstruction techniques found in the literature are reviewed. Improvements in the reconstruction process are proposed and their efficiencies are discussed. While current algorithm begins from the layer closest to the patient, in the present study a new reconstruction algorithm is developed. It differs by starting the reconstruction process from the distal end of the detector. Based on this new algorithm, studies related to its optimization are conducted. Lastly, an algorithm based on the identification of the most probable scenario is developed.

The potential algorithms are evaluated on data simulated with GATE (based on Monte Carlo interactions) and PROCASIM (design to simplify the physical interactions between protons and the detector). The fraction of correctly reconstructed tracks and the computational efficiency of the algorithms are analyzed to determine the most viable one.

Keywords: proton Computed Tomography, tracking algorithm, Digital Tracking Calorimeter

Nomenclature

<i>ALPIDE</i>	ALice P <i>IX</i> el D <i>ET</i> ector
<i>ATLAS</i>	A Toroidal LHC Apparatu <i>S</i>
<i>CBCT</i>	Cone Beam Computed Tomography
<i>CERN</i>	Conseil Européen pour la Recherche Nucléaire
<i>CKF</i>	Combinatorial Kalman Filter
<i>CNN</i>	Convolutional neural networks
<i>CPU</i>	Central Processing Unit
<i>CT</i>	Computed Tomography
<i>DAF</i>	Deterministic Annealing Filter
<i>DTC</i>	Digital Tracking Calorimeter
<i>EAA</i>	Elastic Arms algorithm
<i>GATE</i>	Geant4 Application for Emission Tomography
<i>Geant4</i>	GEometry ANd Tracking 4
<i>GSF</i>	Gaussian-Sum Filter
<i>HL – LHC</i>	High Luminosity Large Hadron Collider
<i>HT</i>	Hough transform
<i>HU</i>	Hounsfield Units
<i>KF</i>	Kalman Filter
<i>LHC</i>	Large Hadron Collider
<i>LSTM</i>	Long Short Term Memory

LUT Look Up Table
MC Monte Carlo
MCS Multiple Coulomb Scattering
NN Neural Networks
pCT proton Computed Tomography
PRaVDA Proton Radiotherapy Verification and Dosimetry Applications
PROCASIM PRoton CALorimeter SIMulator
PTCOG The Particle Therapy Cooperative Group
RAM Random-Access Memory
RMS Root Mean Square
RNN Recurrent Neural Network
ROOT Object-oriented data analysis software package written at CERN
RSP Relative Stopping Power
WEPL Water-Equivalent Path Length

Contents

1	Introduction	1
1.1	Treatment accuracy	3
1.2	Interactions of protons with matter	5
1.2.1	Inelastic Coulomb interactions	5
1.2.2	Multiple Coulomb Scattering	6
1.2.3	Inelastic nuclear interactions	6
1.3	Research Objectives	7
1.4	Outline of the manuscript	7
2	Technical background	9
2.1	Proton CT concept	9
2.1.1	Detector design in proton CT project	10
2.2	Simulation of a proton radiographic image	11
2.2.1	Monte Carlo Simulations	11
2.2.1.1	GEANT4	12
2.2.1.2	GATE	12
2.2.1.3	Scanner description	13
2.2.1.4	Output data	13
2.2.2	Simulations using PROCASIM	14
2.2.2.1	Parameterization of the cone	16
2.3	Current tracking algorithm	19
3	State of the art	23
3.1	Reconstruction Algorithms	24
3.1.1	Classical Methods	24
3.1.1.1	Track finding	24
3.1.1.1.1	Global Methods	24
3.1.1.1.2	Local Methods	27
3.1.1.2	Track fitting	28
3.1.1.2.1	Kalman Filter	29
3.1.2	Adaptive Methods	33
3.1.2.1	Neural Networks	33
3.1.2.1.1	Recurrent Neural Networks	33

3.1.2.1.2	Convolutional Neural Networks	35
3.1.2.2	Gaussian-Sum Filter	35
3.1.2.3	Deterministic Annealing Filter	36
3.1.2.4	Elastic Nets and Deformable Templates	37
3.1.3	Other Methods	37
3.2	Discussion	38
3.3	Conclusions	39
4	Improved reconstruction algorithms	40
4.1	Track reconstruction quality	40
4.2	Last layer starting algorithm	40
4.3	S_{\max} value optimization	44
4.3.1	Constant S_{\max}	44
4.3.2	Proton energy dependent S_{\max}	50
4.3.2.1	Theoretical MCS	50
4.3.2.2	Measured MCS from simulated data	51
4.3.2.3	Comparison between theoretical and measured MCS values	53
4.3.2.4	Track reconstruction	53
4.4	Robustness of the reconstruction algorithm	60
4.5	Proton beam width in the detector	61
5	Most probable scenario algorithm	64
5.1	Optimized version	66
5.2	Adaptation to Monte Carlo simulations	70
5.3	Discussion and conclusions	73
6	Discussion and conclusions	75
7	Limitations and recommendations for future work	77

List of Figures

1.1	The depth dose curve in water for 107 MeV protons [2]. The deposited dose is low at the entrance region, and then increases rapidly towards the end of the particle range, and then a maximum dose deposition in the Bragg Peak before the protons come to rest.	2
1.2	Comparison plans of Photon and Proton Beam Therapy for the treatment of distal esophageal cancer. Note the relative amount of normal tissue spared of scattered doses of radiation in the proton plan compared to the photon plan [3].	2
1.3	Range uncertainties during particle therapy treatment. The main source of uncertainty is introduced by variation in patient anatomy and range degradation, followed by the use of CT images in the treatment planning. Then, the error introduced by patient positioning and finally, the uncertainties produced by fluctuations of the beam [4]. . .	4
1.4	Schematic illustration of proton interaction mechanisms: (A) energy loss via inelastic Coulombic interactions, (B) deflection of proton trajectory by repulsive Coulomb elastic scattering with nucleus, (C) removal of primary proton and creation of secondary particles via inelastic nuclear interaction (p: proton, e: electron, n: neutron, γ : gamma rays) [5].	5
1.5	The linear stopping power of protons in water [2].	6
2.1	Proton CT imaging concept. The trajectory of each particle crossing an object is measured using two trackers (after and before the object). The residual energy of the particle is measured using a detector. . . .	9
2.2	A close-up of two layers, consisting of the different components that approximate the current planned layer design. To fully slow down and stop a 230 MeV proton, 25 - 70 layers will be necessary depending on size of the Al absorber. If it's fixed to 3.5 mm, 41 layers are needed. . .	11
2.3	Schematics of proposed prototype. In order to obtain a spectrum of different proton beams to hit the DTC, the thickness of the energy degrading water phantom is modulated from 0 cm to the maximum range of a 250 MeV beam [6].	13

2.4	Distribution of protons on the detector surface at 20 cm distance from the water phantom. Yellow color represent protons that only experienced MCS. Blue and red dots reflect protons that have undergone elastic/inelastic nuclear reactions, respectively. The data was generated using GATE with the full physics list [7].	15
2.5	Schematic of the ellipse created by the intersection between the cone and the next layer of the detector. The size of the cone is calculated by the selection of an angle, β , in the Gaussian distribution with $\sigma = \theta_0$. Then, the value of $\varphi \in [0, 2\pi)$ is chosen randomly.	16
2.6	The intersection between the cone $C(0,0;\alpha,\delta,\theta)$ and a horizontal plane with $z \geq 0$ is an ellipse with its major axis parallel to the x-axis and its minor axis parallel to the y-axis [8].	16
2.7	The cone's axis has an arbitrary direction described by the angular parameters α and δ . The cone's opening angle is β . The intersection between the cone and a horizontal plane with $z > 0$ is an ellipse $E(z;\alpha,\delta,\beta)$. The major axis of the ellipse forms the angle δ with the 0x axis [8].	18
2.8	Schematic of the tracking algorithm steps. Hits in the different layers of the detector are represented by black stars. Starting from the first layer of the detector, the first hit (in yellow) of a track is selected (A). Then the next candidates are search using a cone defined by $S_{\max}(B)$. If calculated S_n using the new candidate is lower than S_{\max} , this hit is added to the track (C). If more that one hit is identified, the candidate with lowest S_n is added (D). These steps are repeated in the following layers (E) until the last layer of the detector is reached (F). Subsequently, hits belonging to this track are removed from the search pool and the reconstruction of the next track starts (G). The algorithm finishes when all the identified tracks are reconstructed (H).	20
2.9	Example of track reconstruction. Black lines represent tracks that have been correctly reconstructed while gray lines are tracks that are well reconstructed but are not complete. Incorrectly reconstructed tracks are visualized in red. Green lines show reconstructed secondary particles. Blue dost correspond to unused hits.	21
2.10	Efficiency of the current algorithm for different proton densities. In order to compare results for various energies, a 250 MeV pencil beam is degraded with a water phantom of different thicknesses (from 5 cm to 34 cm). Efficiency is measured as the fraction of tracks in which all cluster have the same eventID (whole track correct graph) and the fraction of tracks where the first and last cluster has the same eventID (correct endpoint graph). Solid lines reflect only correctly and fully reconstructed tracks, while dashed lines indicate the percentage of all well-reconstructed tracks, including the ones that are not complete. Each reconstruction is iterated 50 times.	22

3.1	Representation of the track finding stage. Red dots correspond to the hits produced by particles in the different layers of the detector while they travel through it. Hits that are considered to originate from the same particle are assigned to an specific subset. Each subset, represented with different colors, corresponds to a different tracks.	25
3.2	Steps in Hough Transform: A track built from different hits produced probably by the same particle is drawn in (A). This track corresponds to a dot in the parameter space (B). However, in reality, many tracks can be drawn through each hit (C). Each track in (C) represents a dot in the parameter space (D). In (F) each curve of a color represents the infinite number of tracks that go through the hit of the same color in (E). The blue point in (F) indicates the track (represented in (E)) that connect all the hits in the pattern space.	26
3.3	Representation of the track fitting stage. Red dots correspond to the hits produced by particles in the different layers of the detector while they travel through it. Hits are already assigned to a specific track in the track finding stage. However, the tracks needs to be parametrized using the selected hits (black curved lines).	29
3.4	Kalman Filter algorithm: starting from the first layer, a new hit is added to an specific track during the prediction step (green arrows). Afterwards, a recalculation of the estimated track parameters is executed in the filter step (orange arrows). Both processes are repeated until the last layer is reached. The triangular shadows surrounding the arrows reflect the uncertainty at each step. As more measured are added to the track, the uncertainly in the estimations is reduced, and therefore the shadows that surrounds the arrows are increasingly narrow.	31
3.5	Kalman filter-smoother algorithm: it start with the Kalman filter explained in Fig. 3.4 (orange arrows). Then, a smoothing step (purple arrows) runs in the opposite direction and using the results of the KF. Using the information of both filters, a better estimation of the track parameters can be calculated. The triangular shadows surrounding the arrows reflect the uncertainty at each step. The uncertainly in the estimations is reduced as the number of measurements increases, and therefore the shadows surroundings the arrows are increasingly narrow.	32
3.6	A neural network is an interconnected group of nodes. Each circular node represents an artificial neuron and an arrow represents a connection from the output of one artificial neuron to the input of another. Each connection has assigned a weight (W_{ij}) that control the strength of the signal.	34

3.7	Schematic of the most used Neural Networks: (A) Hopfield Neural Network: all neurons are both input and output neurons. The network iterates to a stable state, and the output of the network consists of the new activation values of the neurons. (B) Long Short Term Memory network: apart from the input and output cells, memory cells are also incorporated into hidden layers neurons. It remembers previous values and regulate the flow of information into and out of the network taking into account previous values. (C) Convolutional Neural Network: it consists of an input, an output and multiple hidden convolutional layers. These convolutional layers apply a convolution operation to the input passing the result to the next layer.	36
4.1	Efficiency of the last layer starting algorithm as a function of proton density for different beam energies. Data simulated with GATE. The definition of correctly reconstructed track is given in Section 4.1. . . .	42
4.2	Efficiency of the last layer starting algorithm as a function of proton density for different beam energies. Data simulated with PROCASIM. The definition of correctly reconstructed track is given in Section 4.1.	43
4.3	Distribution of S_n values in each layer. Red line represents the 2σ value of the distribution. Two different S_{\max} are shown. From [6]. . .	44
4.4	Efficiency of last layer starting algorithm depending on the substituted values in Eq. 4.1 for a energy pencil beam of 250 MeV degraded by 25 cm of water, resulting in a mean energy of 133.96 MeV. The graphs represent the fraction of tracks in which its start- and endpoints have the same eventID and are fully tracked.	46
4.5	Efficiency of the last layer starting algorithm depending on the substituted values in Eq. 4.1 when a proton beam of energy 250 MeV is degraded by 5 cm of water, resulting in a mean energy of 229.92 MeV. The graphs represent the fraction of tracks in which the first and last cluster added have the same eventID and are fully tracked.	47
4.6	Efficiency when $S_{\max}(n_p) = 0.047n_p^{-0.176}$ rad is applied for the reconstruction of a pencil beam degraded with a water phantom of various thickness, resulting in different energies. Data simulated with GATE. The definition of correctly reconstructed track is given in Section 4.1. .	49
4.7	The RMS angle (in rad) of Highland's approximation (Eq. 1.2) depending on the energy of the proton (in MeV).	50
4.8	Distribution of angular change at different layers from reconstructed tracks. Data were simulated with GATE with a 250 MeV beam modulated by a water phantom of 5 cm (229.93 MeV) and 11 cm (204.33 MeV). The red line represent the empirical 4σ value of the distribution at each layer.	51
4.9	Left: Fitted curve (in blue) and 4σ curve (in red) over the 2D histogram of distribution of the angular difference between layers. Right: Fitted curves for different number of protons.	52
4.10	Fitted curves for different energies.	52

4.11	Comparison between the MCS values calculated following Highland's equation (in red) and the ones measured from simulated data (in blue) for two different energies.	53
4.12	Reconstruction of 100 simulated track when a 250 MeV proton beam is degraded by water phantom of 5 cm thick and the value of S_{\max} is based on: (A) Highland's formula, (B) four times Highland's formula and (C) equation explained in Section 4.3.1 (constant value).	54
4.13	Reconstruction of MC simulated track of energy 229.92 MeV. The proton enters from left side (layer 0) and continues until it stops (layer 37). At layer 17, one scatter interaction happens and changes the direction of the proton by some angle. When option (ii) is used to define the search angle, the track is not fully tracked (left). On the contrary, when choice (iii) is applied, the track is correctly reconstructed (right).	56
4.14	Value of search angle is calculated following Highland's formula for different energies (red curve) and two variations of it (blue and green curves).	57
4.15	Efficiency when variable value of S_{\max} is applied for the reconstruction of a pencil beam degraded with a water phantom of various thickness, resulting in different energies. Data simulated with PROCASIM. The definition of correctly reconstructed track is given in Section 4.1.	58
4.16	Efficiency results when the value of S_{\max} changes throughout the reconstruction of the track depending on the expected scatter. In order to measure the efficiency for various energies, a pencil beam is degraded with a water phantom of different lengths. Data simulated with GATE. The definition of a correctly reconstructed track is given in Section 4.1.	59
4.17	Track reconstructed results when the algorithm starts from the first seed it finds (left) and when the last identified seed is established as the origin of the reconstruction process (right). Blue arrows indicate the staring seed for each scenario.	60
4.18	Examples of track reconstructions. (A) A water phantom of different thicknesses is used to degrade a 250 MeV proton beam to get a mean energy of ~ 234 MeV (5 cm degrader thickness) and ~ 69 MeV (34 cm degrader thickness). (B) Protons are simulated directly with the intended energies. Data simulated with GATE.	62
4.19	Fraction of correctly reconstructed tracks following the definitions in Section 4.1 when a proton beam of energy 250 MeV is degraded by water phantom of different thickness (with degrader) and when protons are simulated directly with the desired energy (without degrader). In both scenarios, tracks simulated with GATE are reconstructed using the algorithm described in Section 4.2.	63

5.1	Example of the most probable scenario algorithm. Given 3 clusters in a layer and 3 clusters in the next layer, all possible combinations are studied, in this case, 6 different scenarios, represented by different colors.	64
5.2	Schematic of how the angular deflection θ between two clusters located in consecutive layers is calculated for the general case (left) and when θ is computed between a seed in the first layer and a cluster in the next one, where \vec{u} is assumed to be perpendicular to the detector plane (right).	65
5.3	Two hits in consecutive layers are represented in pink. The four corners of the pixel in which the second cluster was detected are represented as yellow stars in (A). The angular deflection, θ , is the difference between initial direction of the first hit and the angle between first hit and second hit. The smallest value is assigned to θ_{\min} (B) while the largest one to θ_{\max} (C).	66
5.4	Track reconstruction result when: algorithm explained in Section 4.2 is used (left) and when most probable scenario algorithm (both original and optimized version) is applied (right). In both studies, 8 tracks are simulated using PROCASIM.	67
5.5	Flowchart of the most probable scenario algorithm. Red arrow shows the optimized version path (Section 5.1). Green arrow points the path to the adapted version to MC simulations (Section 5.2). ¹ This step is described in Fig. 5.1. ² Following Eq. 5.3. ³ This step is described in Fig. 5.9.	68
5.6	Fraction of tracks whose start- and endpoints have the same eventID for different proton densities when the most probable scenario is calculated before assigning each cluster to the respective track. Data simulated with PROCASIM.	69
5.7	Example of a type of error in the reconstruction process due to the proximity of two tracks. Subfigures (A) and (B) show the same reconstruction result from two different points of view. Subfigures (C) and (D) show the zoomed image of the area selected in Subfigures (A) and (B) with green and yellow frames respectively. The two tracks that introduce confusion in the reconstruction process are marked with blue arrows. As can be seen, the tracks are so close to each other that it is difficult to differentiate them in most of their trajectory.	70
5.8	Examples of the reconstruction result when a energy pencil beam of 250 MeV is degraded by 5 cm (left) and 16 cm (right) of water. During the reconstruction process, the algorithm has to deal with single hits (marked with red arrows) that can introduce errors and cause tracks to be reconstructed erroneously.	71

5.9	Schematic of the most probable seed-cluster selection process. In case the same cluster is assigned to multiple hits in the prior layer, different probabilities of the whole scenario (e.g. all combination between seeds-clusters) are calculated combining each time the shared cluster with one of the hits, while the rest are not linked to any cluster. In (A) a cluster (black star in red frame) is shared by two hits in the previous layer (yellow and blue stars in red frame), so the probability of two different scenarios are calculated. The same process is repeated in (B), where two clusters (black stars in red frames) match with more than one seed in the prior layer (yellow and blue stars in red frames). In this scenario, the probability of four different combinations are computed. In the end, the most likely combination is chosen.	72
5.10	Study of the fraction of tracks whose start- and endpoints have the same eventID for different proton densities and energies. Data simulated with GATE.	73

List of Tables

4.1	Percentage of correctly reconstructed tracks depending on the way S_{\max} is calculated.	55
4.2	Percentage of correctly reconstructed tracks depending on the starting seed order.	60

Chapter 1

Introduction

Cancer is a large family of diseases that involve abnormal cell growth with the potential to invade or spread to other parts of the body. Cancer is the second leading cause of death globally, and is responsible for an estimated 9.6 million deaths in 2018, causing about 1 in 6 deaths [9]. Currently, the three main therapies for cancer are radical surgery (removing the tumor), radiotherapy (killing the tumor cells with radiation), and chemotherapy (the use of anti-cancer drugs). The type of treatment depends upon the location and grade of the tumor and the stage of the disease, as well as the general state of the patient.

In the last two decades, radiotherapy using ions, commonly referred to as particle therapy, has become increasingly popular to treat cancer patients. The Particle Therapy Cooperative Group (PTCOG) reported that, by the end of 2017, 199 845 patients had been treated with particles. The potential of using these particles in radiation therapy was first proposed by Wilson (1946) after discovering that protons and heavy ions deposit a maximum dose, also called the Bragg peak, at a precise depth in a tissue as shown in Fig. 1.1.

The most common form of particle therapy is performed with protons and carbon ions and have several advantages over traditional radiotherapy using photons [10]. The primary advantage is a superior dose distribution, as particle therapy irradiates a smaller volume of healthy tissue during treatment as can be seen in Fig. 1.2. This is due to the finite range of ions in matter. By carefully selecting the energy of the incoming ions, such that they stop inside the area to be irradiated, no dose is delivered to the tissue downstream from the target. This has two implications. First of all, a lower dose to the healthy tissues leads to fewer side effects of the treatment. Secondly, since the dose to tumour is often limited by what is tolerated by the normal tissue, the target dose can be increased leading to better tumour control for some tumours.

The advantages of particle therapy translate into a significantly reduced number of patients suffering from side effects from their radiation treatment cure [11]. Among these are a reduction of the number of patients developing secondary cancers from

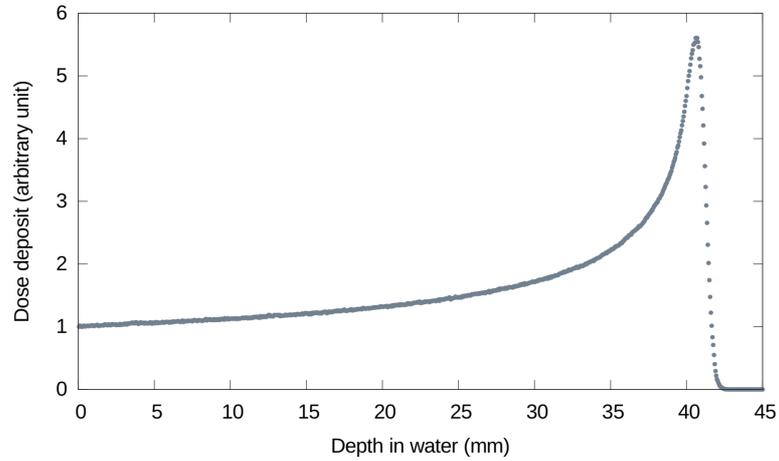


Figure 1.1: The depth dose curve in water for 107 MeV protons [2]. The deposited dose is low at the entrance region, and then increases rapidly towards the end of the particle range, and then a maximum dose deposition in the Bragg Peak before the protons come to rest.

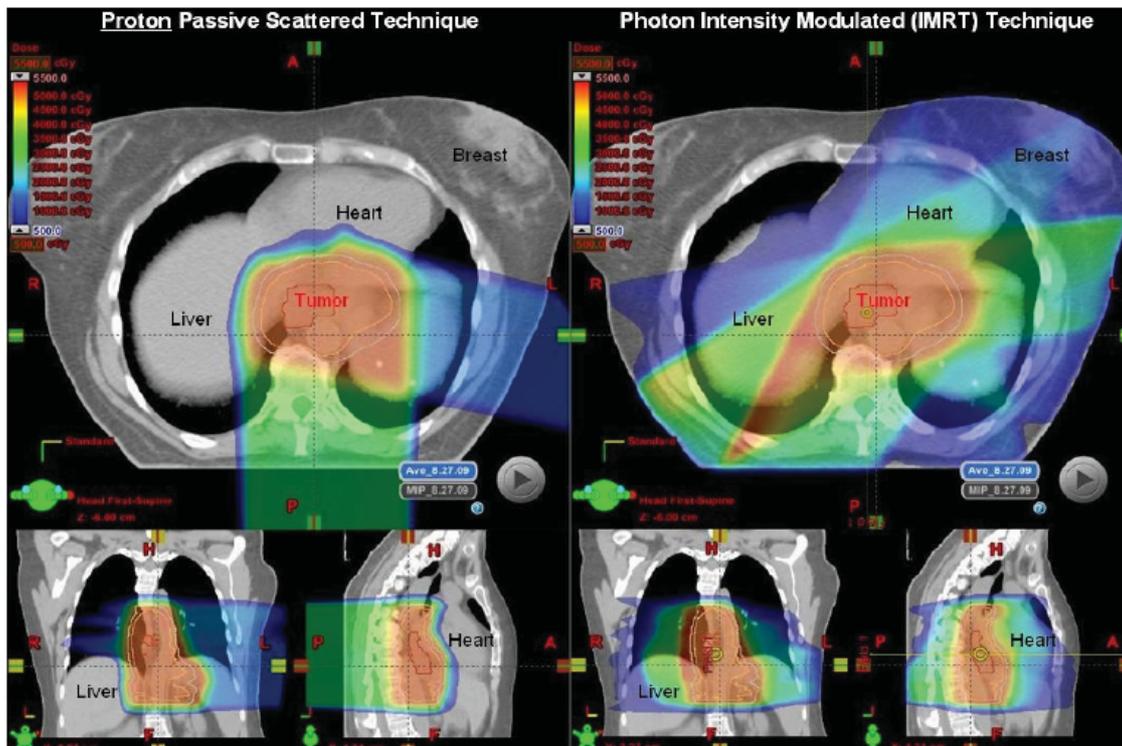


Figure 1.2: Comparison plans of Photon and Proton Beam Therapy for the treatment of distal esophageal cancer. Note the relative amount of normal tissue spared of scattered doses of radiation in the proton plan compared to the photon plan [3].

the irradiation, a reduction of damage to normal tissue and less damage to nearby organs [1]. In addition, pediatric patients could also benefit from this procedure as it could limit the radiation exposure to healthy and developing tissues [12, 13].

1.1 Treatment accuracy

In particle therapy, one of the main concerns is the impact of range uncertainties on tumour coverage. Due to the precise dose delivery, an error in the range calculation could potentially cause the treatment beam to miss the distal part of a tumour or irradiate more healthy tissue downstream of the tumour. Fig. 1.3 shows the main causes of range uncertainties, where the less predominant effect is on the top and the main one at the bottom. The pyramid is topped by the uncertainties produced by beam fluctuations (± 0.7 mm) [14], followed by the error introduced by variation in the patient positioning (± 0.7 mm) [14]. Below the uncertainties produced by the use of Computed Tomography (CT) images to calculate the dose plans ($\pm 2.5\%$ - 5% [15] + up to 18% [16]). At the bottom of the pyramid can be found the errors introduced by variation in the patient anatomy during the course of the treatment planning ($\pm 2.5\%$ up to ∞) [17]. Range uncertainties are unique to heavy charged-particle therapy and require additional margins along the beam path (whereas uncertainties in conventional therapy can be considered as isotropic expansion of the volume to be treated). Clinically, safety margins around the treatment volume are introduced in order to ensure tumour coverage—i.e. of the order of 3.5% of the prescribed range plus an additional millimetre [18]. This adds a substantial dose to healthy tissue. Therefore, proton therapy has to be evaluated not just in terms of the success of the irradiation in destroying the cancer, but most importantly in terms of how well the effects of the radiation on the rest of the body are minimized [19].

As mentioned before, nowadays, dose plans in proton therapy are based on CT images. The CT images are reconstructed based upon photon interaction with matter, thus a conversion is required for calculating the Relative Stopping Power (RSP) for how the protons traverse and deposit dose in the patient's body during proton therapy. A problem with this approach is that x-rays interact very differently with materials compared with protons, resulting in relations between Hounsfield Units (HU) and RSP that are not unique (and can therefore be ambiguous) [19]. This conversion procedure introduces range uncertainties typically of the order of 2.5% – 5% [15], corresponding to $5 - 10$ mm at a treatment depth 20 cm into the patient [2]. On the other hand, Dual Energy CT can further reduce these uncertainties by at least 0.4% [20]. It combines information from two images made using different x-ray energies in order to resolve the ambiguities in the HU to RSP conversion [19]. However, resolution of CT images, including possible x-ray CT artifacts, can also affect the accuracy of reconstructed RSP maps [21, 16].

Proton CT offers a potential solution to this discrepancy as it directly maps the RSP of the particles in the patient, which will allow a reduction of treatment margins [22]. While protons of therapeutic energies are stopped in the body, during a pCT

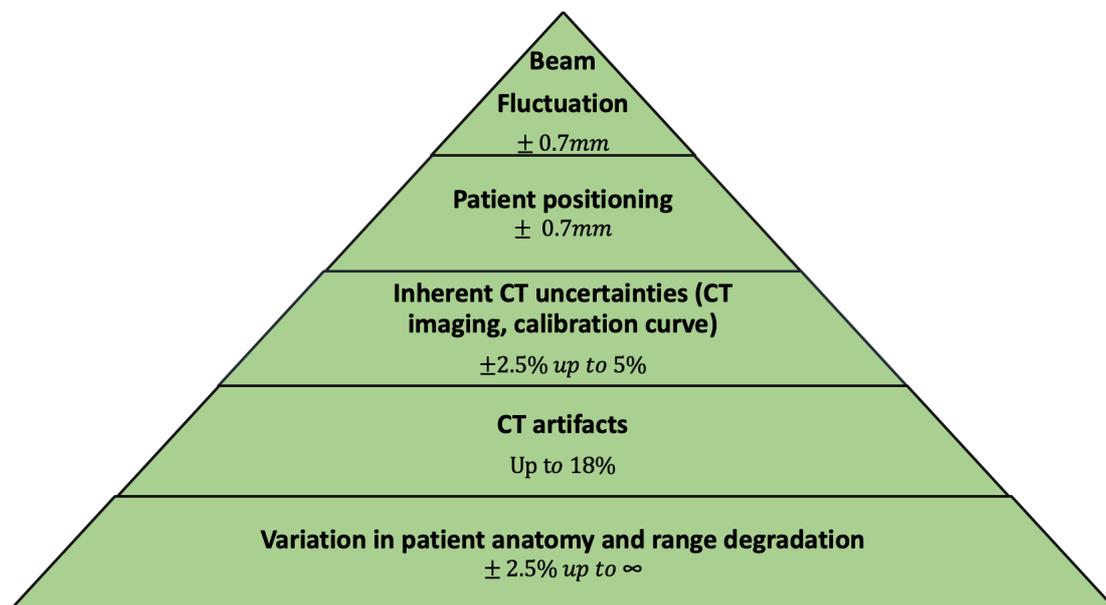


Figure 1.3: Range uncertainties during particle therapy treatment. The main source of uncertainty is introduced by variation in patient anatomy and range degradation, followed by the use of CT images in the treatment planning. Then, the error introduced by patient positioning and finally, the uncertainties produced by fluctuations of the beam [4].

scan, a high-energy proton beam is directed at the patient and the proton beam must have sufficient energy to completely pass through the patient being imaged. Then, the protons' residual energies are measured in a detector behind the patient. The information about the residual energy from each proton can then be used, together with the proton's estimated path through the patient, as a basis for reconstructing a volumetric RSP. This RSP map can be used directly in a treatment planning system [1].

Arbor et al. [21] showed that RSP reconstruction and proton range calculations based on proton CT are more precise and more uniform, allowing a reduction of the treatment margins used to plan proton therapy. Moreover, early on, it was realized that proton radiography potentially provides better density resolution and tissue contrast, compared to conventional x-ray imaging [23], and more recent work appears to confirm it [24]. Proton radiography could also provide quick verification of patient setup in a cancer treatment facility, with very low radiological dose, in addition to a real-time monitoring of the treatment beam [25, 26]. Also, proton CT does not suffer from artifacts that often appear in reconstructions of x-ray CT scans [19]. Another potential benefit is the reduction of imaging doses compared to x-ray CT and Cone Beam Computed Tomography (CBCT) [21].

1.2 Interactions of protons with matter

In general, proton interactions with matter can be divided into three main groups in the therapeutic energy range under consideration:

1.2.1 Inelastic Coulomb interactions

Protons continuously lose kinetic energy via frequent inelastic collisions with atoms (Subfigure (A) in Fig. 1.4). In the energy range considered for proton imaging (typically between 20 and 300 MeV), the energy loss process is dominated by interaction of the proton with the bound outer-shell electrons of atoms in the matter penetrated. Protons have a mass which is large compared to the mass of the electrons, so only a small fraction of the proton's energy is lost in a single interaction, however, any deflection of the protons can be neglected. The mean energy loss per distance travelled, also called stopping power S , is well-described by the Bethe theory [27].

For a particle with speed v , charge z , and energy E , travelling a distance x into a target of electron number density n and mean excitation potential I , the relativistic version of the formula reads, in SI units:

$$-\left\langle \frac{dE}{dx} \right\rangle = \frac{4\pi}{m_e c^2} \cdot \frac{nz^2}{\beta^2} \cdot \left(\frac{e^2}{4\pi\epsilon_0} \right)^2 \cdot \left[\ln \left(\frac{2m_e c^2 \beta^2}{I \cdot (1 - \beta^2)} \right) - \beta^2 \right] \quad (1.1)$$

where c is the speed of light and ϵ_0 the vacuum permittivity, $\beta = \frac{v}{c}$, e and m_e the electron charge and rest mass respectively.

The electronic stopping power of protons in water is represented in Fig. 1.5. Through Eq. 1.1, one can see what leads to the peculiar shape of the Bragg curve, that characterizes the stopping power of the material as a function of the depth, and particularly the peak (Bragg peak) for low values of β at the end of the range, as can be seen of Fig. 1.1 [2].

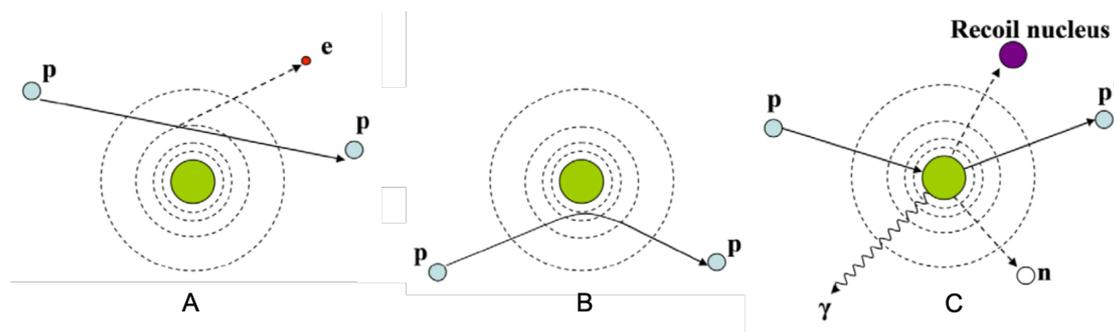


Figure 1.4: Schematic illustration of proton interaction mechanisms: (A) energy loss via inelastic Coulombic interactions, (B) deflection of proton trajectory by repulsive Coulomb elastic scattering with nucleus, (C) removal of primary proton and creation of secondary particles via inelastic nuclear interaction (p: proton, e: electron, n: neutron, γ : gamma rays) [5].

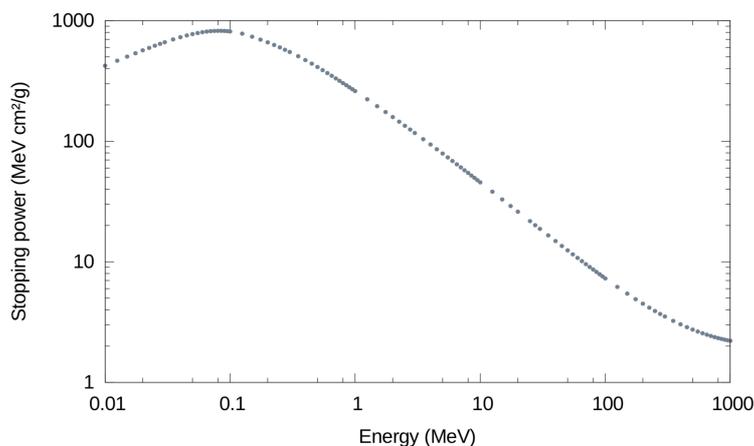


Figure 1.5: The linear stopping power of protons in water [2].

1.2.2 Multiple Coulomb Scattering

In addition to slowing down in matter, protons in the energy range considered undergo multiple small-angle deflections due to the Coulomb field of the nuclei, and deviate from their original path but almost no energy loss [2, 18]. A schematic of this interaction is represented in Subfigure (B) of Fig. 1.4. The amount of scattering is dependent on the atomic weight of the scattering material; a material with high atomic number scatters more strongly than a material with low atomic number. The Coulomb scattering distribution can be represented by the theory of Molière [28, 29].

When protons pass through a slab of material they suffer millions of collisions with atomic nuclei. The statistical outcome is a multiple scattering angle, θ_0 , whose distribution is approximately Gaussian. The root mean square (RMS) scattering angle θ_0 for “thin” objects ($x \ll X_0$) can be computed using the Highland formula [30]:

$$\theta_0 = \frac{14.1 \text{ MeV}}{p\nu} \sqrt{\frac{x}{X_0}} \left(1 + \frac{1}{9} \log_{10} \frac{x}{X_0} \right) \quad (1.2)$$

where x is the target thickness, X_0 is the radiation length of the target material and $p\nu$ is the product of proton momentum and speed at the point of interest:

$$p\nu = \frac{\tau + 2}{\tau + 1} E \quad \text{where} \quad \tau \equiv \frac{E}{mc^2} \quad (1.3)$$

and E and m represents the energy and mass of the proton, respectively, and c is the speed of light.

1.2.3 Inelastic nuclear interactions

Inelastic nuclear reactions between protons and the atomic nucleus are less frequent but have a much more profound effect. In a nuclear reaction, the projectile proton enters the nucleus where it is absorbed; the nucleus may emit secondary particles: a proton, deuteron, triton, or heavier ion or one or more neutrons [5]. These secondaries tend to have much lower energies and much larger angles than primary protons (Subfigure (C) in Fig. 1.4).

1.3 Research Objectives

In Section 1.1 the need for clinical development of a pCT system has been explained. Although pCT has not yet been clinically realized as an imaging modality, several experimental setups have been developed [19, 31, 32].

For the calculation of the RSP map, the protons' residual energies in the detector needs to be measured, in addition to the protons' path estimation. However, one of the main problems is the stochastic 'zig-zag' shape trajectories of protons within the traversed material due to multiple Coulomb scattering (MCS). This physical effect limits the spatial resolution of proton radiographic images and consequently also of the reconstructed volumetric images by "blurring" or "spreading" the incident energy away from the primary interaction site. Moreover, a certain amount of spatial resolution is necessary in order for proton imaging to be useful as a complementary imaging modality in a proton therapy facility [7]. Therefore, the development of an efficient proton track reconstruction in the detector will mitigate the effect of MCS allowing that fewer protons are needed during the scan, i.e. a lower dose to the patient and a shorter scan time and a higher signal to noise ratio. A proton track reconstruction system has already been developed by Pettersen [1]; however, it has some limitations on the track density that can be reconstructed correctly at higher beam intensities. Consequently, an improvement of this track reconstruction algorithm for protons traversing the detector needs to be carried out.

To this end, the general purpose of this thesis is to optimize the current reconstruction algorithm to improve its performance, especially at higher beam intensities. This general objective can be divided into the following specific objectives:

1. To provide a critical analysis of different particle track reconstruction algorithms as well as key studies that have been completed on this specific topic with the aim of proposing some improvement strategies.
2. To implement and evaluate different improvement proposals of the current algorithm.

1.4 Outline of the manuscript

- **Chapter 2: Technical background.** In this chapter, the description of the proton CT used in this work is presented. Additionally, different simulations tools are introduced. Ultimately, the current reconstruction algorithm is described.
- **Chapter 3: State of the art.** This section reviews the main track reconstruction techniques found in the literature and proposes potential improvements to optimize the current algorithm and get a better efficiency.
- **Chapter 4: Improved reconstruction algorithms.** This chapter describes different modifications of the current algorithm to enhance its efficiency.

- **Chapter 5: Most probable scenario algorithm.** In this chapter a new strategy is presented to get a better efficiency. Based on the study of different scenarios, this reconstruction algorithm determines which of them is the most probable.
- **Chapter 6: Discussion and conclusions.**
- **Chapter 7: Limitations and recommendations for future work.**

Chapter 2

Technical background

2.1 Proton CT concept

The proposed approach to proton CT is illustrated in Fig. 2.1. The object is traversed by a beam of protons of known energy. The protons used for imaging must have sufficient energy to penetrate the body part to be imaged. A proton-tracking detector is placed behind the patient, which records the entrance and exit points and angles of individual protons. This information is necessary to determine the most likely proton path inside the patient. Then, the residual energy of protons traversing the image object is measured with a detector. The information on this residual energy is used to compute the water-equivalent path length (WEPL) of the protons and the relative stopping power (RSP) of the tissues.

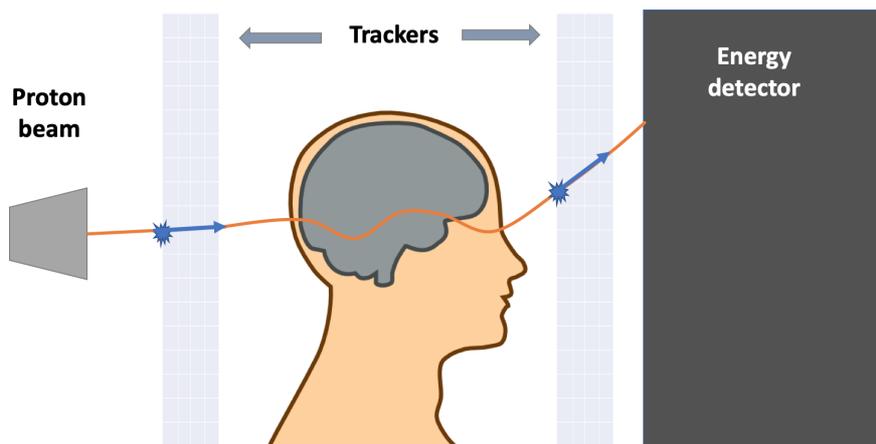


Figure 2.1: Proton CT imaging concept. The trajectory of each particle crossing an object is measured using two trackers (after and before the object). The residual energy of the particle is measured using a detector.

The RSP, here noted ϱ , of a tissue is defined as the ratio between the stopping power, S (defined in Eq. 1.1) of the tissue to that of water:

$$\varrho = \frac{S_{\text{tissue}}}{S_{\text{water}}} \quad (2.1)$$

The WEPL, here noted L , of a particle at the exit of the radiated object is the distance this particle would have had to go through in water to exit with the same energy, and is calculated by:

$$L = \int_0^l \varrho(\vec{r}) dl \quad (2.2)$$

with l the path of the proton in the tissue. From the integration of the Bethe formula, the WEPL is equal to the integration of the reciprocal stopping power of water, thus:

$$\int_{E_{\text{out}}}^{E_{\text{in}}} \frac{dE}{S_{\text{water}}(E)} = \int_0^l \varrho(\vec{r}) dl \quad (2.3)$$

By measuring the energy E_{out} upon exit from the object, and knowing the energy E_{in} at the entry, the left side of Eq. 2.3 can be calculated. Alternatively, a calibration of the system makes it possible to access directly pre-computed WEPL [2]. In consequence, it is possible to reconstruct the RSP on the right side of the same equation. In order to do so, the path l of a proton needs to be estimated.

2.1.1 Detector design in proton CT project

For proton detection, a Digital Tracking Calorimeter (DTC) is going to be used (still under development and construction), that measures the energy of each individual proton after traversing the patient at the same time. The detector consists of approximately 40 layers, each layer $\sim 27 \times 15 \text{ cm}^2$ high-granularity 1-bit pixel sensor array using the ALice P*IX*el D*ET*ector (ALPIDE) chip with an active area of $15 \times 30 \text{ mm}^2$ (1024 x 512 pixel array of $28 \times 8 \text{ }\mu\text{m}^2$ pixels) and readout rate of $\sim 5 - 10 \text{ }\mu\text{s}$ [33]. This sensor layer is followed by an energy absorber of 3.5 mm aluminum. A design based on the original prototype from [34] can be seen in Fig. 2.2.

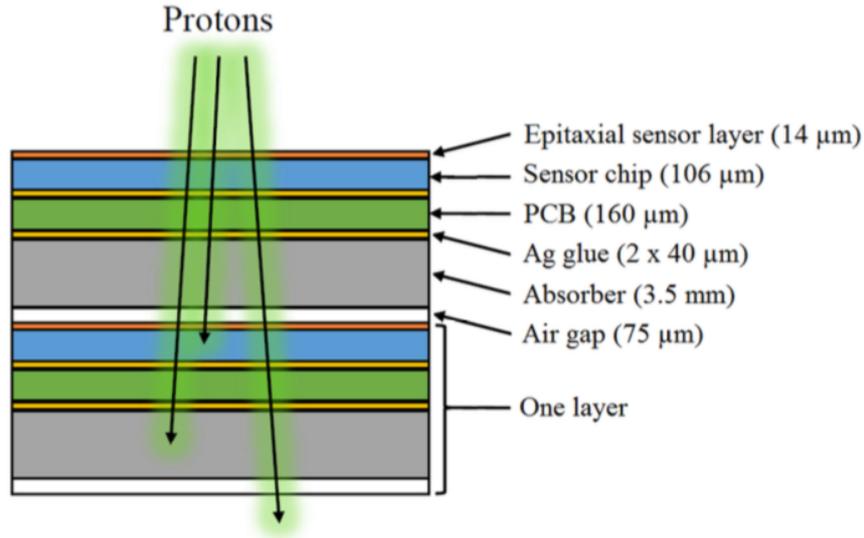


Figure 2.2: A close-up of two layers, consisting of the different components that approximate the current planned layer design. To fully slow down and stop a 230 MeV proton, 25 - 70 layers will be necessary depending on size of the Al absorber. If it's fixed to 3.5 mm, 41 layers are needed.

2.2 Simulation of a proton radiographic image

As mentioned in the previous section, the DTC is currently under construction. Therefore, to carry out the data analysis, simulations need to be performed. In this thesis, the simulations are set up using both the GATE platform, which is based on Monte Carlo interactions, and PROCASIM, a simulator created to simplify the physical interactions between the protons and the detector and have more control over the simulation process.

2.2.1 Monte Carlo Simulations

Monte Carlo (MC) methods are a class of computational algorithms that search for solutions to mathematical problems using statistical sampling with random numbers. This method can be used to automatically transport the particles shot into the detector by simulating the particle interactions in matter. The particle is generated at a given position, with a given direction and energy. The principle of particle transport in a MC simulation code can be schematically explained in three steps [2]:

- (i) First, the mean free path λ of the particle in the considered medium is computed. The interaction length of the particle, x , is randomly selected. The interaction length is defined as the distance that the particle will travel before the next interaction. Then, the probability of interaction of the particle is calculated following: $p(x) = \frac{1}{\lambda} \exp(-\frac{x}{\lambda})$. During this distance, the particle will travel in a straight line, following the direction and energy defined previously.

- (ii) A single particle can undergo different processes (like proton excitation in water, proton ionization in water, ...), therefore, the physical process happening during the interaction is randomly selected in the list of possibilities, according to the cross-section of each process.
- (iii) Finally, after the interaction process is selected, the new state of the particle is computed.

These steps are repeated until the particles are stopped or leave the simulation volume. Most of the physics quantities (energy, position, energy deposit...) are accessible at anytime during the simulation.

2.2.1.1 GEANT4

Geant4 (GEometry ANd Tracking 4) is an open access toolbox developed by CERN and written in C++ [35]. The response of a detector can be simulated and particles can be tracked inside volumes. For the calculations, models and experimental cross-sections are used. The “detailed” simulation of multiple Coulomb scattering (MCS), with all the collisions and interactions, is computationally very expensive, therefore, in contrast to the MC workflow, explained previously, Geant4 makes use of “condensed” simulations, where the global effect of the collisions is simulated after each particle position calculation. Most Geant4 multiple scattering models belong to this second class, though hybrid approaches using both condensed and detailed simulation processes also exist [2].

For condensed MCS models, multiple scattering is not selected among the processes in step (ii) previously described, but is applied at each step. Therefore, the function for each Monte Carlo step of charged particles in a media becomes [36]:

- (i) Propose step limit. Before any step, a loop is performed over all physics processes to establish the step limits.
- (ii) Convert this step length into “true” step length taking into account scattering along the step.
- (iii) Sample scattering angle and change particle direction at the end of the step.
- (iv) Sample displacement of end point.

2.2.1.2 GATE

GATE (Geant4 Application for Emission Tomography) is an advanced opensource software developed by the international OpenGATE collaboration and dedicated to numerical simulations in medical imaging and radiotherapy. It is based on Geant4 and plays a key role in the design of new medical imaging devices, including movement of source(s) and detector(s) as well as a simplified management of geometry for detectors and phantoms.

The various aspects of the simulation are described in a macro file [2]:

- Scanner geometry.
- Detector model.
- Source(s) and phantom(s) definition.
- Physical interactions of each particle type (Physics list).
- Response of the detector.

2.2.1.3 Scanner description

In the simulated proton projections using GATE (version 7.1), the scanner contains a Pencil Beam Scanning system (the spatial distribution of the beam is described by a Gaussian (σ_{xy})) [6], and a geometrical implementation of the DTC. A water phantom of variable thickness is used to slow down a 250 MeV proton beam, in order to present realistic energy spectra with residual proton ranges that span the complete detector in depth [6]. The schematics of the setup is shown in Fig. 2.3.

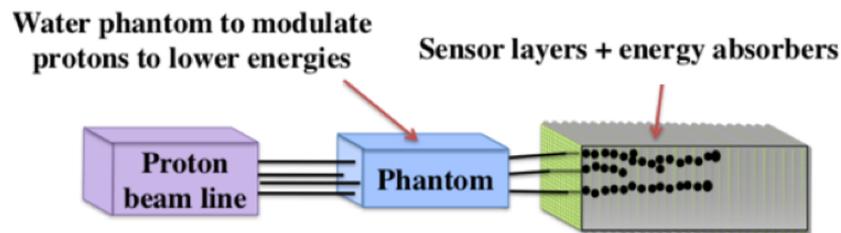


Figure 2.3: Schematics of proposed prototype. In order to obtain a spectrum of different proton beams to hit the DTC, the thickness of the energy degrading water phantom is modulated from 0 cm to the maximum range of a 250 MeV beam [6].

A Physics List needs to be selected, which contains a set of physics models. There are Physics Lists for electromagnetic physics, for hadronic elastic and for hadronic inelastic physics, etc. For MC simulations in proton therapy and in proton imaging, the recommended physics list is QGSP_BIC_EMY. It has been specifically created to address simulation problems for which high level of accuracy is requested [1].

2.2.1.4 Output data

At the end of a run involving hundreds or thousands of particles, the protons' position and deposited energy when they traverse each layer of the detector are stored in ROOT files [37]. ROOT is an analysis framework distributed freely as a C++ library. It includes functionalities such as file I/O, efficient data containers, least-squares fitting and powerful visualization tools [6]. These files contain the information on all interactions in the calorimeter. They contain, among others:

- `parentID`: The parent track's event ID. 0 if the current particle is a beam particle.
- `time`: Time in simulation.
- `edep`: Deposited energy in this event / interaction.
- `stepLength`: The length of the current step.
- `posX`, `posY` and `posZ`: Global (X,Y,Z) position of event.
- `localPosX`, `localPosY` and `localPosZ`: Local (in mother volume) (X,Y,Z) position of event.
- `baseID`: ID of mother volume scanner, == 0 if only one scanner defined.
- `sourcePosX`, `sourcePosY` and `sourcePosZ`: Global (X,Y,Z) position of source particle.
- `eventID`: History number.

2.2.2 Simulations using PROCASIM

GATE takes into account all the physics processes involved, including MCS, elastic and non elastic nuclear interactions as well as the transport of any generated secondary particle. For this reason, it is difficult to determine whether an incorrect track is due to MCS or nuclear interactions. Therefore, the goal of PROCASIM is to simulate only MCS of a proton.

Protons that experience nuclear interactions lead to large scattering angles and are the main cause for the wider envelopes obtained with the full physics list. They propagate through air until they stop at larger distances from the spot centre than those which have undergone MCS only. This effect can be visualized in Fig. 2.4 where each dot represents a proton hitting the detector.

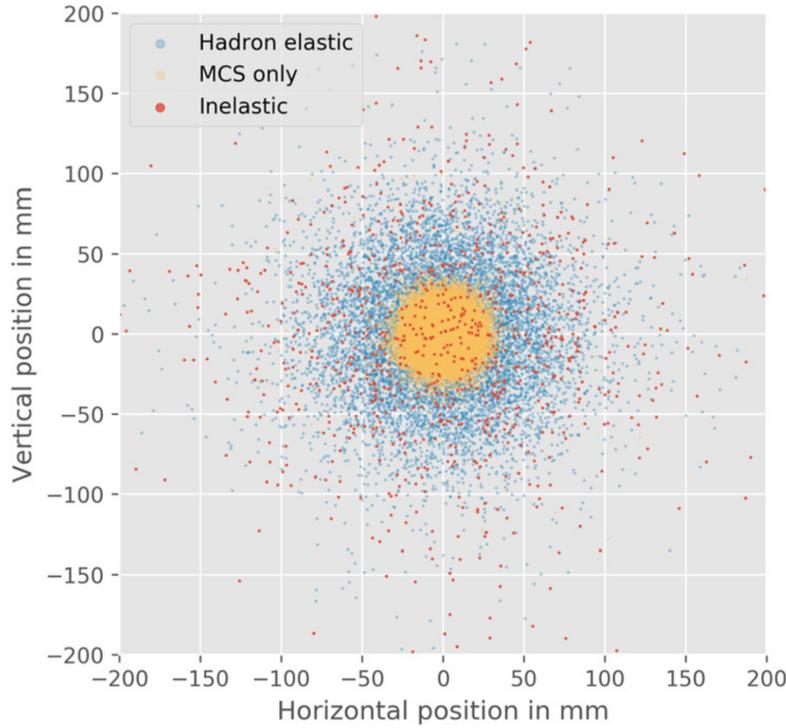


Figure 2.4: Distribution of protons on the detector surface at 20 cm distance from the water phantom. Yellow color represent protons that only experienced MCS. Blue and red dots reflect protons that have undergone elastic/inelastic nuclear reactions, respectively. The data was generated using GATE with the full physics list [7].

Protons that are subject only to MCS are selected and protons which have undergone nuclear interactions can be partly filtered out by limiting the span of exit angles and energy measurements. Consequently, the reconstruction algorithm should focus on reconstructing protons that only undergo MCS.

It is currently complex to identify if a track has been incorrectly reconstructed due to either inelastic nuclear interaction or MCS. The proposed simulator, PROCASIM (PROton CALorimeter SIMulator), facilitates the study of the algorithm's efficiency when only protons that underwent MCS are reconstructed.

In PROCASIM, each particle is generated from a single point in the first layer of the calorimeter. Then, its trajectory in the material is computed from modeling based on Highland's equation (Eq. 1.2). In this model, the position of the proton in the next layer is estimated using the expected RMS value of the MCS angle distribution θ_0 at each layer.

In the proposed model, the MCS effect is simulated using cone-surface projections. The cone is defined by the conicity angle equal to a random angle, β , selected from the Gaussian distribution with $\sigma = \theta_0$. The position of the proton at each layer is estimated by selecting a random point in the ellipse created by the intersection between the cone and the next layer. Fig. 2.5 shows a 3D representation of this concept.

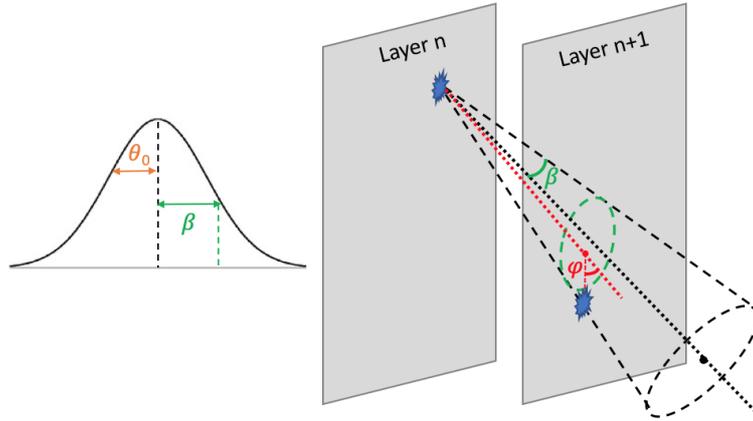


Figure 2.5: Schematic of the ellipse created by the intersection between the cone and the next layer of the detector. The size of the cone is calculated by the selection of an angle, β , in the Gaussian distribution with $\sigma = \theta_0$. Then, the value of $\varphi \in [0, 2\pi)$ is chosen randomly. The energy and the proton density that will be simulated are selected directly by the user. After simulating the indicated number of particles, the protons' position at each layer of the detector is stored in ROOT files, the framework in which this analysis is performed, explained in Section 2.2.1.4.

2.2.2.1 Parameterization of the cone

Based on the idea outlined in Fig. 2.5, the estimated position can be numerically calculated following the schematic represented in Fig. 2.6. The cone with its apex situated at the origin of the coordinate system is defined by $C(0,0;\alpha,\delta,\theta)$.

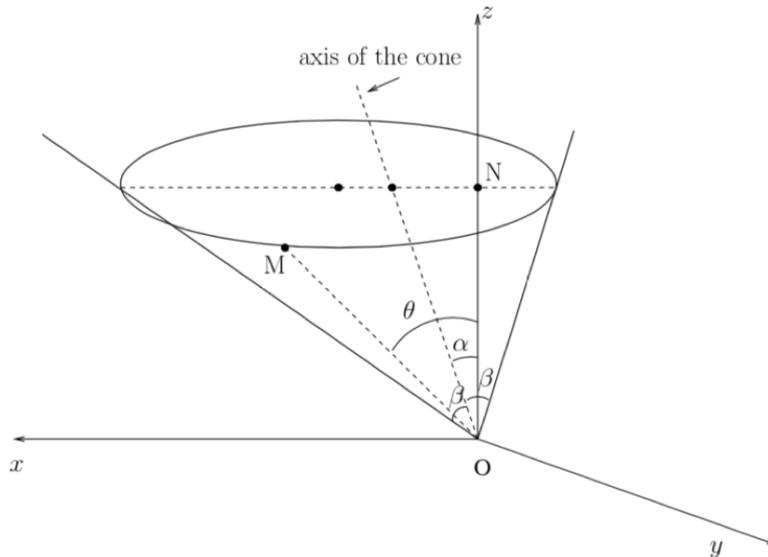


Figure 2.6: The intersection between the cone $C(0,0;\alpha,\delta,\theta)$ and a horizontal plane with $z \geq 0$ is an ellipse with its major axis parallel to the x-axis and its minor axis parallel to the y-axis [8].

In the first layer, it is assumed that the rotational axis of the cone is oriented in the vertical plane xOz and accordingly set $\delta=0$, where δ is the rotation of $Oxyz$ around the vertical axis Oz . Following the procedure in [8], a point $M(x, y, z)$ of the cone $C(0,0;\alpha,0,\beta)$ has the parametric representation:

$$\begin{cases} x = z \frac{\sin \alpha \cos \alpha}{\cos^2 \alpha - \sin^2 \beta} + \frac{\sin \beta \cos \beta}{\cos^2 \alpha - \sin^2 \beta} \cos \varphi \\ y = z \frac{\sin \beta}{\sqrt{\cos^2 \alpha - \sin^2 \beta}} \sin \varphi \\ z = z \end{cases} \quad (2.4)$$

where:

- $z \geq 0$ corresponds to the layer.
- β represents the expected scatter of the proton. This value is calculated using the Box–Muller transform [38], that generate a standard normal distributed random number, which is transformed to a Gaussian distribution with $\sigma = \theta_0$. The value of θ_0 is computed from Highland equation (Eq. 1.2) for 3.5 mm of Aluminum. θ_0 changes depending on the energy of the proton.
- $\varphi \in [0, 2\pi)$ is defined as the intersection between the cone $C(0,0;\alpha,0,\beta)$ and the horizontal plane that includes M.
- α is the angle between previous vector and Oz : $\alpha = \cos^{-1} \left(\frac{dz}{x^2 + y^2 + dz^2} \right)$ where dz correspond to the distance between layers, in this case, 3.935 mm (3.5 mm of Al + rest of components).

For the rest of the layers, the cone will be defined by $C(0,0;\alpha,\delta,\beta)$, where δ is the angle between the major axis of the ellipse and Ox axis, as can be seen in Fig. 2.7.

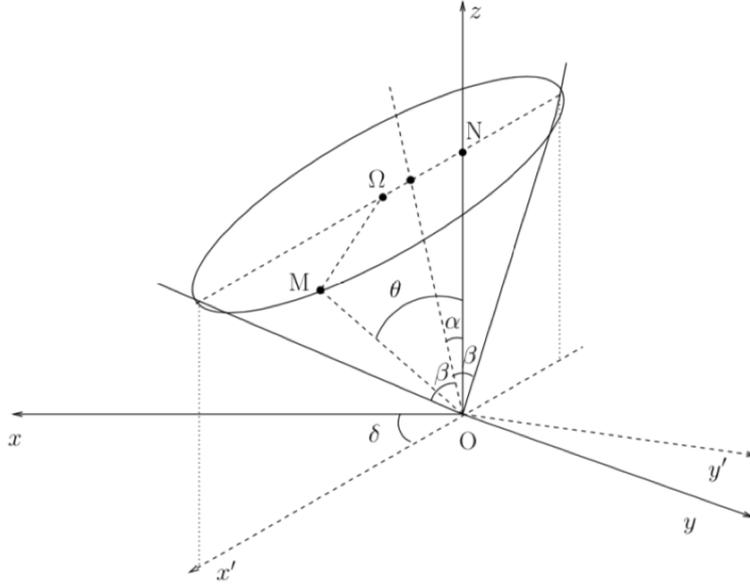


Figure 2.7: The cone's axis has an arbitrary direction described by the angular parameters α and δ . The cone's opening angle is β . The intersection between the cone and a horizontal plane with $z > 0$ is an ellipse $E(z; \alpha, \delta, \beta)$. The major axis of the ellipse forms the angle δ with the Ox axis [8].

Therefore, talking into account this new coordinate system, in the equation of the cone (Eq. 2.5), x , y should be replaced by, respectively, x' and y' . The equation of the rotation is:

$$\begin{cases} x' = x \cos \delta + y \sin \delta \\ y' = -x \sin \delta + y \cos \delta \\ z = z \end{cases} \quad (2.5)$$

For each layer, a new cone $C(0,0;\alpha,\delta,\beta)$ is calculated to determine the x and y position of the particle in each interaction until the last layer is reached.

2.3 Current tracking algorithm

One of the advantages of using a pixel-based detector design is the ability to reconstruct many concurrent proton tracks. Therefore, an in-detector track reconstruction algorithm needs to be able to handle the proton intensities and the high levels of MCS between subsequent layers that are usually associated with the protons of therapeutic energies, besides lost tracks due to inelastic collisions. Each incident proton is expected to come to a complete stop inside the detector [34].

A track reconstruction algorithm has been developed for the geometry explained in Section 2.1.1 by Pettersen [6]. This algorithm is based on the track-following scheme explained in Section 3.1.1.1.2, in which a growing track searches for deeper-laying activated pixels, while the accumulated angular change of the track, S , is compared with a global maximum, S_{\max} . The main steps of the algorithm are described below:

- (i) Starting from the first layer of the detector, identify all starting hits of the tracks, called seeds (hits in layer 0 in Subfigure (A) in Fig. 2.8). The search of the rest of hits, called track candidates, goes towards the last layer of the detector.
- (ii) At each sensor layer, a search cone is applied for each seed in order to identify all possible matches in the next layer (Subfigure (B) in Fig. 2.8). The angle of the search cone is determined by S_{\max} and its value determines the amount of deflection that is allowed during track reconstruction. Too small values lead to prematurely discarded track candidates, and too large values cause confusion by including wrong candidates where there should be none [6].
- (iii) Find the angular change for each candidate in the next layer and calculate:

$$S_n = \sqrt{\sum_{\text{layer}}^n (\Delta\theta_{\text{layer}})^2}$$
 (Subfigure (B) in Fig. 2.8), where n indicates the proton density.
- (iv) Identify the hits where $S_n < S_{\max}$ and add them to the track (Subfigure (C) in Fig. 2.8). If several such hits are identified, the one with lowest S_n is chosen as the next track segment (Subfigure (D) in Fig. 2.8).
- (v) Repeat the above steps and follow all track candidates in the "tree" recursively (Subfigure (E) in Fig. 2.8) until the last layer is reached (Subfigure (F) in Fig. 2.8).
- (vi) The hits from the resulted track are removed from the search pool and the same process is repeated for the next seeds (Subfigure (G) in Fig. 2.8) until all tracks are reconstructed (Subfigure (H) in Fig. 2.8).

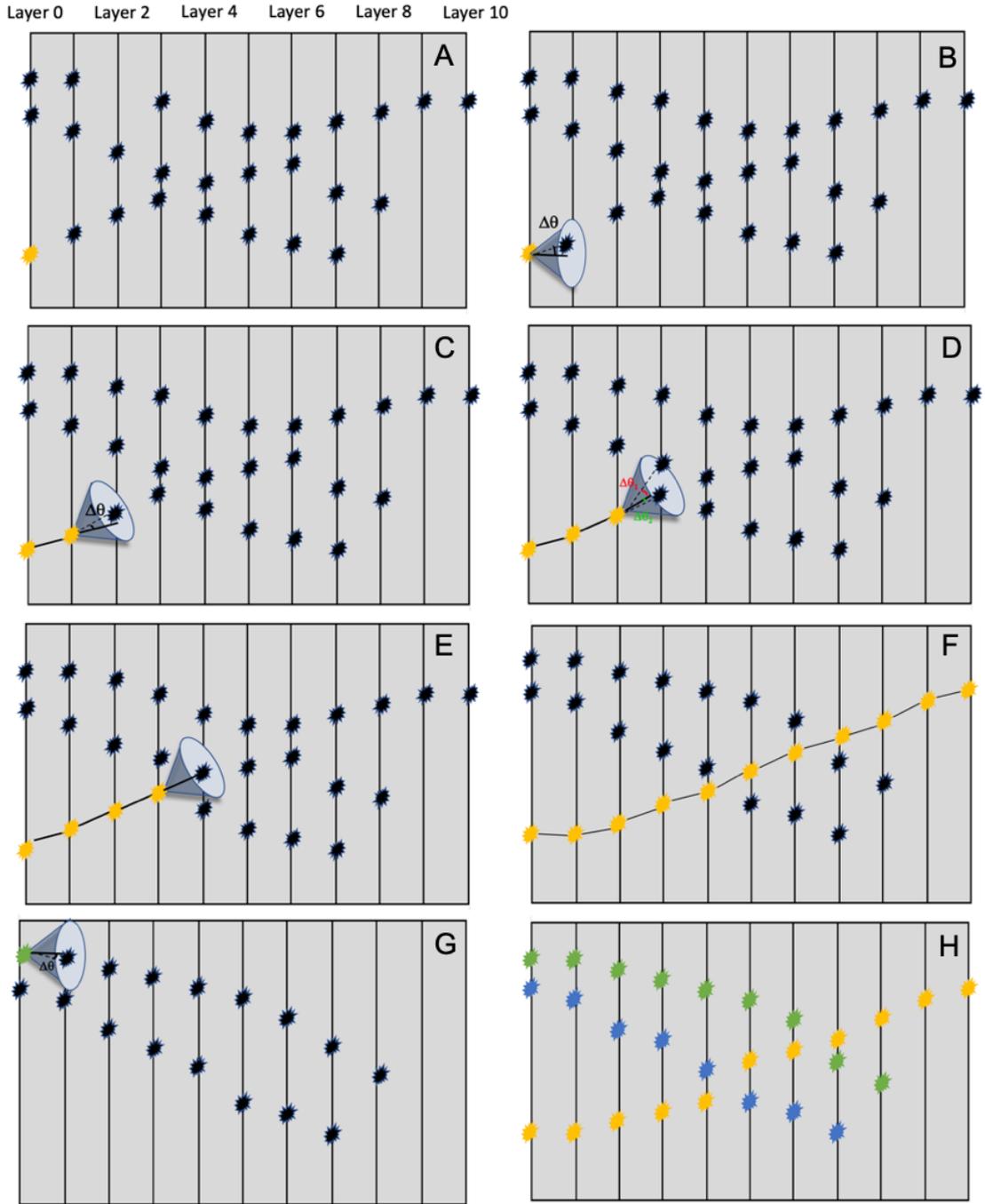


Figure 2.8: Schematic of the tracking algorithm steps. Hits in the different layers of the detector are represented by black stars. Starting from the first layer of the detector, the first hit (in yellow) of a track is selected (A). Then the next candidates are searched using a cone defined by S_{\max} (B). If calculated S_n using the new candidate is lower than S_{\max} , this hit is added to the track (C). If more than one hit is identified, the candidate with lowest S_n is added (D). These steps are repeated in the following layers (E) until the last layer of the detector is reached (F). Subsequently, hits belonging to this track are removed from the search pool and the reconstruction of the next track starts (G). The algorithm finishes when all the identified tracks are reconstructed (H).

An example of the track reconstruction applied on the data simulated with GATE is shown in Fig. 2.9. Correctly reconstructed tracks are represented as black lines, correctly but not complete track are visualized as gray lines, incorrectly reconstructed tracks as red lines and unused pixel hits as blue dots. Green lines are reconstructed secondary particles. This classification is done through checks between the primary proton identification ID tag obtained from MC simulations, which indicates the primary proton responsible for the cluster, and the final entry in the track. From the correctly reconstructed tracks, the correct residual ranges can be calculated which are the required values for the volumetric reconstruction of the stopping power map. In this example, 47 of the 62 tracks have been reconstructed correctly and are complete. A higher detector occupancy decreases the probability that all hits in a given reconstructed track originate from the same primary proton.

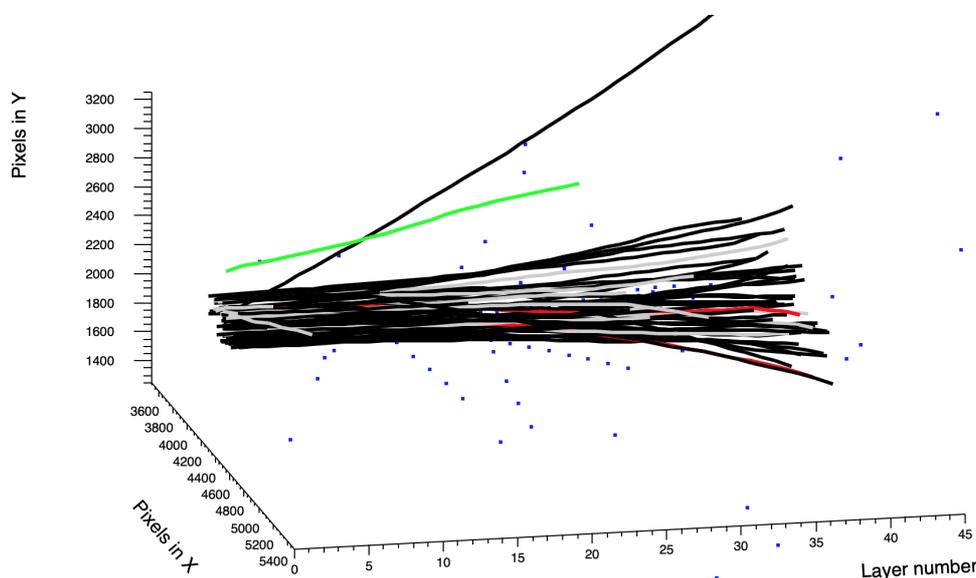


Figure 2.9: Example of track reconstruction. Black lines represent tracks that have been correctly reconstructed while gray lines are tracks that are well reconstructed but are not complete. Incorrectly reconstructed tracks are visualized in red. Green lines show reconstructed secondary particles. Blue dots correspond to unused hits.

In order to study the efficiency of the algorithm and to be able to compare the results with new proposals, the portion of tracks simulated with GATE that are identified as correctly reconstructed was calculated for different proton beam energies. A well reconstructed track means that it has the same proton history ID from GATE (eventID) at its start- and endpoints, and it has to be fully tracked. In addition, the fraction of tracks that have the same eventID in all the clusters is also calculated. Fig. 2.10 shows the efficiency graphs for both scenarios. Solid lines represent well-reconstructed and complete tracks, while dashed lines indicate tracks that have been reconstructed correctly but are incomplete. As can be observed, the higher the detector occupancy, the lower the efficiency of the algorithm.

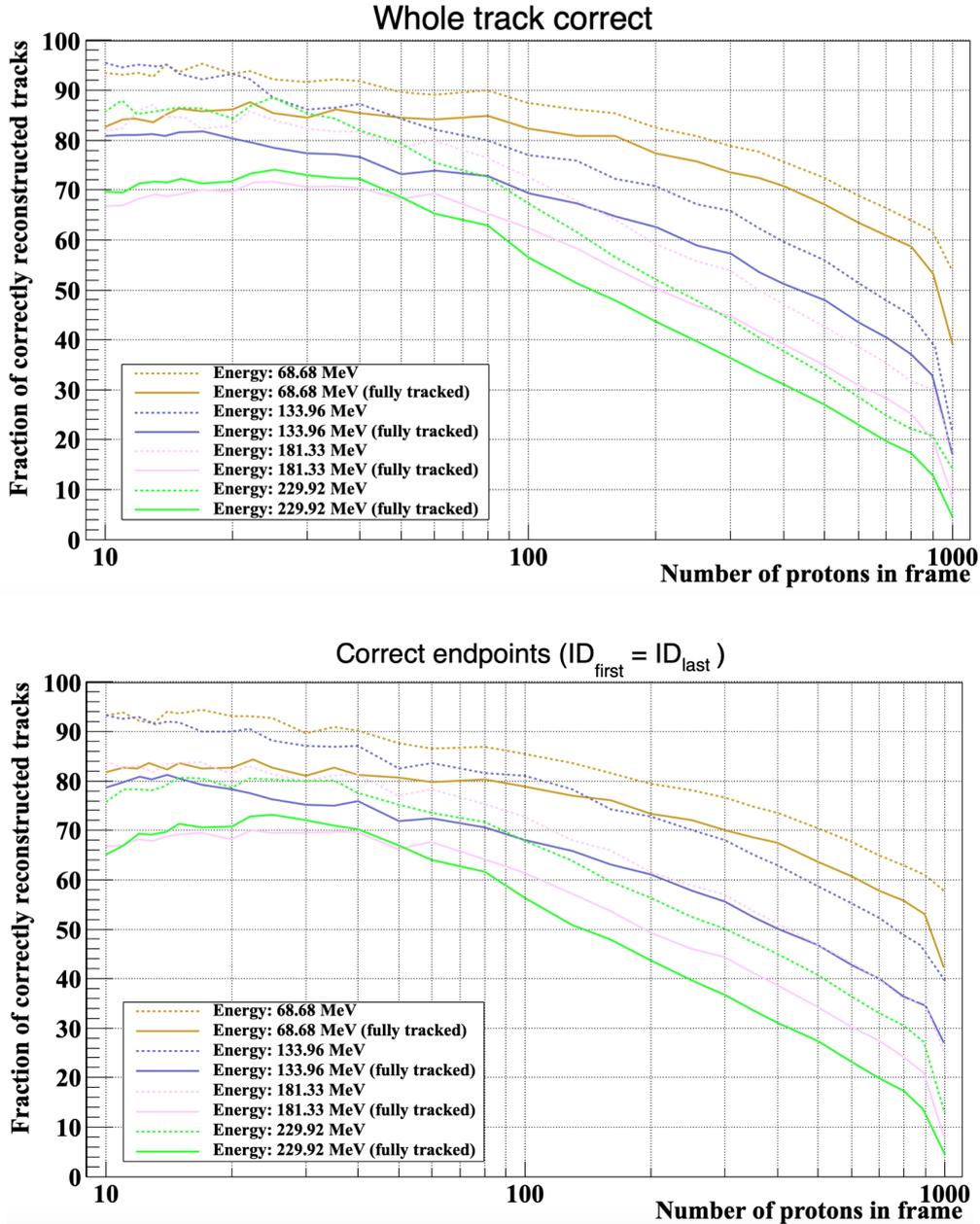


Figure 2.10: Efficiency of the current algorithm for different proton densities. In order to compare results for various energies, a 250 MeV pencil beam is degraded with a water phantom of different thicknesses (from 5 cm to 34 cm). Efficiency is measured as the fraction of tracks in which all cluster have the same eventID (whole track correct graph) and the fraction of tracks where the first and last cluster has the same eventID (correct endpoint graph). Solid lines reflect only correctly and fully reconstructed tracks, while dashed lines indicate the percentage of all well-reconstructed tracks, including the ones that are not complete. Each reconstruction is iterated 50 times.

Chapter 3

State of the art

A considerable amount of literature has been published on pCT. For example, recent studies [39, 40, 41, 42, 43] have focused on the achievement of fast readout electronic and development of a detector system that can be introduced into the clinic. Much of the recent literature also focuses on proton path approximation in the imaged object [7, 44, 45, 46, 47]. The current generation of pCT design utilizes detectors that measure the position and direction of individual protons prior to and post-traversing the patient to maximize the knowledge of the path of the proton within the patient [48]. Other studies [49, 50, 51], have concentrated on developing the pCT reconstruction algorithm to get a 3D volumetric image.

In conclusion, there has been relatively little literature published on the track reconstruction algorithm in the detector of a pCT. The main reason is that only pixel based-calorimeters need in-detector tracking, where all the hits created by a proton in the different layers need to be matched. This is in contrast to scintillator-based energy detectors [19]. The difference between these two types of detectors is the number of proton events that can be measured: while in scintillator-based energy detectors a single proton event is detected, pixel based-calorimeters can measure many events per readout. Apart from the system under study [6], only the Proton Radiotherapy Verification and Dosimetry Applications (PRaVDA) Consortium is working on the tracking algorithm on the pCT detector [52]. However, other studies have shown that to improve the performance of the pCT, a proton track reconstruction algorithm in the detector needs to be implemented [53, 54, 55, 56]. Therefore, a study of different track reconstruction algorithms for other type of particles and systems was carried out to find the best algorithm that can be implemented in the specific case of pCT, covering in this way, the aforementioned need.

However, in most of these studies, high and intermediate energy particles are analyzed, while in a pCT low energy protons are used. Since the particles interact differently depending on the energy, this factor is decisive when choosing the most appropriate reconstruction algorithm. Therefore, after an extensive and thorough investigation, none of the methods explained below will be used.

3.1 Reconstruction Algorithms

In the topic under discussion, track reconstruction is the task of finding the trajectories of the particles in the detector to measure their energy and estimate the parameters of these tracks. The tracks can most easily be reconstructed and extrapolated if they are not disturbed by material. If there were no material effects the charged particles would move parallel to the beam-line [57]. Low mass detectors are used for the detection of proton position. However, absorbent material between layers is needed to slow down and stop protons in the calorimeter. The interaction of the proton with this detector material causes the proton's path to change along the track.

Track reconstruction can be subdivided into classical and adaptive methods [58].

3.1.1 Classical Methods

In the classical approach, the task of track reconstruction is traditionally divided into two different stages: Track finding and track fitting.

3.1.1.1 Track finding

Track finding is a pattern recognition or classification problem and aims at dividing the set of measurements in a tracking detector into subsets.

Each subset contains measurements, called track candidates, which are the hits in the different layers of the detector believed to originate from the same particle. In Fig. 3.1 hits are represented with red dots. Then, the candidates of a specific track are assigned to the same subset, represented by different colors in Fig. 3.1. An additional subset contains measurements believed not to come from any of the relevant tracks, but for instance from noise in the electronics or from low energy particles spiralling inside the tracking detector. Track finding should be conservative and keep a track candidate in case of doubt rather than discarding it, as a track candidate discarded at this stage is impossible to recover at any later stage.

Track finding often uses the knowledge of how a charged particle moves inside the bulk of the detector.

Many hypotheses have to be explored in order to find the set of interesting track candidates, and track finding can in general be a cumbersome and time-consuming procedure. Computational speed is an important issue, and the choice of algorithms may be dictated by this fact [57].

Methods of track finding can in general be classified as global or local.

3.1.1.1.1 Global Methods

Global methods treat all measurements in the detector simultaneously and therefore, result independent of the starting point or order of hits. An example of a global approach that is recurrent in the literature is the Hough transform.

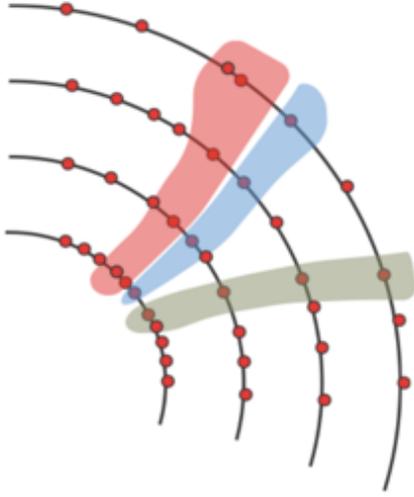


Figure 3.1: Representation of the track finding stage. Red dots correspond to the hits produced by particles in the different layers of the detector while they travel through it. Hits that are considered to originate from the same particle are assigned to an specific subset. Each subset, represented with different colors, corresponds to a different tracks.

Hough Transform

The Hough transform (HT) is a useful tool for establishing meaningful groups of feature points that satisfy some parametric constraint. The actual constraint is specified by defining a feature space of possible parameters, with each point in the space representing an instance of the ideal constraint [59]. Different studies propose HT as a pattern recognition algorithm [60, 61].

Fig. 3.2 summarizes the steps of this transform in 2D. In Subfigure (A) a track built by different hits is shown in the pattern space. This track can also be parametrized by ρ and θ following the function $\cos\theta \cdot z + \sin\theta \cdot y = \rho$. In Subfigure (B) a feature space is created using these parameters. The track is now represented as a dot in the new feature space. However, through each hit many tracks can be drawn (only a few are shown in Subfigure (C) and only for an specific hit). Each possible track drawn in Subfigure (C) can be represented as a dot in the feature space (Subfigure (D)). In the end, an infinite number of tracks can be built for each hit, resulting in a continuous curved line in the feature space. In Subfigure (F) the potential tracks for each hits are represented in the feature space by a curved line using the same color as the corresponding hit in Subfigure (E). The track that connects all the hits in Subfigure (E) corresponds to the dot in feature space (blue dot in Subfigure (F)) where curves intersect, or at least approach each other closely.

This filter can be combined with other algorithms: Delcourt [62] and Siklér [63] fit the track candidates found with HT through a Kalman filter while Aggleton et al. in [64] uses a combinatorial Kalman Filter (both filters are explained in Section 3.1.1.2.1).

Nevertheless, Aggleton et al.[64] showed that although HT is well suited to the task of recognizing tracks from a set of hits in the detector, other algorithms can run faster achieving the same score (efficiency). Moreover, the resulting histogram can be very complex when many tracks are taken into account [60].

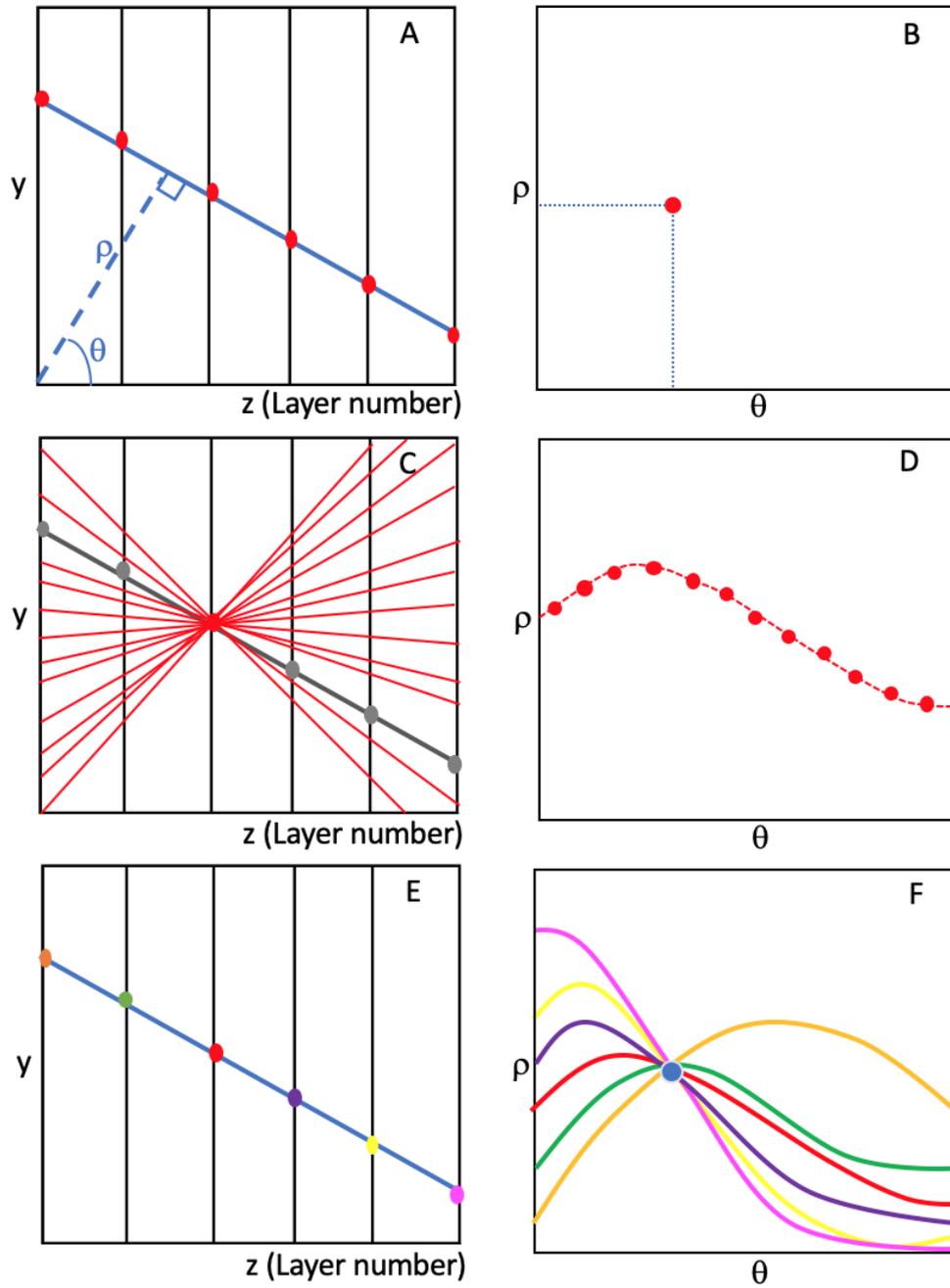


Figure 3.2: Steps in Hough Transform: A track built from different hits produced probably by the same particle is drawn in (A). This track corresponds to a dot in the parameter space (B). However, in reality, many tracks can be drawn through each hit (C). Each track in (C) represents a dot in the parameter space (D). In (F) each curve of a color represents the infinite number of tracks that go through the hit of the same color in (E). The blue point in (F) indicates the track (represented in (E)) that connect all the hits in the pattern space.

3.1.1.1.2 Local Methods

In contrast to global methods, local methods go through the list of measurements sequentially. They start from so called track seeds: short segments of tracks that allow to extrapolate the track with a track model. Via extrapolation to other measurement sites, hits are picked up if they pass a certain quality criterion, i.e. fit to the track. Three main techniques can be found in the literature: track following, track road and Kalman Filter (KF). The latter can also be used as a track fitting algorithm, therefore it will be explained in Section 3.1.1.2.1.

Track following

As outlined in [58], track following starts from a track seed. The seed provides an initial estimate of the track parameters to be used as initial state vector. From the seed, the track is extrapolated to the next detector layer containing a measurement. The measurement closest to the predicted track is included in the track candidate. A list of tracks for each event is then available and a parametric fit, such as χ^2 , from which a score is calculated to obtain an estimate of the track parameters. In case of events with multiple tracks, the track with the best score from the fit is chosen. This procedure is iterated until too many detector layers with missing measurements are encountered or until the end of the detector system is reached [65].

Track following is relatively easy to apply to tracking scenarios with moderate track density and often leads to a reasonable computational effort since the number of hits to be considered is roughly proportional to both the number of layers and the number of tracks [60]. However, the application to situations with large hit density soon reaches its limitations, since in dense environments, track following runs the risk of losing its trail whenever several possible continuations exist. The main complications can be summarized as follows:

- Some expected hits may be missing because of limited device efficiency, called a track fault. This also includes the case where the hit exists, but is out of expected coordinate bounds.
- Wrong hits may be closer to the presumed trajectory than the proper hits and be picked up instead. This can happen easily just after the seeding phase when the precision of the track parameters is still limited, or when some false hits have already been accumulated. A wrong hit may stem from another reconstructable track or from detector noise.
- Track following will always accept the hit with, for example, the smallest χ^2 contribution, which is possibly a good solution when the hit density is small. In the presence of multiple scattering and high hit densities, a wrong hit will frequently have a smaller χ^2 contribution than the proper one, or replace a proper hit which is missing due to detector inefficiency, or shadowed by another track passing the same cell. On the other hand, full evaluation of all possible hit combinations would exceed all bounds of computing resources when applied to dense events.

In track following, seeds can be constructed in the region of the tracking detector close to the interaction region, where the measurements frequently are of very high precision, or in the most distant region, where the track density is lower [58].

There have been numerous studies in which the algorithm is started from the first layer of the detector [66, 67, 68, 60]. Then a track candidate search algorithm is used to compare and rate the individual tracks by assigning a relative track score to each track. The track candidates compete against each other for the highest score and for the hits that are shared between them. A similar algorithm is presented by Lacuesta [69] where the pattern recognition works primarily inside-out but adding an outside-in tracking in order to efficiently reconstruct tracks from secondary interactions. The main advantage of this technique is that first very loose track candidates are selected and then a stringent ambiguity processor selects good track candidates.

Alternatively, an example of track following algorithm that starts from the outermost layer is presented by Amrouche et al. [70]. In this case, the algorithm starts with a random seed in the outer region and builds up a tree using the hits on the next layer until the innermost layer is processed.

Track road

In contrast to track following methods, the track road algorithm does not use extrapolation, but a much more precise interpolation between points to predict extra points on the track. It is initiated with a set of measurements that could have been created by the same charged particle. Therefore, by using initial points at both ends a simple model of the track is now used to predict the positions of further points on the track, by defining a 'road' around the track model. Measurements inside the boundaries of the road constitute the track candidate [58].

In principle, the better the model the narrower the road can be, but the theoretical road width of three standard deviations of the detector resolution can rarely be used. This is due to systematic errors in the position, to signal clusters, to signals being hidden by background signals, etc. However, the method of track roads is slower than the track following method [71].

3.1.1.2 Track fitting

Once the track finding is finished and hits supposedly belonging to the different tracks are identified, the properties of the tracks (e.g. position of the particle, direction, speed, momentum if there is a magnetic field...) need to be determined as accurately as possible [57]. This information is needed to carry out further studies.

This track parametrization starts from the track candidates identified in the track finding step. Fig. 3.3 shows an example of this stage: once hits (represented by red dots) in the detector are assigned to a track, the properties of this track are calculated. The resulted track estimations are represented by the black curved lines connecting the hits.

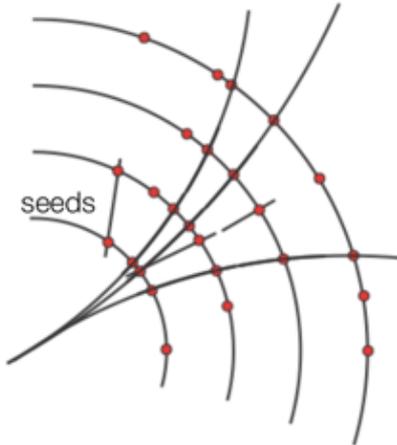


Figure 3.3: Representation of the track fitting stage. Red dots correspond to the hits produced by particles in the different layers of the detector while they travel through it. Hits are already assigned to a specific track in the track finding stage. However, the tracks need to be parametrized using the selected hits (black curved lines).

The track fit should be as computationally fast as possible, it should be robust against mistakes made during the track finding procedure, and it should be numerically stable. The track fit is also used to decide whether the track candidate hypothesis is valid. Such a test can be based on the value of the χ^2 , i.e. the sum of the squared standardized differences between the measured positions in the track candidate, and the estimated positions of the track at the points of intersection of the detector devices. If this value is too high, the set of measurements is not statistically compatible with the hypothesis of having been created by a single particle [58].

With very few exceptions (for example [72] or [73]), the estimation of the track parameters is based on least-squares methods. The linear, global least-squares method is optimal if the track model is linear, i.e. if the track propagator between layers is a linear function, and if all probability densities encountered during the estimation procedure are Gaussian. If the track propagator is non-linear, the linear least-squares method is still the optimal linear estimator [58]. However, although least-squares estimators are easy to compute, they lack robustness [74]. If there is substantial multiple scattering, the estimated track can deviate significantly from the real track. Therefore, the actual track can be followed more closely by explicitly estimating the new position of the particle at each detector layer. However, large numbers of measurements lead to a high computational cost of these methods due to the need of inverting large matrices [58]. A recursive formulation of the least squares method is the Kalman filter, which requires the inversion of only small matrices. KF can be used as track finding and/or track fitting.

3.1.1.2.1 Kalman Filter

Experience at the Large Hadron Collider (LHC) located at CERN, in Geneva, has shown that methods based on Kalman filter are robust and provide high performance. Moreover, they can incorporate estimates of multiple scattering directly into the trajectory of the particle [75].

The KF was first described and partially developed in technical papers by Swerling (1958), Kalman (1960) and Kalman and Bucy (1961) [76]. It was developed to determine the trajectory of the state vector of a dynamical system from a set of measurements taken at different times [77]. Therefore, Kalman filters are ideal for systems which are continuously changing. They have the advantage that they are light on memory (they don't need to keep any history other than the previous state), and they are very fast, making them well suited for real time problems and embedded systems.

Depending on the energy of the particles, the effects of multiple scattering can be larger, the same, or smaller than the effects of measurement error. At high energies, where the measurement error dominates over the multiple scattering, the track fitting step in KF is very important, it makes the best estimate of the proton track by drawing a least-squared-fit straight line. However, at low energies, where multiple scattering dominates, KF “connects the dots”, making a track from one hit to the next. In the intermediate region, the KF is excellent at balancing the effects of measurement errors and multiple scattering errors [78].

The KF technique used for track fitting has several advantages over a global fit:

- The KF treats multiple scattering and energy loss in and between layers. As a consequence, the fitted track better follows the true trajectory [60, 79].
- A global fit requires inversion of a matrix with the dimension of the state vector. The KF only needs the inversion of a matrix with the dimension of the measurement. This makes the KF a fast algorithm [60, 80].
- The fact that measurements are added step by step makes the KF an ideal technique for finding the measurements on the track. A global fit would require a re-fit when a new measurement is added to the track [60].
- The estimated state vector closely follows the actual path of the particle; material effects can be evaluated more precisely, and the quality of the linear approximation is better [79, 80].

As mentioned in Section 3.1.1.1.2, KF can be used as a tracking finding and track fitting algorithm, so it can be divided in two main steps:

- The **prediction step** (corresponding to the track finding stage): an estimate is made for the next measurement from the current state vector. It is very useful to discard noise signals and hits from other tracks. Once the outcome of the next measurement is observed, these estimates are updated using a weighted average, with more weight being given to estimates with higher certainty. In general, the decision power will increase when more and more hits are accumulated in the track candidate [60].
- The **filter step** (corresponding to the track fitting stage): the predicted track parameters are updated with the measurement in the current plane which updates the state vector [60].

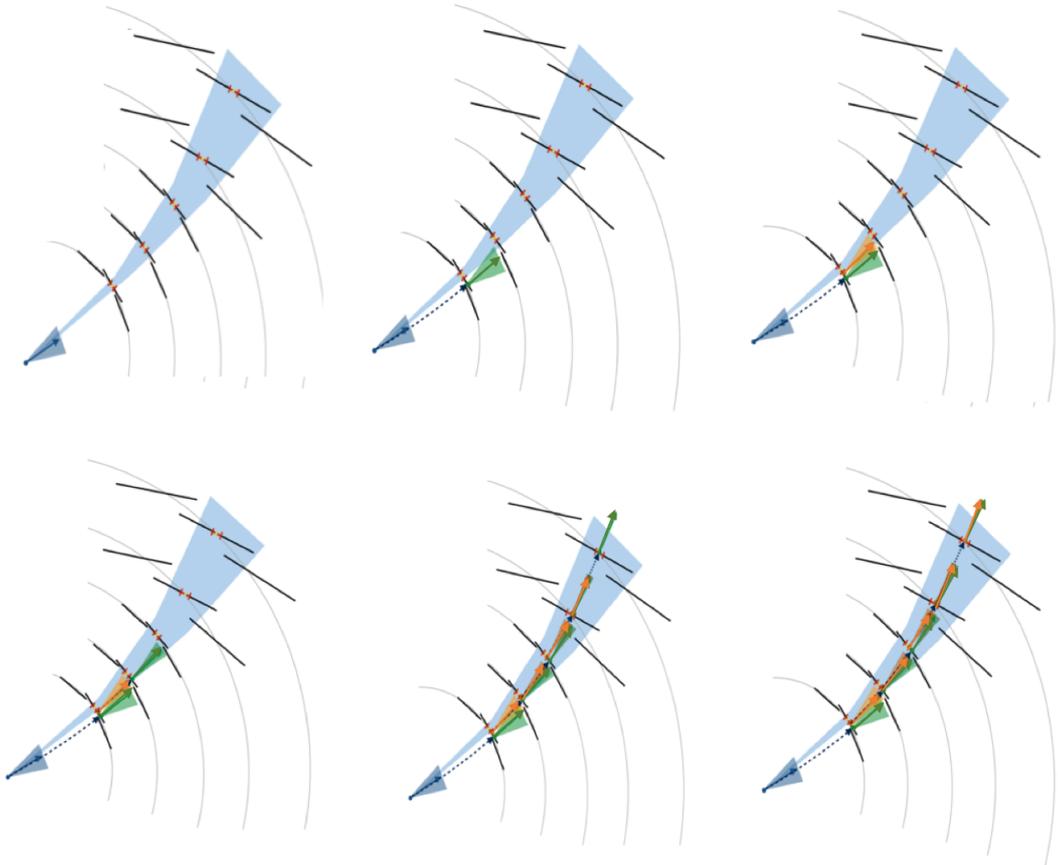


Figure 3.4: Kalman Filter algorithm: starting from the first layer, a new hit is added to a specific track during the prediction step (green arrows). Afterwards, a recalculation of the estimated track parameters is executed in the filter step (orange arrows). Both processes are repeated until the last layer is reached. The triangular shadows surrounding the arrows reflect the uncertainty at each step. As more measured are added to the track, the uncertainty in the estimations is reduced, and therefore the shadows that surrounds the arrows are increasingly narrow.

These steps are represented in Fig. 3.4. The fit proceeds in an iterative way through the full list of hits, from the inside outwards. Starting from the first layer of the detector, a hit in the next layer is assigned to a track in the prediction step (green arrows). Then, in the filter step (orange arrow), the estimated track parameters are recalculated using the new measurement. Subsequently, the prediction step is repeated to add a new measurement in the next layer, followed by the filter step. Both processes are repeated until the last layer of the detector is reached.

However, the full information about the trajectory is only available at the final hit of the trajectory (when all hits are known), which, in the presence of multiple scattering, has several consequences [80]:

- Predictions into detectors further outwards are not optimal.
- The power of discrimination between measurements which may belong to the track is rather poor at the begin of the track fit.

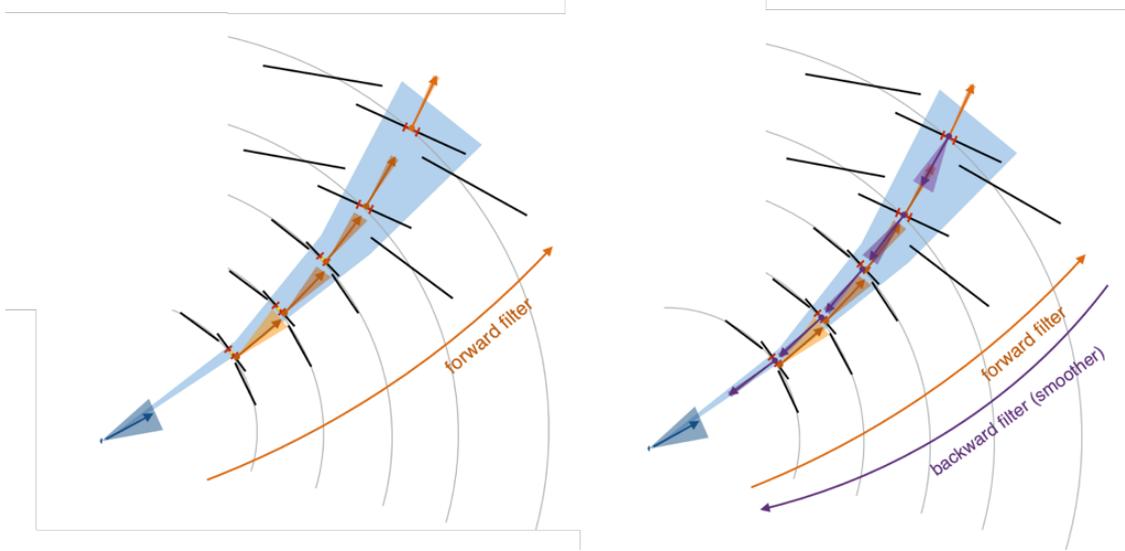


Figure 3.5: Kalman filter-smoother algorithm: it start with the Kalman filter explained in Fig. 3.4 (orange arrows). Then, a smoothing step (purple arrows) runs in the opposite direction and using the results of the KF. Using the information of both filters, a better estimation of the track parameters can be calculated. The triangular shadows surrounding the arrows reflect the uncertainty at each step. The uncertainty in the estimations is reduced as the number of measurements increases, and therefore the shadows surrounding the arrows are increasingly narrow.

Therefore, adding a smoothing part to the KF is a very useful complement to solve the problems mentioned above and extract the track parameters. The combined filter-smoother algorithm is used in different studies [81, 82, 83]. This second filter is initialized with the result of the KF, and is run backward towards the beam-line (Fig. 3.5). The track parameters at the surface associated with any of its hits, can then be obtained from the weighted average of the track parameters of these two filters, evaluated on this same surface, as one filter uses information from all the hits found before, and the other uses information from all the hits found after the surface. Therefore, this algorithm allows the computation of optimal estimates of the track parameters anywhere along the track, using the full information.

A number of studies have recognized that KF is the most useful algorithm for the track algorithm problem [60, 84].

Some authors have also suggested a variation of the KF to reconstruct the tracks: the Combinatorial Kalman Filter (CKF). It has been applied in various studies to increase the tracking efficiency [85, 86, 67, 68, 87]. The basic idea is an iterative tracking in which the initial iterations search for tracks that are easiest to find (e.g. produced near the interaction region). After each iteration, hits associated with tracks are removed, thereby reducing the combinatorial complexity, and simplifying subsequent iterations in a search for more difficult classes of tracks.

Although the success of the KF has been proven, finding a suitable initial value can be a complicated task [88, 85].

3.1.2 Adaptive Methods

During track reconstruction, track candidates may be assigned wrong measurements. Additionally, it is possible that a track passes two neighbouring cells in a layer. Such an environment with multiple, ambiguous and faulty measurements poses a challenge to the Kalman filter as it does not include a mechanism to handle competing measurements or reject measurements. Adaptive methods incorporate mechanisms that can handle competing measurements in a layer and optimize hit assignment [89]. At this point the traditional boundaries between pattern recognition (track finding) and parameter estimation (track fitting) start to dissolve.

The most used adaptive methods for particle track reconstruction in a detector are: Neural Networks (NN), Gaussian-Sum Filter (GSF), Deterministic Annealing Filter (DAF) and Elastic nets and Deformable Templates.

3.1.2.1 Neural Networks

Track reconstruction presents a challenging pattern recognition problem at the High Luminosity Large Hadron Collider (HL-LHC) [90]. Collision events contain on average 200 interactions and $O(10k)$ particles which leave $O(100k)$ space-point “hits” in the detectors. Today’s algorithms have trouble scaling to these conditions [91]. It is thus worthwhile to investigate new solutions such as methods based on deep learning. An artificial neuron manifests a simple processing unit, which evaluates a number of input signals and produces an output signal. A Neural Network consists of many neurons interacting with each other: the output signal of a neuron is fed into the input of many other neurons [92]. Fig. 3.6 represents a simplified view of NN. The connections between artificial neurons are called ‘edges’. Artificial neurons and edges typically have a weight that adjusts as learning proceeds. The weight increases or decreases the strength of the signal at a connection. Artificial neurons may have a threshold such that the signal is only sent if the aggregate signal crosses that threshold. Typically, artificial neurons are aggregated into layers. Different layers may perform different kinds of transformations on their inputs. Signals travel from the first layer (the input layer), to the last layer (the output layer), possibly after traversing the layers multiple times [93].

The idea of studying deep NN architectures is also supported by authors like Farrell et al. [94] and Mankel [60]. NN are, in principle, well suited for the tracking problem, given their ability to learn effective representations of high-dimensional data through training, and to model complex dynamics. The most commonly NN used in particle track reconstruction are explained below.

3.1.2.1.1 Recurrent Neural Networks

The first attempt to equip track reconstruction methods with adaptive behaviour was the application of the Hopfield network to track finding [95, 96]. Subfigure (A) in Fig. 3.7 shows an schematic of this type of Recurrent Neural Network (RNN). It has feedback loops from its outputs to its inputs. After applying a new input,

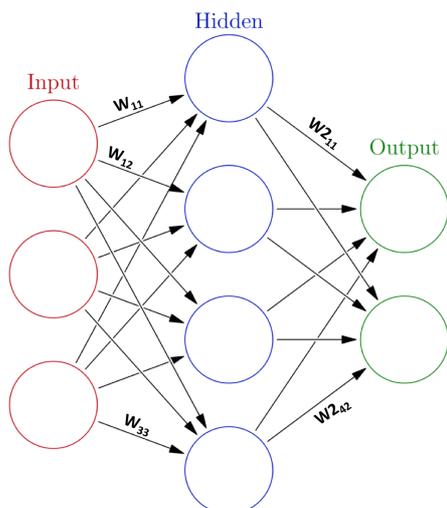


Figure 3.6: A neural network is an interconnected group of nodes. Each circular node represents an artificial neuron and an arrow represents a connection from the output of one artificial neuron to the input of another. Each connection has assigned a weight (W_{ij}) that control the strength of the signal.

the network output is calculated and fed back to adjust the input. This process is repeated until the outcome become constant. The structure of the network is not fixed but determined by the data, and each state of the network represents a hypothesis about which hit belongs to which track. The weights of the connections between the nodes are determined by some quality indicator and the sharing (or not sharing) of common hits between the two track candidates [97]. In this case, the competition takes place sequentially as each state of the network is superseded by a better one due to the dynamics of the update.

The incorporation of a physical track model into the Hopfield network has never been tried and its implementation is difficult to accomplish [58]. For this reason, the Hopfield network is not able to deliver a statistically optimal estimate of the track parameters.

Another type of RNN was proposed in recent studies [94, 90, 70]: Long Short Term Memory (LSTM) networks. This type introduces a memory cell, a special cell that can process data when it has time gaps, connecting previous information to the present task. It can process new measurements by “keeping in mind” the previous ones. They have three “gates”: input, output, forget, and they also have just the regular input. Each of these gates has its own weight meaning that connecting to this type of cell entails setting up four weights (instead of just one). Input gate decides how many information from last sample will be kept in memory; output gate regulate the amount of data passed to next layer, and forget gates control the tearing rate of memory stored. A schematic of this type of RNN is drawn in Subfigure (B) in Fig. 3.7.

LSTM networks can be used as state estimators in a way that is similar to a Kalman Filter [98]. In this proposal, the detector layers form the sequence of the model, and the predictions are classification scores of the hits for a given track candidate. Each training sample input is prepared by using only the seed hit in the first layer plus all of the event’s hits in the remaining layers. For each training sample, the LSTM model reads the input sequence and returns a prediction sequence for each corresponding detector layer. For the bins on the detector layer that have hits in them, this score

quantifies confidence that a hit belongs to the track candidate. Once trained, the model's predictions can be used to assign labels to hits. Each event will contain a number of track candidates and thus each hit will have an associated score from the model predictions on each of those candidates. The assignment is made by choosing the candidate that gave the highest prediction score to the hit.

Farrell et al. [94] consider LSTMs as an alternative to the combinatorial KF.

3.1.2.1.2 Convolutional Neural Networks

Convolutional neural networks (CNN) has also been explored in studies as a useful method for pattern-recognition in track finding [94]. A CNN consists of an input and an output layer, as well as multiple hidden layers that consist of convolutional layers. Subfigure (C) in Fig. 3.7 shows a schematic of this type of NN. CNN are algorithms that can identify aspects of visual data. Convolutional layers apply a convolution operation to the input, passing the result to the next layer [99]. Each layer in the network is tuned to respond to specific patterns, and the ensemble of features are sampled as input to the next layer to form features that recognize more complex patterns. For example, the first convolutional layers are commonly trained to detect edges of varying orientation, and the second layer features are tuned to recognize curved lines [100]. So it is expected that the initial layer of a CNN for tracking would identify stubs of compatible hits in adjacent layers. Later layers of the network would then connect stub features together to form track segments, and so on until a model of an entire track or set of tracks is constructed. In [90] it was shown that CNNs are able to construct representations of the detector data in a ground-up fashion, useful for hit assignment or parameter and uncertainty estimation.

3.1.2.2 Gaussian-Sum Filter

A large number of existing studies agree that Kalman filter is optimal when the model is linear and all random noise is Gaussian. However, the probability density functions involved are usually non-Gaussian, as the measurement errors usually have a Gaussian core with tails and the material effects (energy loss and multiple scattering) have long tails. Furthermore, the large background noise, occurring for example from neighbouring tracks or electronic noise can cause hit assignment errors [79, 101, 102, 58, 103]. One method that takes non-Gaussian distributions better into account is the Gaussian-sum filter (GSF) developed by Kitagawa [104]. In this method, all involved distributions are modelled by combinations of multi-variate Gaussian probability density functions [101].

GSF can be seen as the weighted sum of several Kalman filters. It is implemented as a number of such filters run in parallel where only the weights of the components are calculated separately. At each step, the mixture that models the state vector is convoluted with the energy loss mixture, making that the number of components of the state vector rises exponentially [105]. Thus, this number has to be limited to a predefined maximum at each step, which is achieved by clustering (collapsing) components which are close, according to a defined distance metric, pair-wise until

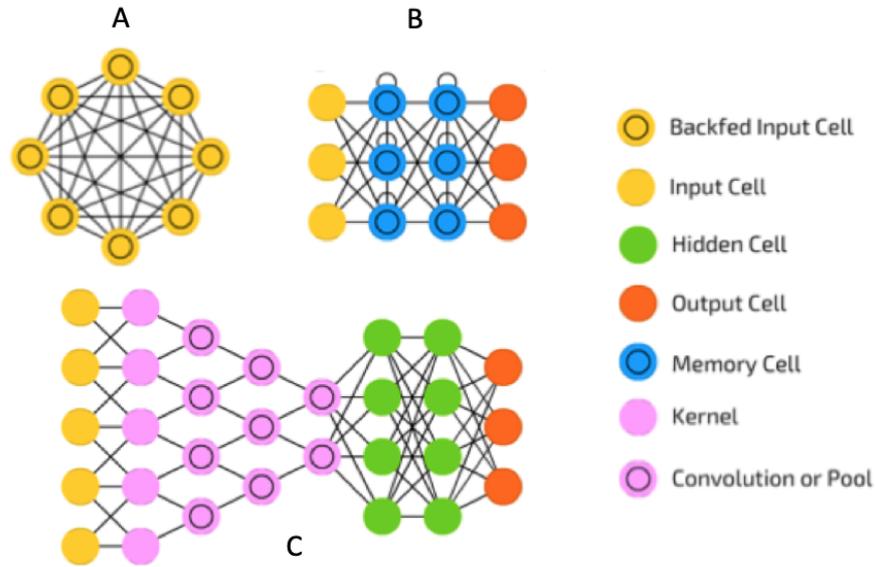


Figure 3.7: Schematic of the most used Neural Networks: (A) Hopfield Neural Network: all neurons are both input and output neurons. The network iterates to a stable state, and the output of the network consists of the new activation values of the neurons. (B) Long Short Term Memory network: apart from the input and output cells, memory cells are also incorporated into hidden layers neurons. It remembers previous values and regulate the flow of information into and out of the network taking into account previous values. (C) Convolutional Neural Network: it consists of an input, an output and multiple hidden convolutional layers. These convolutional layers apply a convolution operation to the input passing the result to the next layer.

the desired number of components is reached [102]. The output of the filter is the full Gaussian mixture of the state vector. To make this method more robust, Fruhwirth [106] introduced the concept of a missing hit, which implies that an additional component is created whenever a detector unit with a measurement is reached.

The main drawback of this approach is the sensitivity to wrong or noisy measurements particularly in the early phases of the filter where the track parameters are poorly defined [58]. Therefore, good initialization of the track parameters is essential. In [58], an experiment was done in which the robustness of the estimators was studied by contaminating the tracks with noise, which deteriorated the performance of the GSF considerably, showing its inherent lack of robustness.

3.1.2.3 Deterministic Annealing Filter

As discussed in [101], in very dense environment, hits may be degraded due to contamination by nearby tracks since the wrong hit may be chosen by the Kalman filter. The problem of insufficient information in the initial phase of the filter can be overcome by adopting an iterative procedure. After a first pass of filter plus smoother the track position can be predicted in every layer, using information from all the other layers. Based on these predictions, the assignment probabilities of all competing hits can be computed in every layer. The iteration stops when the weights have converged [79]. This process is called Deterministic Annealing Filter (DAF) [107]. It

is a single-track fit which allows for competition between hits, such that several hits may compete for a track on the same surface.

The main limitation of this filter is the need of an initial track seeds. At present, the initial pattern recognition is done by the combinatorial Kalman filter, and the tracks to refit are selected in its output. For the DAF, the hits are simply collected in a cone around the selected track. However, with this seeding strategy, the track finding efficiencies cannot be improved with respect to the combinatorial KF [79]. Therefore, better seeding methods would be needed, since this step is especially delicate.

However, as they are slower than the stand alone KF, this method can only be used where appropriate.

3.1.2.4 Elastic Nets and Deformable Templates

Some authors like Strandlie [58], Mankel [60] and Fruhwirth [79] proposed, as an alternative solution, the application of the Elastic Arms algorithm (EAA) developed by Ohlsson, Peterson and Yuille [108]. The algorithm aims to concurrently resolve the problem of finding correct hit-to-track assignments together with fitting the selected points to the respective tracks. This is done by defining a suitable energy function, which is basically a sum of squared distances between the hits in the detector and the arms, which describe the parameters of the tracks to be reconstructed. Each term in the sum is switched on or off, depending on the state of a binary hit-to-arm assignment variable. The global minimum of this energy function with respect to the track parameters and the assignment variables gives the solution to the track reconstruction problem [79].

The first test in an LHC scenario using EAA was done by Lindstrom in 1995 [109]. More recent studies showed the use of EAA in the ATLAS detector in CERN [110] to reconstruct the particle tracks. EAA was initiated by the output of a Hough transform track finding procedure [58, 60].

A related method called elastic tracking algorithm was proposed by Gyulassy [111]. The basic idea is to interpret the classical Radon transform as an interaction energy between a template track and the hits in the detector. The parameters of the template track giving the minimum interaction energy define the solution of the problem [79].

3.1.3 Other Methods

The present study focus on the more recurrent methods, however apart from the described techniques, there are many other particle track reconstruction algorithms found in the literature (for the sake of brevity, these examples are not explained here):

- Broken lines [112, 113, 114].
- Tracklet algorithm [115, 62].
- Fuzzy random transform [60].
- Linear approximation algorithm [70].

- Nearest neighbours algorithm [70].
- Matching algorithm [116].
- Cellular automaton [97].
- Greedy algorithm [97].
- Multi-track filter [101].
- Adaptive filter [102].

3.2 Discussion

The purpose of this literature study was to provide a critical analysis of main track reconstruction techniques to improve the current algorithm.

Although very few studies were found about tracking algorithms used in the pixel-based detector of a pCT, a considerable amount of literature has been published related to particle track reconstructions in other scenarios.

This review has shown that although adaptive methods of track reconstruction are in widespread use today [57], the most often used algorithm for track reconstruction is still the Kalman filter. This filter is especially useful in the case of a measurement error dominated tracking or in situations where the effects of measurement errors and multiple scattering errors need to be balanced. However, when the multiple scattering effect dominates, the prediction step in KF is just a local straight line propagation.

In the KF the residuals of the observations with respect to the predicted state vector, are used only in the update of the state vector, and only in a linear fashion [79]. If the filter is to solve the assignment problem by giving different weights to competing hits, the innovations have to be used in a nonlinear fashion. An example of such a nonlinear filter is the Gaussian-sum filter [117]. The GSF in turn is more adaptive than a KF, as it explores several hypotheses about the observations in parallel until in the end, one of them is selected as the winner or the most probable ones are combined to the final result.

However, the potentially large computational cost of the GSF and a certain lack of robustness have led to the development of a faster and more robust method, the Deterministic Annealing Filter [79]. DAF is “more adaptive” than the GSF, as it does not require an explicit outlier model and thus manages with less assumptions about the data. Since the DAF itself is an iterated KF with reweighted observations, the propagation part is identical to the standard case and therefore much faster than with the GSF [58]. Moreover, the iteration procedure can be motivated from a desire of overcoming the problem of insufficient information in the initial phase of the GSF.

Nevertheless, if the average number of components in the DAF is large, it is worthwhile to give some thought to the implementation of the basic KF algorithm. For example, Strandlie [58] showed that in large LHC experiments, despite the large track multiplicity, track finding and fitting can still be comfortably accomplished by the KF. In addition, Fruhwirth [79] proved that KF is much faster than all the other

methods as it does not involve any iterations.

Regarding neural networks, deep learning has not yet been explored in depth for the problems of particle tracking, so there are many open questions about the best way to incorporate such techniques [94]. The elastic arm or deformable template, which merges a continuous estimation problem with a combinatorial optimization problem, was able to overcome this limitation but at the price of a numerical minimization of a complicated energy function [57, 79].

3.3 Conclusions

Combining the information from the literature study about the different particle track reconstruction methods and the structure of the current tracking algorithm, one of the potential strategies that could be introduced to improve its efficiency is the KF. First of all, according to the literature, it is the most commonly used method nowadays to track particles. Furthermore, it has shown high efficiency, fast computational speeds and simplicity in comparison with adaptive methods like GSF or DAF.

However, in most of these studies, high and intermediate energy particles were analyzed, where the KF is a perfect candidate to track particles. However, in a pCT low energy particles are used, where MCS is the dominant effect. Therefore, the tracking problem becomes a pure “connecting the dots” problem. So, although the KF could be used to solve the tracking problem, in our case, the measurement error can be negligible, and the job can be done with a straight-line propagation to the next layer.

In conclusion, the best option would be to upgrade the existing algorithm, which follows the track following method. Currently, the tracks are built one by one no matter how much MCS that proton has underwent. Once the track has been completed, the hits belonging to that tracks are removed from the search pool so other tracks cannot use them. Therefore, an improvement would be to prioritize the track reconstruction depending on their difficulty: tracks with less scatter (less angle) will be treated first, leaving the most difficult for the end. Other option could be to begin the track reconstruction from the distal end of the detector, which allow us to know exactly the starting energy of each proton. With this information, the value of S_{\max} can be adapted depending on the expected scatter in each layer, i.e. in the first 4-6 layers the scattering should be higher due to the Bragg peak, but after that, a stricter limit could be placed on S_{\max} .

Chapter 4

Improved reconstruction algorithms

This chapter proposes improvement of the algorithm presented in Section 2.3.

4.1 Track reconstruction quality

The efficiency of the algorithms are quantitatively evaluated as a fraction of correctly reconstructed tracks for different numbers of protons per readout frame of the DTC ($\sim 5 - 10 \mu s$). A track is considered to have been reconstructed correctly if it has the same eventID from GATE or PROCASIM at its start- and endpoints, and is fully tracked, which means that the layer in which the proton stops is also included in the reconstructed track. Moreover, the percentage of tracks that have the same eventID in all the clusters is also calculated. Each study has been repeated 50 times and the average is calculated.

Two different simulation softwares have been used. The setup explained in Section 2.2.1.3 has been simulated with GATE, for a pencil beam of size $\sigma_{xy} = 3$ mm. Additionally, tracks have been simulated using PROCASIM following the steps described in Section 2.2.2. The same analysis methodology is applied to results from both simulators.

4.2 Last layer starting algorithm

There are different possible philosophies on how seeds can be constructed. The current algorithm select seeds from the first layer of the detector. One way of improving this is to start the reconstruction process by choosing seeds from the distal end of the detector. In this case, the algorithm starts with a random seed in the outer region and builds up a tree using the hits in the next layer until the innermost layer is processed, following the same procedure as explained in Section 2.3. Once all particles stopping in the last layer have been reconstructed, the same steps are repeated to reconstruct particles stopping in the penultimate layer. This procedure is repeated

until the protons stopping in the second layer are reconstructed.

An advantage of starting from the last layer is that the hit density is lowest because not all particles reach this layer. Therefore, seeds can be combined more easily with hits in the neighbouring layer [69]. Another advantage is that if the algorithm starts at the detector end, each proton's starting and stopping layer is known. This is especially convenient with protons of a short range, e.g. due to nuclear interactions. In the case of a forwards reconstruction, it's very easy to continue such tracks.

Figures 4.1 and 4.2 show the efficiency of the reconstruction algorithm when data is simulated with GATE and PROCASIM, respectively. The efficiency is calculated for different beam energies and proton densities. Solid lines represent well-reconstructed and complete tracks, while dashed lines indicate tracks that have been reconstructed correctly but are not complete.

Higher efficiency is reached when tracks are simulated with PROCASIM. Since protons only undergo small-angle deflections due to MCS, it is easier for the algorithm to find the right candidate (they are closer to the expected position). For the same reason, the difference between fully tracked tracks and incomplete tracks is much greater when particles undergo different physical interactions. In addition, it should be noticed that the results with PROCASIM generally lie close together as compared to Monte Carlo where the data is more spread.

Another trend that can be observed is that the loss of efficiency increases with more protons simulated. As the number of particles rises, the number of potential candidates that are studied by each track, and the proximity between them, also increases. This means that wrong hits may be falsely included because they are closer to the presumed trajectory than the actual hit.

Furthermore, higher energy results in lower efficiency. Also, the difference between the fully tracked and incomplete tracks becomes larger at higher energies. These tendencies will be discussed in Section 4.5.

All these trends are applied to the rest of results obtained in the different studies of this chapter.

Results shown in Fig. 4.1 reveal a significant improvement over the current algorithm as shown in Fig. 2.10, especially at higher densities.

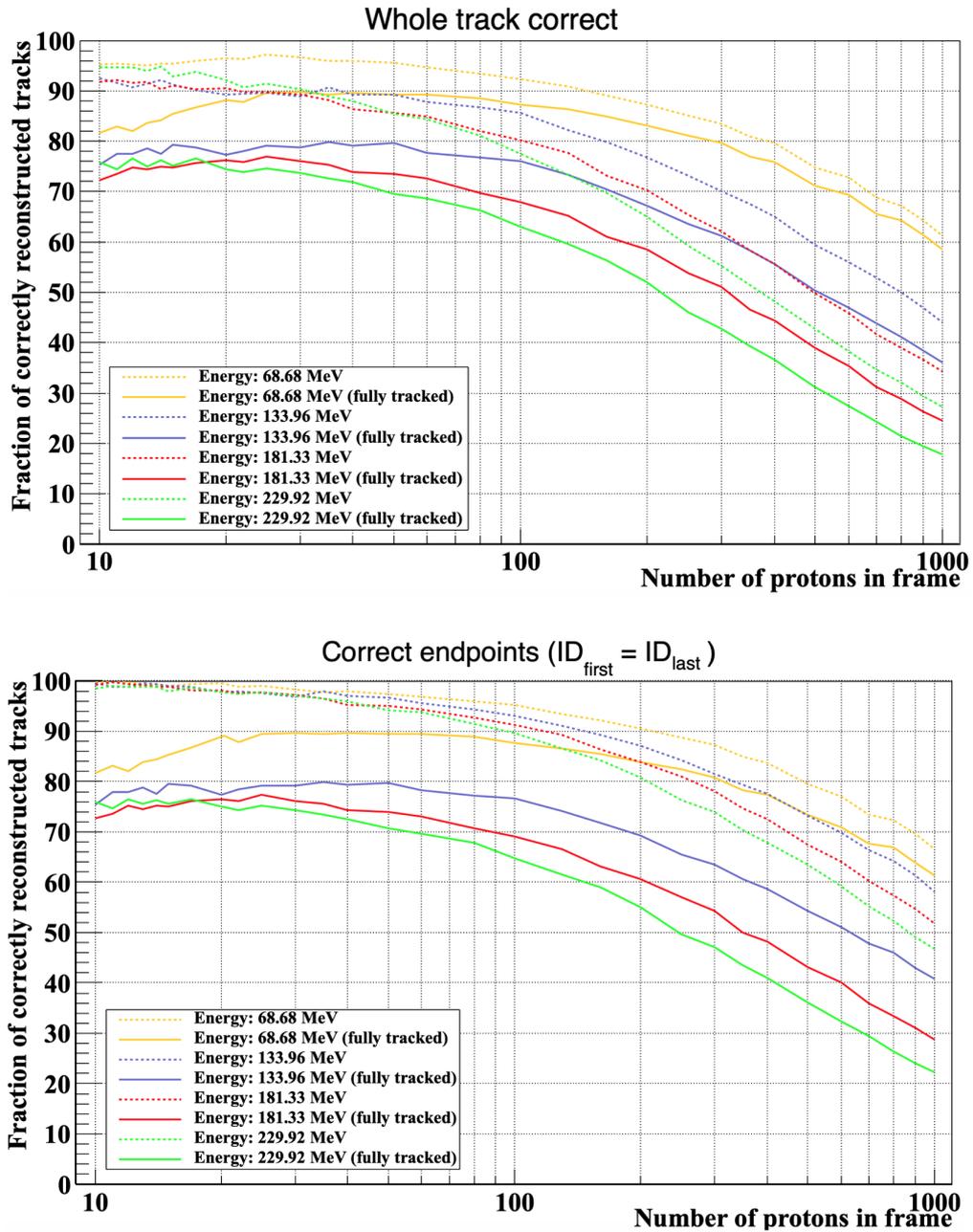


Figure 4.1: Efficiency of the last layer starting algorithm as a function of proton density for different beam energies. Data simulated with GATE. The definition of correctly reconstructed track is given in Section 4.1.

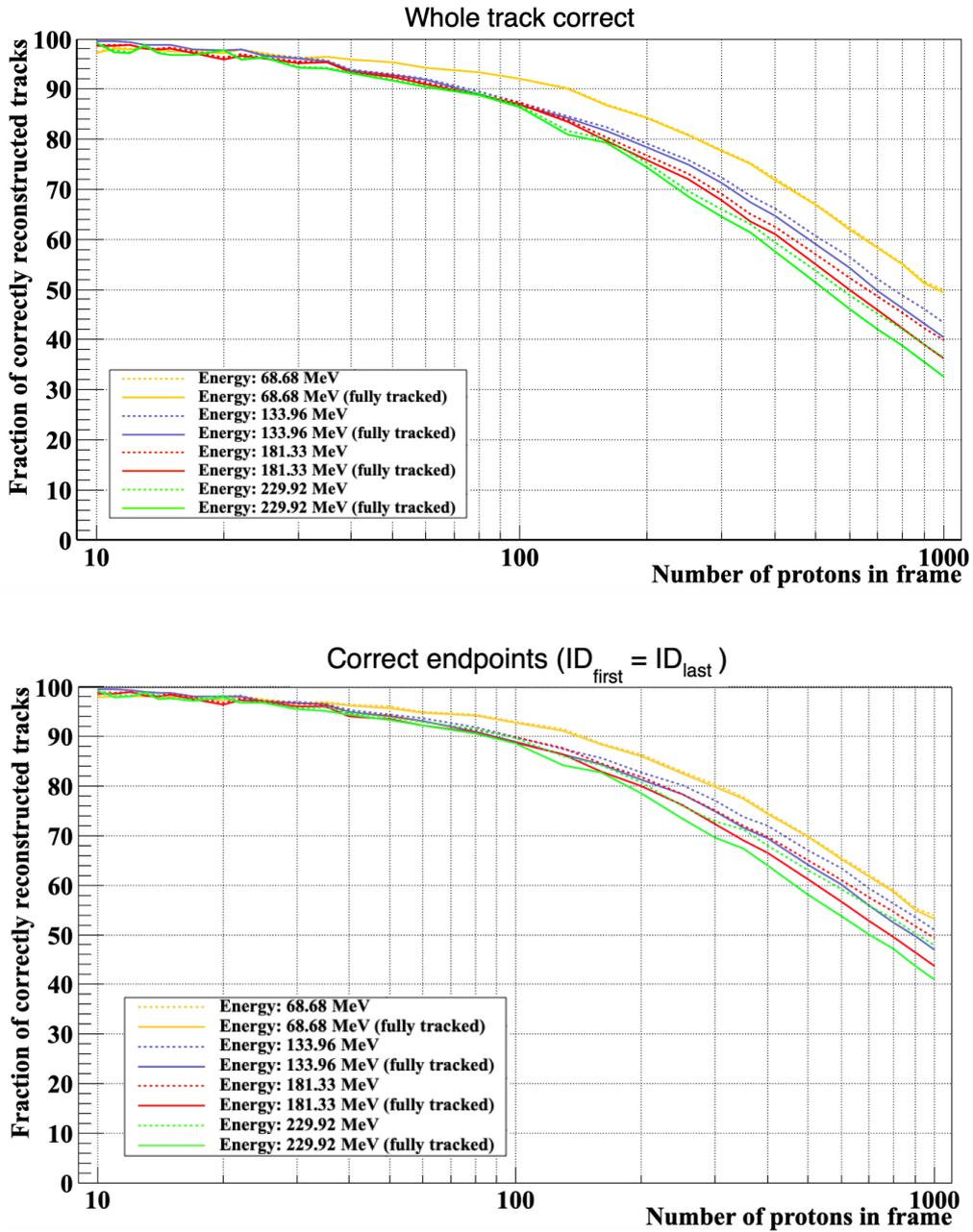


Figure 4.2: Efficiency of the last layer starting algorithm as a function of proton density for different beam energies. Data simulated with PROCASIM. The definition of correctly reconstructed track is given in Section 4.1.

4.3 S_{\max} value optimization

As explained in Section 2.3, S_{\max} determines the amount of curvature that is allowed during track reconstruction. Its value will determine the clusters that will be analyzed as potential additions to the tracks being reconstructed. If this value is too small, some candidates would be prematurely discarded. Conversely, too large values can falsely add a cluster when none should be included (e.g. after inelastic nuclear interactions at high particle densities) [6].

4.3.1 Constant S_{\max}

The value of S_{\max} must be adjusted to get the highest fraction of correctly reconstructed tracks. A study carried out by Pettersen [6] shows how the optimal S_{\max} value can be calculated. First, the 2σ value of the summed angular spread, S_n (explained in Section 2.3), is measured in each layer. Then, the value of S_{\max} should be selected to yield the maximum number of well reconstructed tracks.

Fig. 4.3 shows a two-dimensional histogram of the distribution of S_n values in each layer, found using tracks reconstructed correctly. The value 2σ in the layer where most of the protons stop is 270 mrad. In this example, the value of S_{\max} should be ~ 300 mrad (instead of, for example, 190 mrad) to get the maximum efficiency.

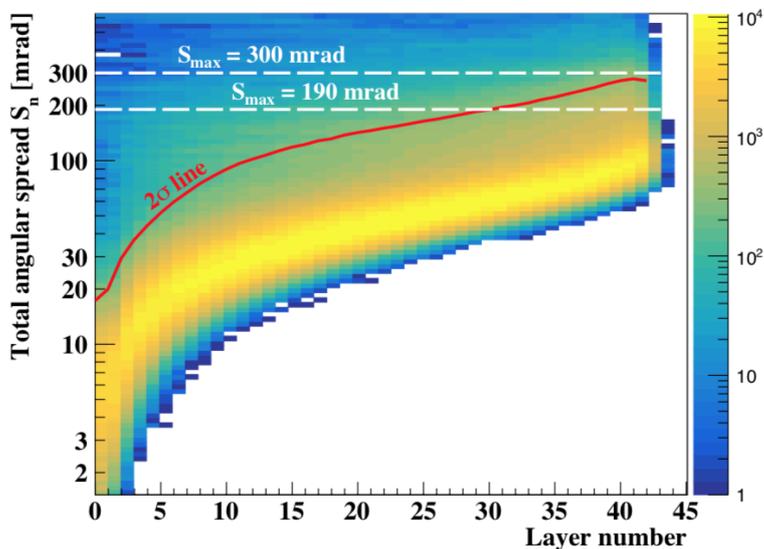


Figure 4.3: Distribution of S_n values in each layer. Red line represents the 2σ value of the distribution. Two different S_{\max} are shown. From [6].

Different values of S_{\max} should be applied depending on the particle density (n_p) and the pencil beam's size (σ_{xy}) following:

$$[S_{\max}(n_p)]^{\sigma_{xy}} = T_1 n_p^{T_2} \text{rad} \quad (4.1)$$

where the value of T_1 and T_2 are calculated from an approximation of the function $S_{\max}(n_p)$ (from a power fit of the n_p/S_{\max}). For a pencil beam size of $\sigma_{xy} = 3$ mm, $S_{\max}(n_p) = 0.469n_p^{-0.176}$ rad [6].

A potential improvement of the algorithm would be an optimization of T_1 and T_2 in such a way that the value of S_{\max} is adjusted as to yield the highest fraction of correctly reconstructed tracks. Therefore, a study of the efficiency changing the value of T_1 and T_2 in Eq. 4.1 for different energies was developed. Three values have been selected for T_1 : 0.047, 0.469 (original value), and 4.69. For T_2 , five values have been studied: 0.1, 0.176 (original value), 0.5, 1 and 1.2.

Fig. 4.4 shows the efficiency graphs when different values of T_1 and T_2 are used for a proton beam with energy 250 MeV degraded by 25 cm of water phantom, resulting in a mean energy of 133.93 MeV. In addition, the results when the beam is degraded with a water phantom of 5 cm thickness (resulting in a 229.92 MeV beam) are represented in Fig. 4.5. In Figures 4.4 and 4.5, T_1 and T_2 are defined as FirstTerm and SecondTerm, respectively, and they are rounded to two decimals places in the legend.

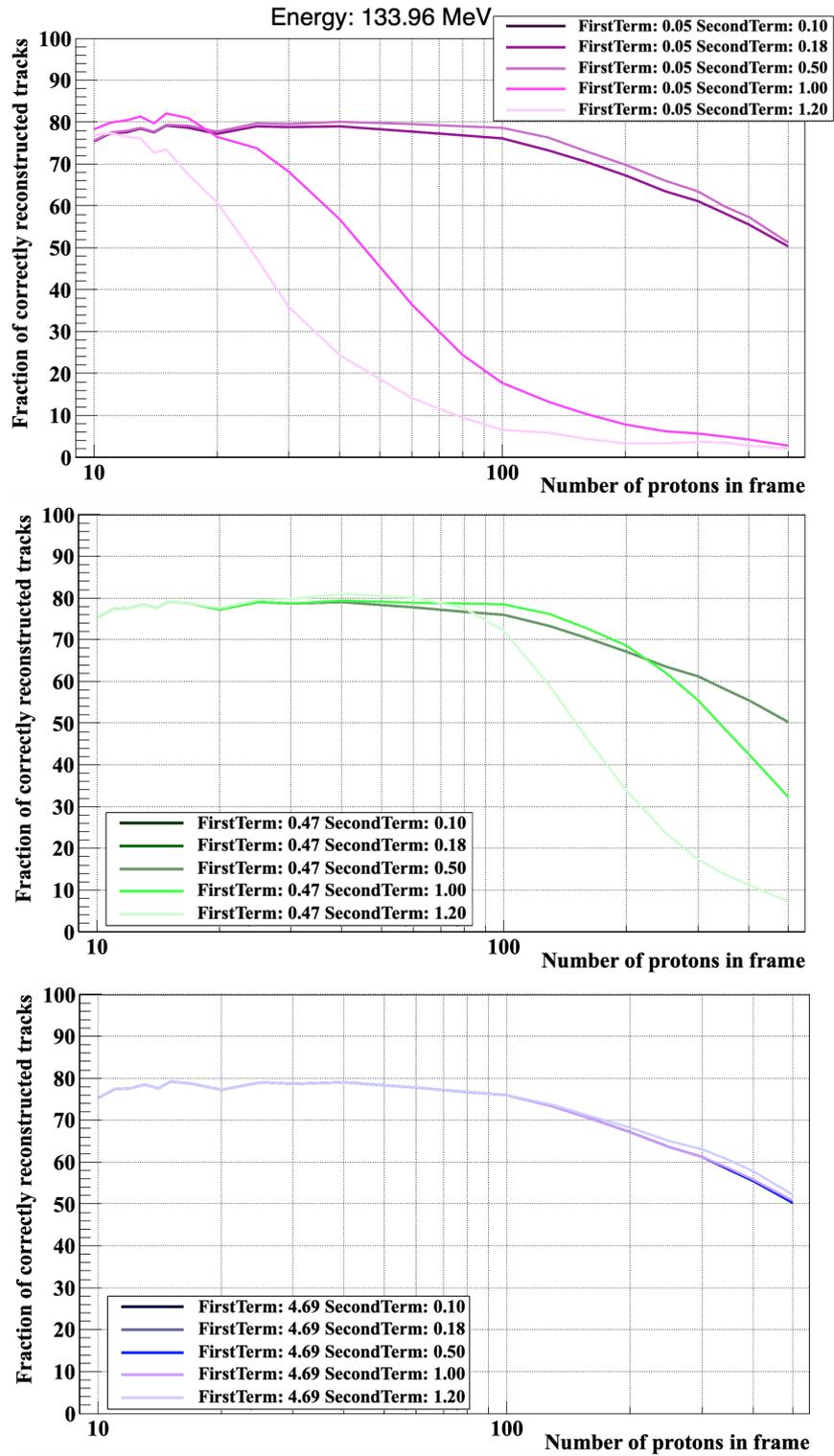


Figure 4.4: Efficiency of last layer starting algorithm depending on the substituted values in Eq. 4.1 for a energy pencil beam of 250 MeV degraded by 25 cm of water, resulting in a mean energy of 133.96 MeV. The graphs represent the fraction of tracks in which its start- and endpoints have the same eventID and are fully tracked.

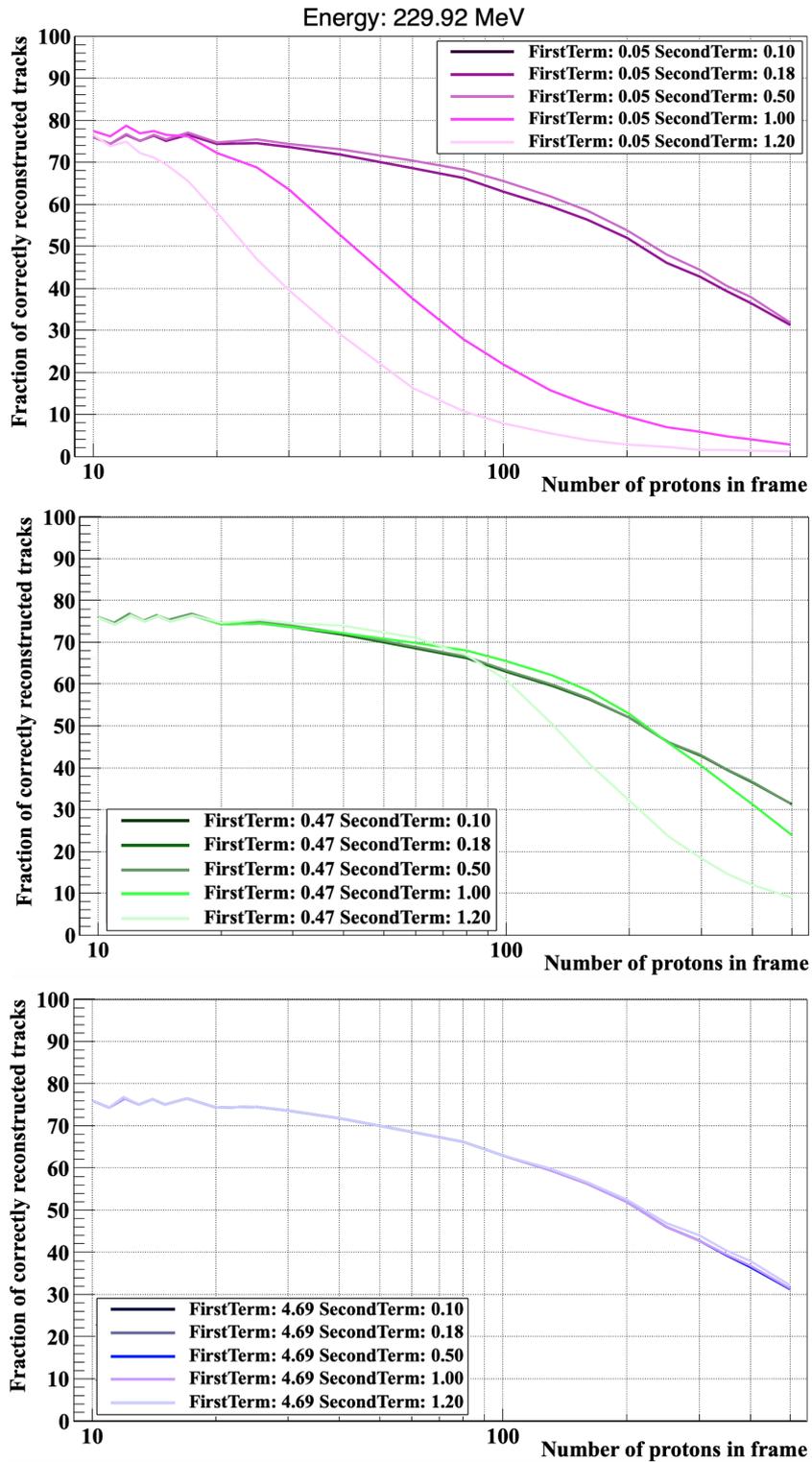


Figure 4.5: Efficiency of the last layer starting algorithm depending on the substituted values in Eq. 4.1 when a proton beam of energy 250 MeV is degraded by 5 cm of water, resulting in a mean energy of 229.92 MeV. The graphs represent the fraction of tracks in which the first and last cluster added have the same eventID and are fully tracked.

Previous studies show that different combinations of T_1 and T_2 can yield similar results. However, higher efficiency is achieved when $T_1=0.047$ and $T_2 = 0.176$, especially for densities between 40 and 300 protons. For higher densities, it seems that the improvement is less meaningful. This means that, for this intensity range, the algorithm is able to reconstruct correctly a higher number of tracks when altering the first term and maintaining the second one. The result is a stricter S_{\max} threshold during the reconstruction process, which is reflected in the number of cluster that will be studied as potential track candidates.

Fig. 4.6 shows the efficiency when $S_{\max}(n_p) = 0.047n_p^{-0.176}$ rad is applied for a beam density up to 1000 protons per readout frame. This analysis is carried out to study the effectiveness of the new parameters not only for intensities greater than 500 protons per readout but for different energies as well. As can be observed, the fraction of well-reconstructed tracks as a function of varying proton densities and phantom thickness does not differ much from the results shown in Fig. 4.1. Although for certain proton intensities slightly better results can be observed (for example when ~ 100 protons are simulated with an energy of 299.92 MeV), overall, it is difficult to appreciate significant improvements. This indicates that, although a stricter threshold is applied, it is difficult to achieve better efficiency by optimization of S_{\max} .

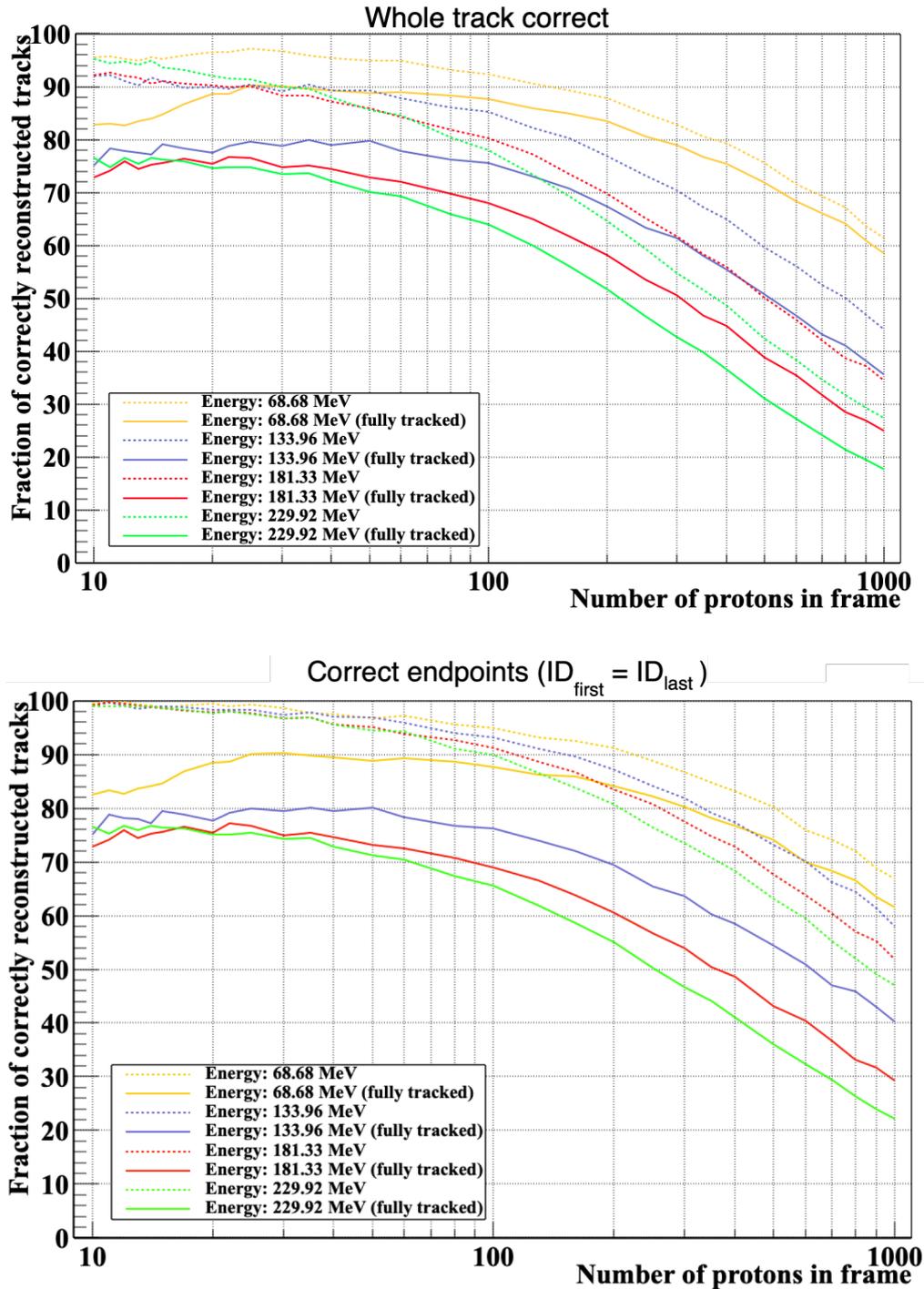


Figure 4.6: Efficiency when $S_{\text{max}}(n_p) = 0.047n_p^{-0.176}$ rad is applied for the reconstruction of a pencil beam degraded with a water phantom of various thickness, resulting in different energies. Data simulated with GATE. The definition of correctly reconstructed track is given in Section 4.1.

4.3.2 Proton energy dependent S_{\max}

Up to now the value of S_{\max} is defined at the beginning of the reconstruction algorithm as a constant value. However, the magnitude of multiple Coulomb scattering (MCS) that protons undergo depends on the energy: protons are subjected to a higher scattering power towards the Bragg peak. Therefore, the value of S_{\max} should be adapted during the reconstruction process depending on the expected scatter.

4.3.2.1 Theoretical MCS

As explained in Section 1.2.2, MCS can be estimated following Highland's formula (Eq. 1.2), a semianalytical Gaussian approximation to Molière's distribution. Fig. 4.7 shows the RMS scatter angle depending on the energy (only the energy range considered for proton imaging is represented) for 3.5 mm of Al. As can be observed, MCS increases rapidly when the proton's energy is close to 0 (Bragg peak).

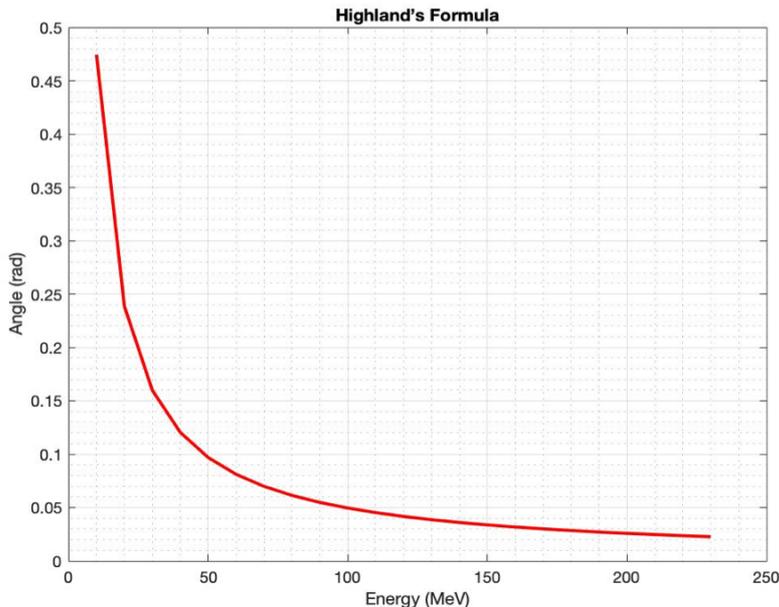


Figure 4.7: The RMS angle (in rad) of Highland's approximation (Eq. 1.2) depending on the energy of the proton (in MeV).

Before the reconstruction process, the starting layer depth and stopping position of each particle in the detector are known, allowing the calculation of the energy of the proton, and therefore the estimated MCS. The energy of the proton is calculated from a Look Up Table (LUT) that relates the distance the particle travelled in the detector and its energy. There are different LUT depending on the material and thickness of the absorber, in this case the LUT for 3.5 mm of Al is used.

From the energy, the expected scatter of the proton can be estimated using Highland's approximation and, therefore, the value of S_{\max} can be updated in each layer, making the search process of new track candidates more accurate.

4.3.2.2 Measured MCS from simulated data

In order to prove that Highland's formula (Eq. 1.2) can be applied to determine the value of S_{\max} , the distribution of the angular deviation was studied for each layer. The severity of angular deviation reflects the amount of MCS at that layer. These values are calculated using tracks that were reconstructed correctly from MC simulations.

Two-dimensional histograms of the angular difference's distribution between each layer and the 4σ value of a fitted Gaussian distribution (being 99.99% the probability that a hit belongs to the interval) for two different energies are shown in Fig. 4.8.

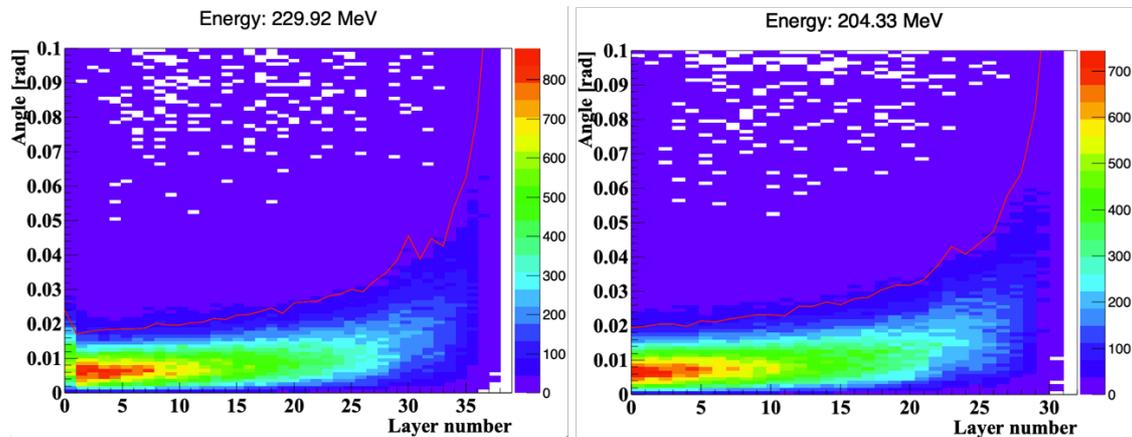


Figure 4.8: Distribution of angular change at different layers from reconstructed tracks. Data were simulated with GATE with a 250 MeV beam modulated by a water phantom of 5 cm (229.93 MeV) and 11 cm (204.33 MeV). The red line represent the empirical 4σ value of the distribution at each layer.

A curve based on polynomial functions that has the best fit to the 4σ graph is calculated to obtain a smoother version. Left panel in Fig. 4.9 shows the fitted curve (in blue) over the 4σ curve (in red). Right panel in Fig. 4.9 illustrates the fitted curve of 4σ values of a fitted Gaussian distribution calculated for different beam intensities. As can be observed, the fitted curve function is independent of the number of detected hits.

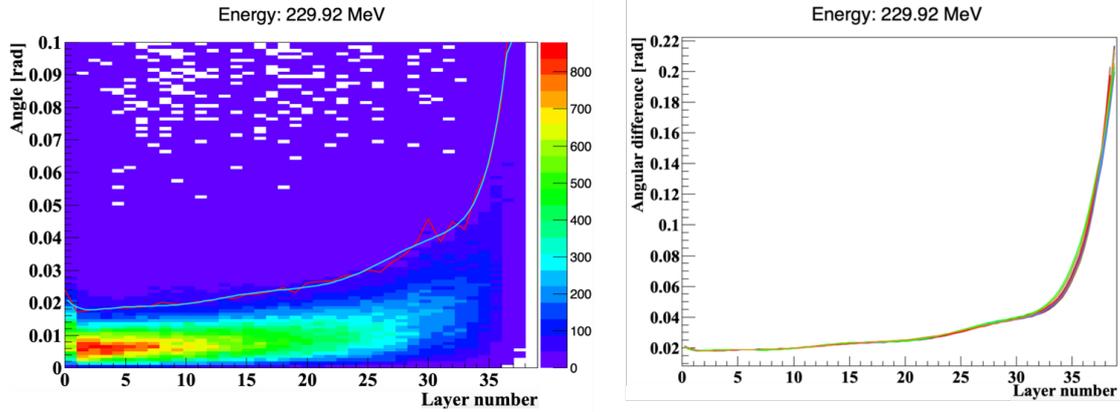


Figure 4.9: Left: Fitted curve (in blue) and 4σ curve (in red) over the 2D histogram of distribution of the angular difference between layers. Right: Fitted curves for different number of protons.

Finally, the same study is repeated for different proton beam energies. The resulting fitted curve of the 4σ values can be seen in Fig. 4.10 as a function of energy. The three graph share similar base shapes, but higher energies stretch out the curve in the horizontal direction.

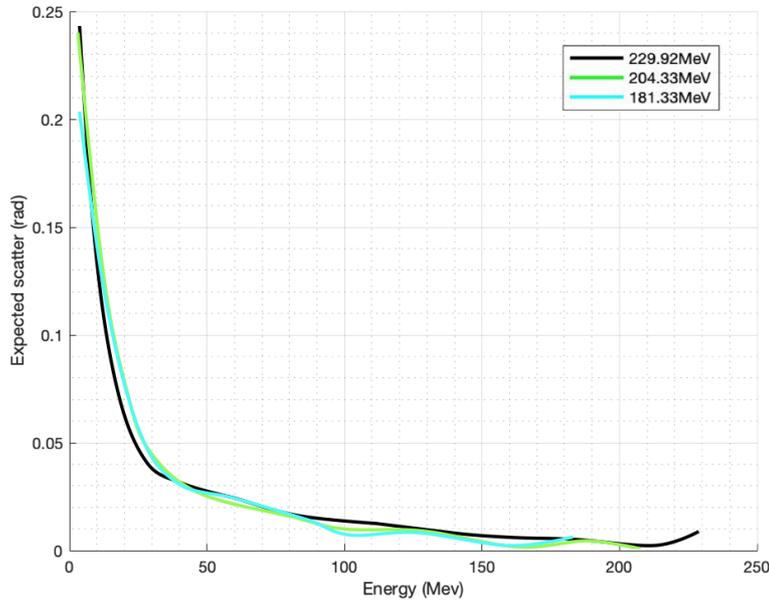


Figure 4.10: Fitted curves for different energies.

4.3.2.3 Comparison between theoretical and measured MCS values

In Fig. 4.11, the expected scatter calculated following Highland's formula (Eq. 1.2) is compared with the scatter measured from simulated tracks.

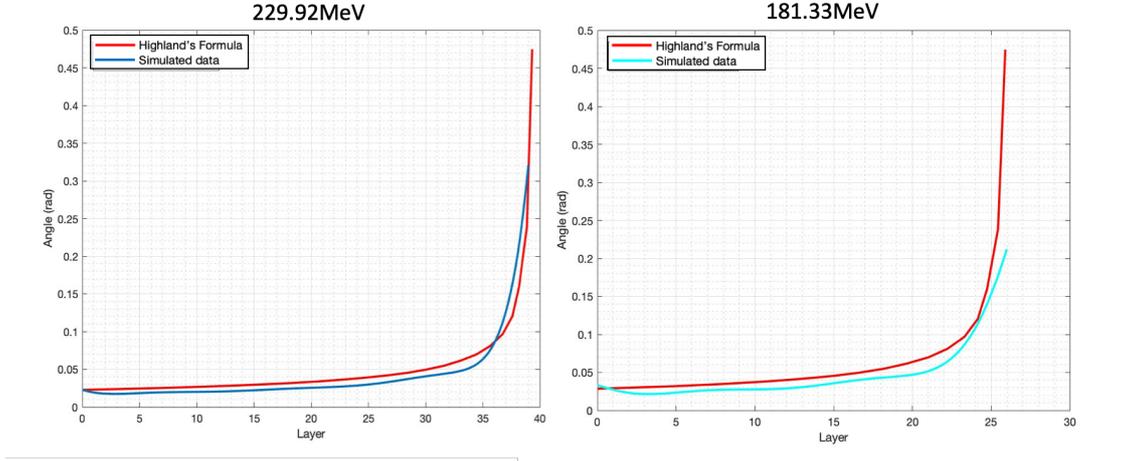


Figure 4.11: Comparison between the MCS values calculated following Highland's equation (in red) and the ones measured from simulated data (in blue) for two different energies.

As expected, the measured data are consistent with the theoretical values. Therefore, Highland's equation will be used for choosing the angle of the search cone at each layer, which will change with the energy of the proton as it crosses the detector.

4.3.2.4 Track reconstruction

Track reconstructions using different values of S_{\max} were compared. For all reconstructions, data were simulated with GATE where a 250 MeV proton beam is modulated by a 5 cm water phantom. Fig. 4.12 shows an example of the results when 100 tracks are simulated for three choices of S_{\max} :

- (i) Following Highland's formula and substituting the different variables for the case of 3.5 mm of Al as target material, $S_{\max} = \frac{2.36}{p\nu}$, where $p\nu = E \left(\frac{E+1876.54}{E+938.27} \right)$ and E represents the proton energy in MeV (Subfigure (A)).
- (ii) $S_{\max} = 4 \cdot \frac{2.36}{p\nu}$, which corresponds to 4 times the value of S_{\max} calculated previously (Subfigure (B)). This is equivalent to the 4σ value of the fitted Gaussian distribution described in Section 4.3.2.2.
- (iii) $S_{\max}(n_p) = 0.469n_p^{-0.176}$ rad (Section 4.3.1)(Subfigure (C)).

As explained in Section 2.3, the algorithm is based on the comparison of S against a global maximum S_{\max} . In case (i) and (ii) the value of S_{\max} depends on the expected scatter, therefore, S is calculated as the angular change between the potential candidate and the last cluster added to the track. On the contrary, in option (iii), the value

of S represents the accumulated angular change of the track (described in Section 2.3), and accordingly, the value of S_{\max} is adjusted to get the highest efficiency as explained in Section 4.3.1.

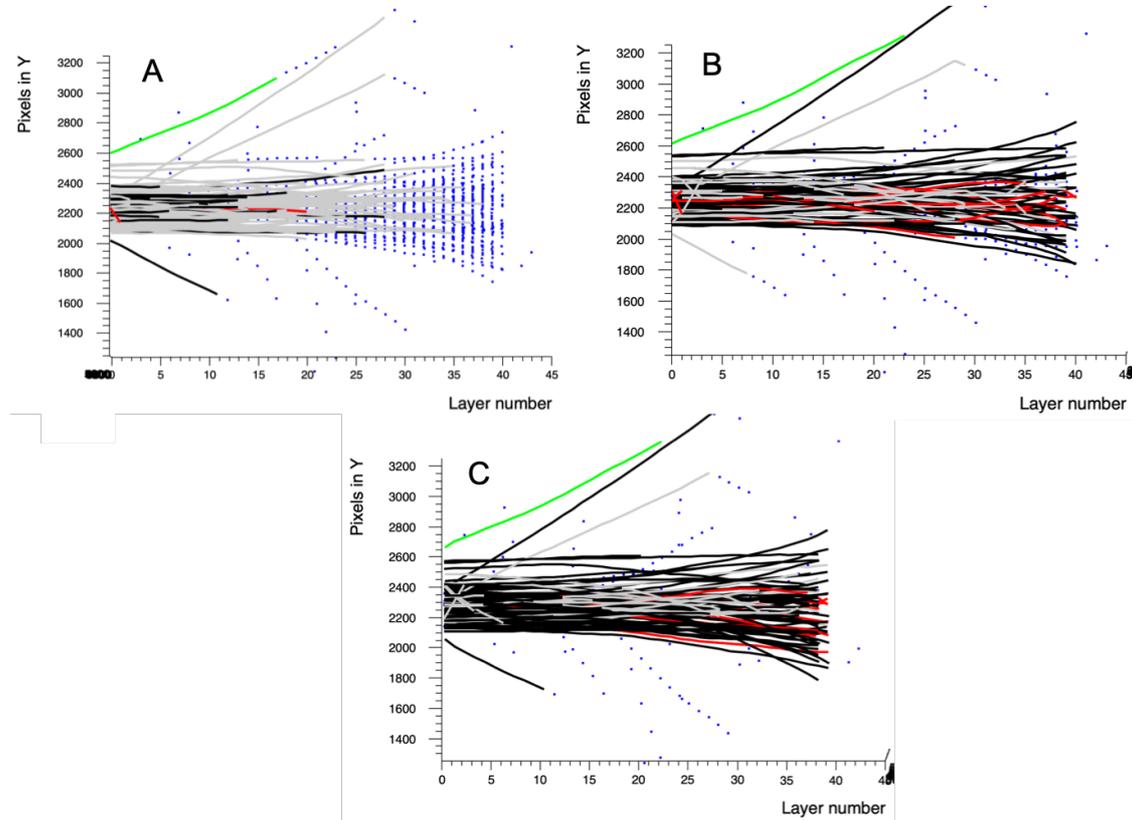


Figure 4.12: Reconstruction of 100 simulated track when a 250 MeV proton beam is degraded by water phantom of 5 cm thick and the value of S_{\max} is based on: (A) Highland’s formula, (B) four times Highland’s formula and (C) equation explained in Section 4.3.1 (constant value).

From Fig. 4.12 it can be inferred that when option (i) is chosen to determine the S_{\max} value, most of the reconstructed tracks are incomplete, as can be observed in Subfigure (A), where most tracks are gray. It reflects that this threshold is too strict, leading to correct candidates being discarded. In the other two choices (Subfigures (B) and (C)), the fraction of correctly reconstructed tracks increases significantly.

Complementary to this, Table 4.1 shows the percentage of correctly reconstructed tracks for the previous scenarios. The results are divided into tracks that have the same eventID in all clusters (all and only complete tracks) and tracks that have the same eventID in the first and last entry only (all and only complete tracks).

	Same ID in all clusters	Same ID in all clusters and complete	Same first/last ID	Same first/last ID and complete
i	~90%	~17%	~97%	~18%
ii	~75%	~55%	~85%	~57%
iii	~83%	~71%	~91%	~71%

Table 4.1: Percentage of correctly reconstructed tracks depending on the way S_{\max} is calculated.

One thing to note in Table 4.1 is that although the percentage of correctly reconstructed tracks seems higher when Highland's formula is used (option (i)), less than 20% are complete. Overall, the best results are obtained when the value of S_{\max} is calculated following option (iii) based on the equation explained in Section 4.3.1. Additionally, the difference between complete and incomplete tracks for the same definition of well-reconstructed track is lower when choice (iii) is applied, while largest difference is reached for option (i). Multiplying the value calculated according to Highland's equation by four (option (ii)) seems to be insufficient. The main reason is the large-angle scatter in the first and intermediate layers due to nuclear interactions.

Fig. 4.13 shows how the algorithm reconstructs the trajectory of a proton that has undergone at least one nuclear interaction. The left panel represents the reconstructed track when option (ii) is chosen to define the search angle. The reason why the reconstructed track is not fully tracked is due to the fact that Highland's equation is an approximation of the MCS, where the scatter angle is small. When the track reconstruction starts from layer 37, at layer 16, where the nuclear interaction takes place (shown as sudden "kink" in the graphs), the next candidate cannot be found based on its search cone, hence, the reconstruction of the track stops. When the algorithm is looking for new seeds in layer 23, it starts the previous tracks again, however (since the algorithm thinks that this is the end of the particle), the expected scatter is much higher in the following layers (from ~ 18 - 22) as was shown in Fig. 4.7, therefore, the track can handle this scatter but assigning a wrong origin. On the contrary, right panel shows the result when the value of S_{\max} is constant (Section 4.3.1). In this case, the track can be reconstructed correctly from the right seed.

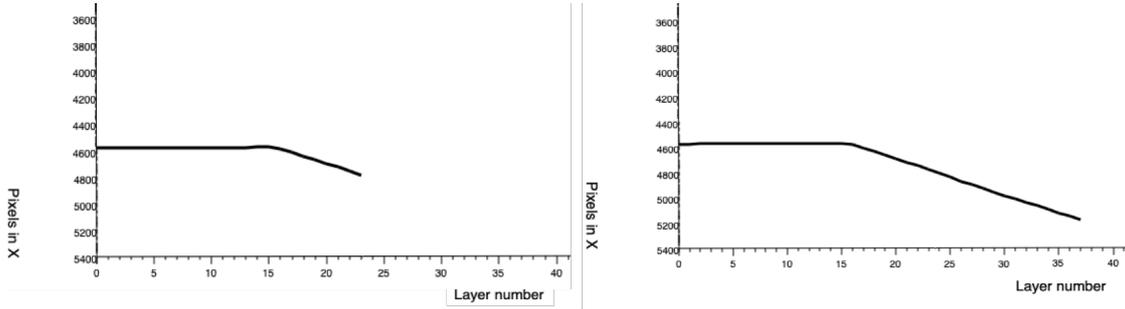


Figure 4.13: Reconstruction of MC simulated track of energy 229.92 MeV. The proton enters from left side (layer 0) and continues until it stops (layer 37). At layer 17, one scatter interaction happens and changes the direction of the proton by some angle. When option (ii) is used to define the search angle, the track is not fully tracked (left). On the contrary, when choice (iii) is applied, the track is correctly reconstructed (right).

Previous results show that the use of S_{\max} as a threshold during the reconstruction process is not enough to reconstruct as many tracks as when option (iii) is applied. Therefore, with the objective of reconstructing tracks with larger scatter angle but without increasing confusion, three different options will be used to calculate the value of the search cone (Fig. 4.14):

- (i) $S_{\max} = \theta_0$ rad.
- (ii) $S_{\max} = 2 \cdot \theta_0 + 0.03$ rad.
- (iii) $S_{\max} = 4 \cdot \theta_0 + 0.07$ rad.

Where θ_0 is the energy-dependent RMS scatter angle calculated following Highland's formula (Eq. 1.2).

The constants in the above formulae were a result of a series of simulation experiments where efficiency and duration were monitored and weighed. (i) was chosen as a baseline threshold with relatively fast execution time. An experimental optimization procedure was initiated in order to select the most viable threshold value with an optimum trade-off in duration and efficiency. Therefore, two threshold were considered: (ii) and (iii), where the addition term shifts the graph vertically while the multiplication term increases the steepness. Most tracks were successfully reconstructed using either (i) or (ii). However, certain outliers were only reconstructed correctly with (iii). The study showed that higher threshold values than (iii) did not result in an increase in well-reconstructed tracks and therefore (iii) was set as the maximum search cone angle.

At first, the algorithm attempts to reconstruct the relatively straight track using (i). If the reconstruction fails because no candidates were found, a new search cone is defined using (ii). If still the track cannot be reconstructed, (iii) is applied to try to find a potential candidate. If after this value no candidates are identified, the reconstruction of this track stops and the reconstruction of the next one starts.

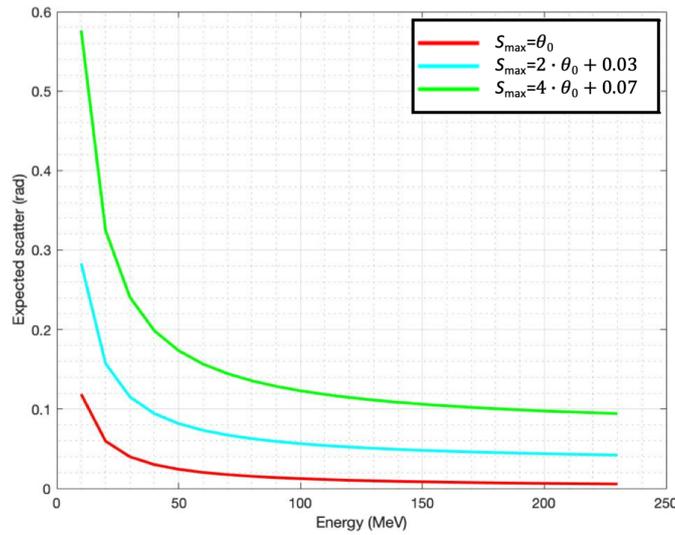


Figure 4.14: Value of search angle is calculated following Highland’s formula for different energies (red curve) and two variations of it (blue and green curves).

First, PROCASIM is used to simulate the tracks that will be reconstructed because the algorithm focuses on reconstructing protons that only underwent MCS. Fig. 4.15 shows the fraction of correctly reconstructed tracks at different number of particles.

Additionally, the efficiency of the algorithm needs to be tested with proton tracks simulated with GATE to also study its effectiveness in more real situations where the particles undergo different types of physical interactions. Fig. 4.16 shows the fraction of well-reconstructed tracks.

As expected, higher efficiency is reached when protons only undergo small-angle deflections due to MCS (i.e. when tracks are simulated with PROCASIM). Because the algorithm is not focused on reconstructing protons that undergo nuclear interactions, in some cases, it is unable to find potential candidates due to a strict threshold in the search process. As a result, the fraction of MC-simulated tracks that are correctly reconstructed is lower. For the same reason, the difference between tracks that are fully tracked and those that are not is much greater when particles undergo different physical interactions.

Although the same tendencies explained in Section 4.2 are applied, some differences can be mentioned. Even when the total number of correctly reconstructed tracks is greater compared with the results shown in Section 4.2, the fraction of tracks that are also fully reconstructed varies depending on the number of simulated tracks. For proton densities lower than 200 - 300 tracks, the algorithm described in Section 4.2 shows slightly better results. On the contrary, the algorithm presented in this section works more successfully at higher intensities, specially at higher energies. However, the difference is not very significant.

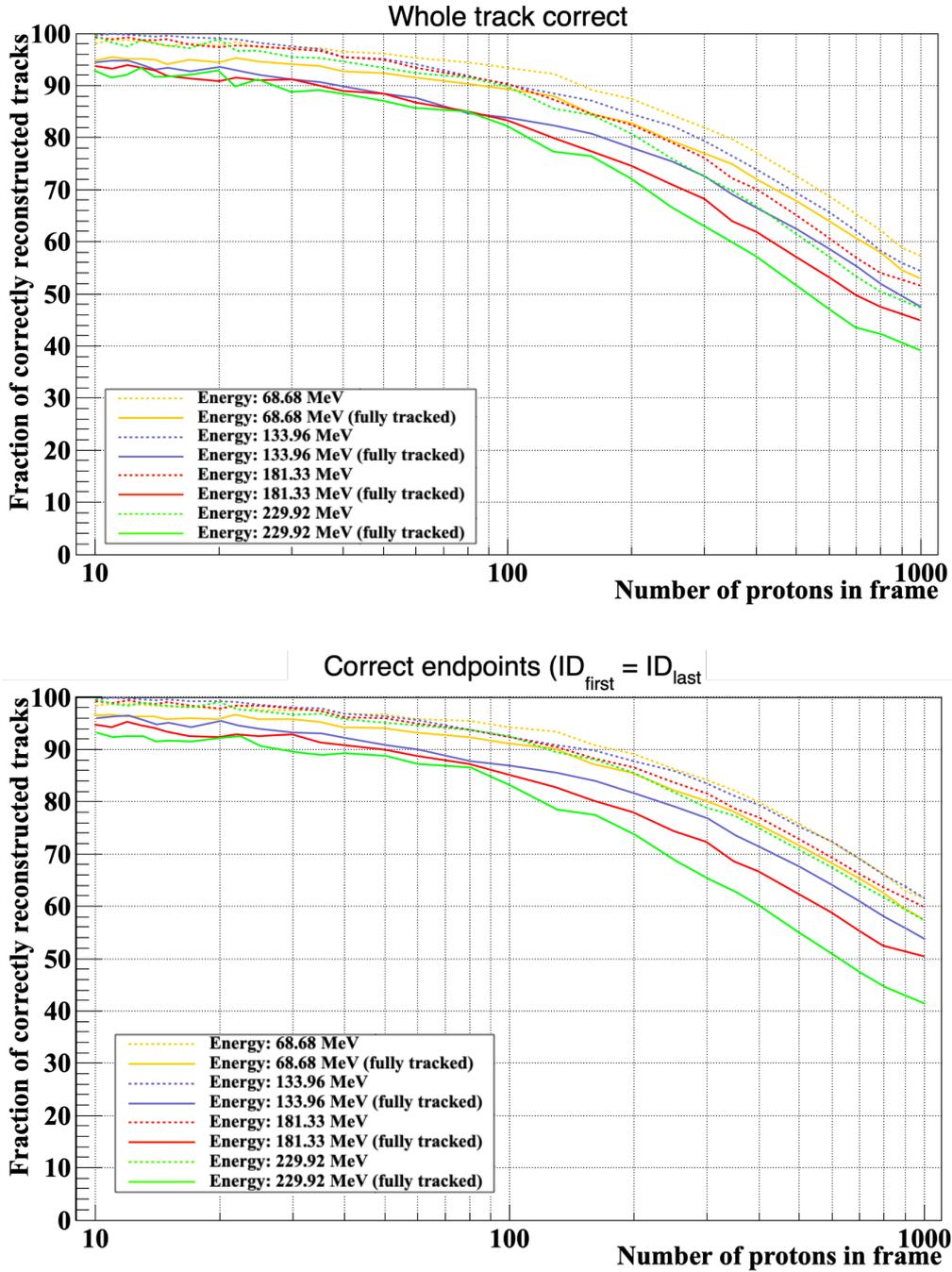


Figure 4.15: Efficiency when variable value of S_{\max} is applied for the reconstruction of a pencil beam degraded with a water phantom of various thickness, resulting in different energies. Data simulated with PROCASIM. The definition of correctly reconstructed track is given in Section 4.1.

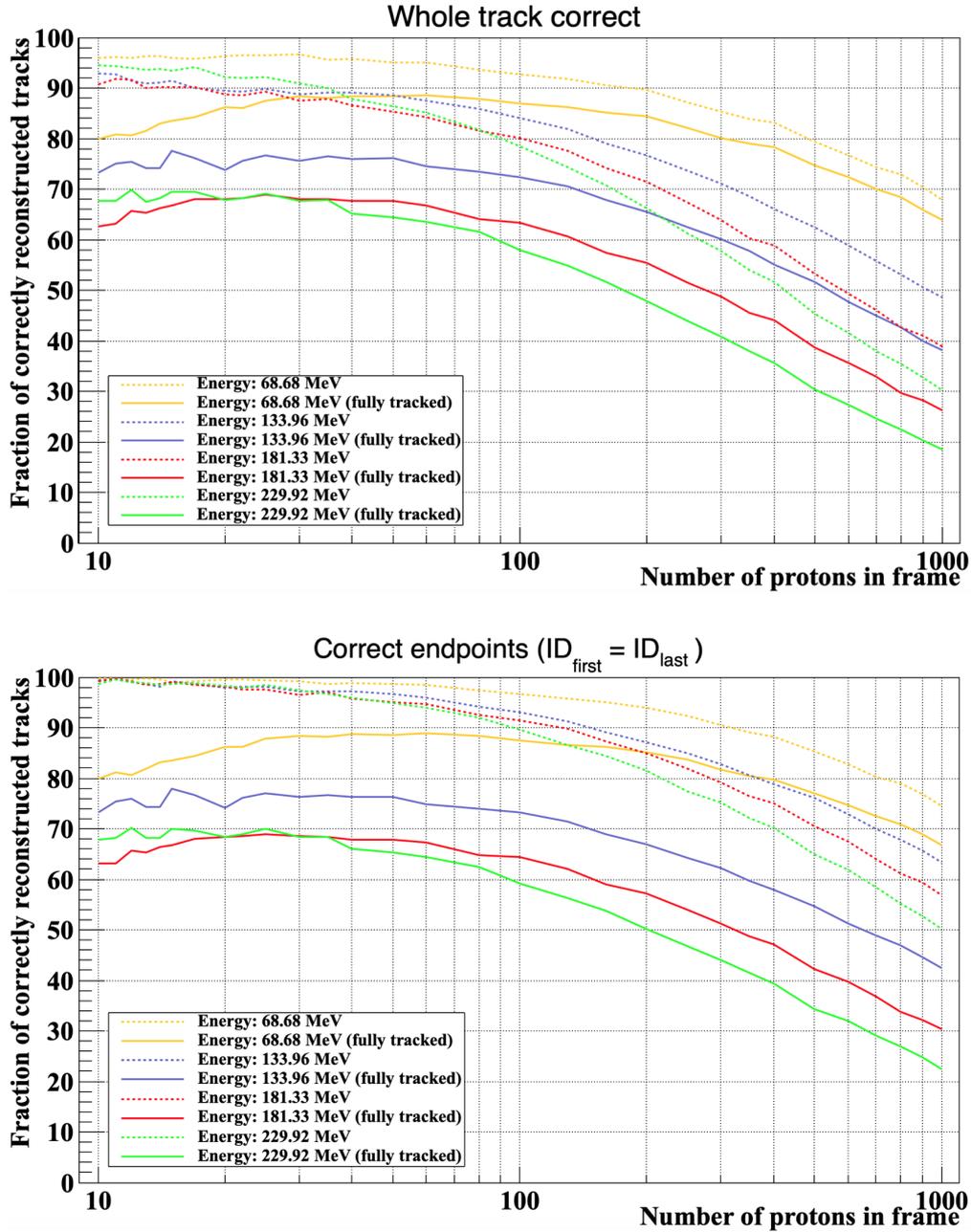


Figure 4.16: Efficiency results when the value of S_{\max} changes throughout the reconstruction of the track depending on the expected scatter. In order to measure the efficiency for various energies, a pencil beam is degraded with a water phantom of different lengths. Data simulated with GATE. The definition of a correctly reconstructed track is given in Section 4.1.

4.4 Robustness of the reconstruction algorithm

The reconstruction algorithm should be equally efficient regardless of the starting point. In all of the previously mentioned methods, the algorithm starts by finding the seeds in the last layer. Then, a seed is chosen and the reconstruction of this specific track begins. Subsequently, another seed is considered and the process is repeated. In order to check if the percentage of correctly reconstructed tracks changes depending on the order in which seeds are selected, two different scenarios were analyzed:

1. Starting from the first seed identified in the layer.
2. Starting from the last seed identified in the layer.

Table 4.2 shows an example of the efficiency of the algorithm depending on the order in which tracks are reconstructed. In addition, Fig. 4.17 shows the resulting track reconstructions. The starting seed is pointed out by a blue arrow. As can be observed, the percentage of correctly reconstructed tracks changes depending on where the origin of the reconstruction process is established. Although there is not a very significant difference, it reflects that the algorithm is not entirely robust.

Position in layer	Same ID all clusters	Same ID all clusters & complete	Same first/last ID	Same first/last ID & complete
First seed	~80%	~67%	~89%	~67%
Last seed	~83%	~71%	~91%	~71%

Table 4.2: Percentage of correctly reconstructed tracks depending on the starting seed order.

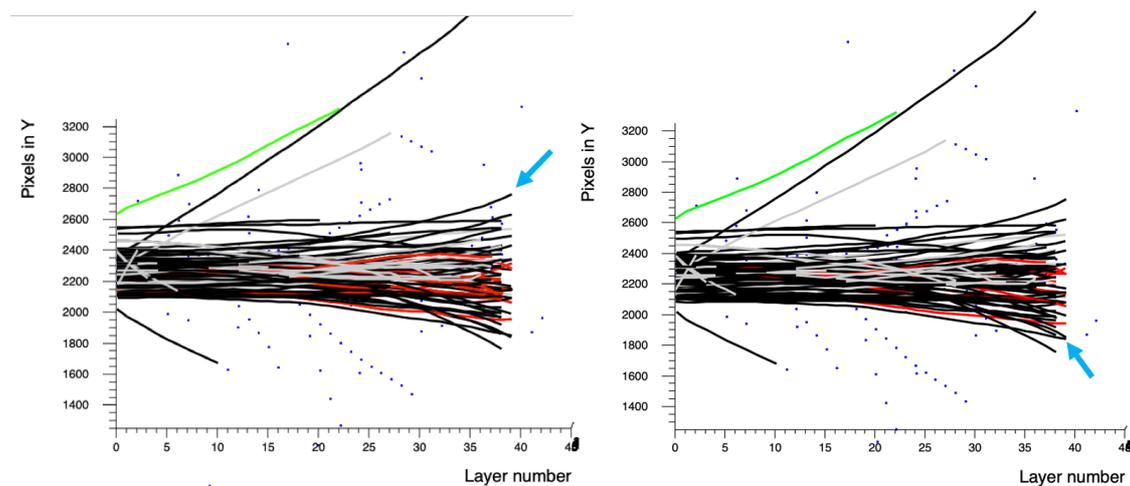


Figure 4.17: Track reconstructed results when the algorithm starts from the first seed it finds (left) and when the last identified seed is established as the origin of the reconstruction process (right). Blue arrows indicate the starting seed for each scenario.

4.5 Proton beam width in the detector

From the efficiency studies, it can be observed that the best results for the same number of particles are obtained for the low-energy proton beam. Vice versa, the lowest efficiency corresponds to the higher energy. One of the reasons is that low-energy protons transverse less layers before they stop. This means that the number of physical interactions with matter is lower, making it easier to reconstruct its trajectory. Another reason is the use of a water phantom of different thicknesses in MC simulations (GATE) to degrade the initial 250 MeV proton beam to the desired mean energy. The use of this degrader increases the width of the proton beam, i.e. the separation between protons at impact is greater in comparison with the distance between particles from a same energetic beam but without the use of the phantom. This is in tune with the proposed usage of the pCT, where the initial proton energy is always the maximum available, and the residual energy is decided by patient/object thickness. Fig. 4.18 shows an example of track reconstruction results using the algorithm explained in Section 4.2 when a water phantom has been used to reduce the energy of the proton beam (initially 250 MeV) and when protons are directly simulated with the desired energy (no degrader). The final energy of a proton beam depends on the thickness of the phantom as it degrades in an inversely manner. Moreover, the resulting width of the beam is directly proportional to the length of the phantom.

Additionally, Fig. 4.19 shows the efficiency graphs for both scenarios when tracks are simulated with GATE and reconstructed with the algorithm explained in Section 4.2. As may be seen, the results obtained when no degrader is used are worse for the same reason explained above: tracks are closer to each other when no degrader is included in the simulation. As expected, the lower energy beam (~ 69 MeV) experiences the largest drop in efficiency since a thicker water phantom is needed (34 cm). However, the reconstruction results of the more energetic proton beam (~ 230 MeV) differ slightly. In this case, only 5 cm of water are needed to get that energy.

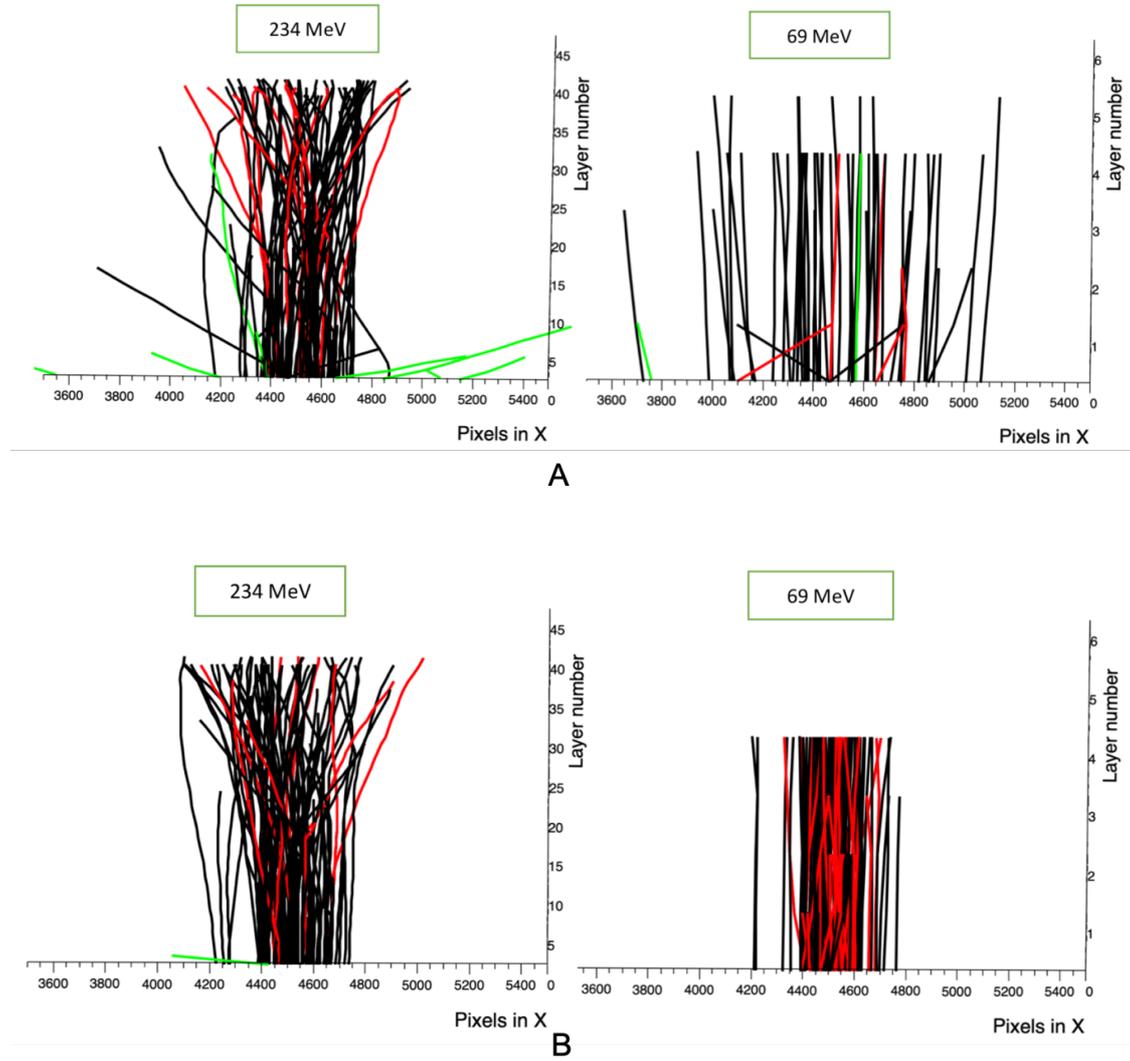


Figure 4.18: Examples of track reconstructions. (A) A water phantom of different thicknesses is used to degrade a 250 MeV proton beam to get a mean energy of ~ 234 MeV (5 cm degrader thickness) and ~ 69 MeV (34 cm degrader thickness). (B) Protons are simulated directly with the intended energies. Data simulated with GATE.

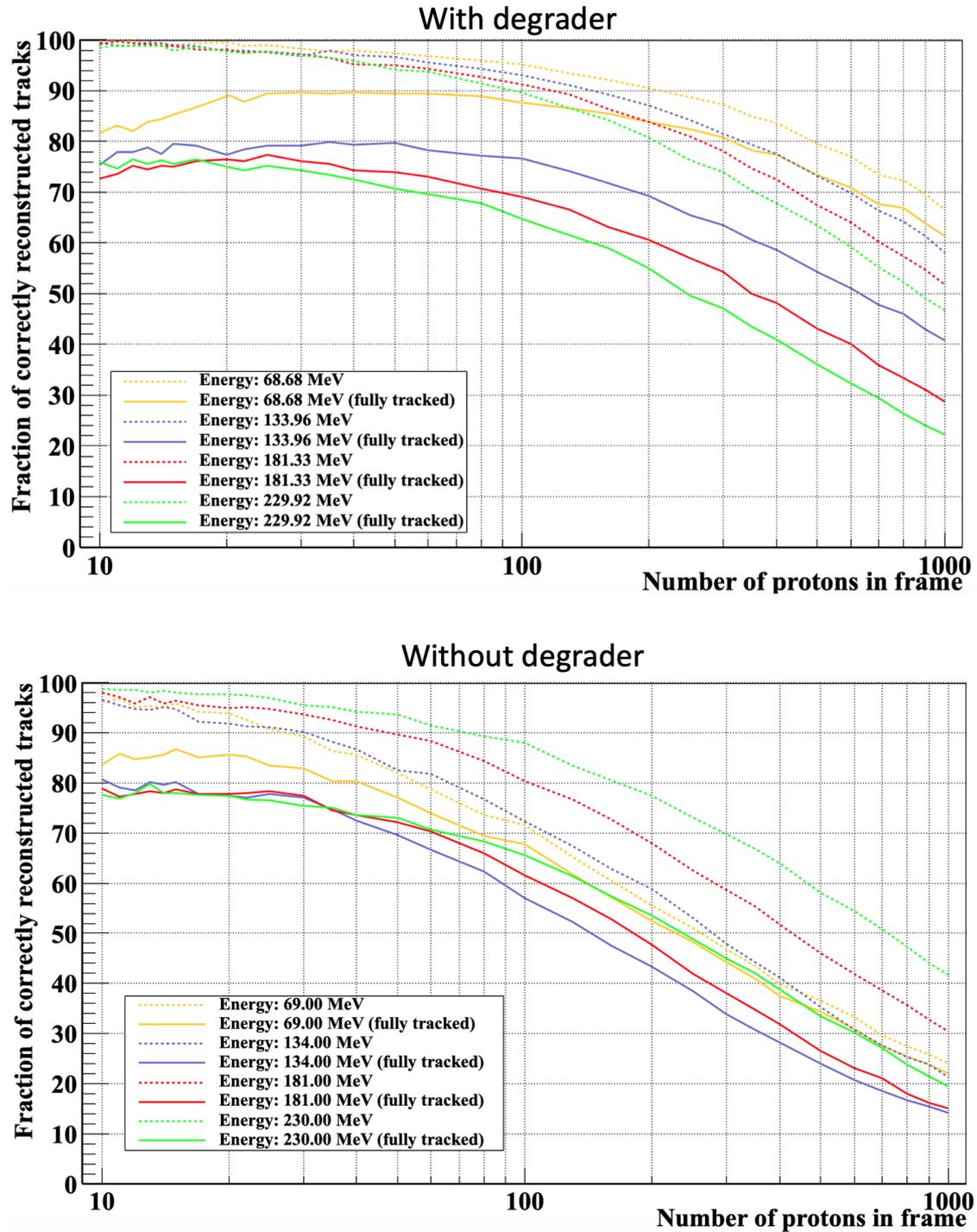


Figure 4.19: Fraction of correctly reconstructed tracks following the definitions in Section 4.1 when a proton beam of energy 250 MeV is degraded by water phantom of different thickness (with degrader) and when protons are simulated directly with the desired energy (without degrader). In both scenarios, tracks simulated with GATE are reconstructed using the algorithm described in Section 4.2.

Chapter 5

Most probable scenario algorithm

The different algorithms proposed until now reconstruct one track at a time. This means that once a cluster has been assigned to a track there is no way that the following tracks can add that cluster, even when the probability that it belongs to another track is greater. This introduces an error in the reconstruction process that is inevitable to solve although a great effort is made in the optimization of the reconstruction parameters. Therefore, an improvement strategy is to calculate the most probably scenario at each layer. In this proposal, the probability that a hit belongs to a track is calculated for all hits and tracks per layer at once. This way, all track combinations are studied before assigning a potential candidate to a track. This approach offers also a potential solution to the lack of robustness of the algorithms presented previously since all tracks are reconstructed at the same time. A schematic of this idea is reflected in Fig. 5.1, where all possible combinations of seeds-clusters are shown for a given case (red frame). Then the probability of all these scenarios is studied before deciding which cluster belongs to which track.

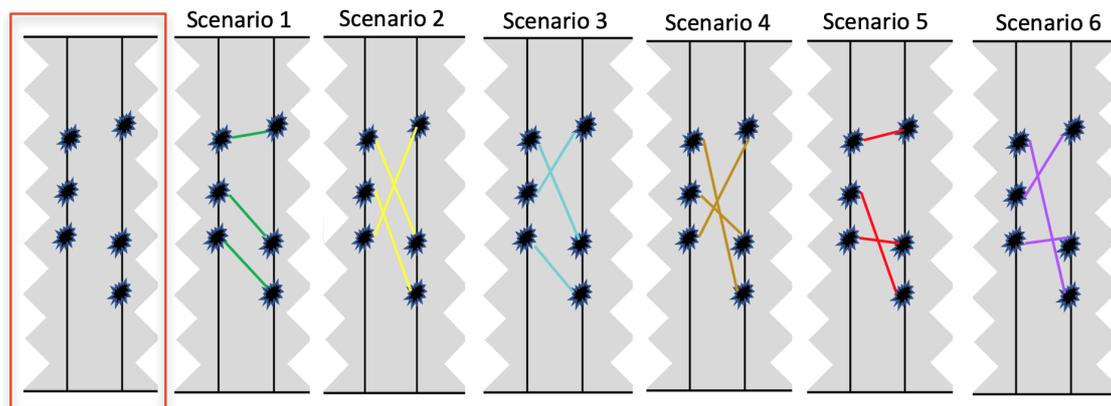


Figure 5.1: Example of the most probable scenario algorithm. Given 3 clusters in a layer and 3 clusters in the next layer, all possible combinations are studied, in this case, 6 different scenarios, represented by different colors.

Starting from the last layer, the scatter angle between the seed and the possible candidate in the previous layer is calculated using the following equation:

$$\theta = \arccos \left(\frac{\vec{u} \cdot \vec{v}}{\|\vec{u}\| \cdot \|\vec{v}\|} \right) \quad (5.1)$$

Where θ is the angular deflection between the actual position of the hit and the extrapolated anticipated position of the last cluster added to the track, and \vec{u} and \vec{v} are the vectors defined in Fig. 5.2. The left panel in Fig. 5.2 represents the general case, while right panel represents the situation at the first layer where no clusters have been added yet, so the particle is assumed to enter perpendicular. Angular deviation is calculated on basis of the difference between the position of the potential candidate and the horizontal axis.

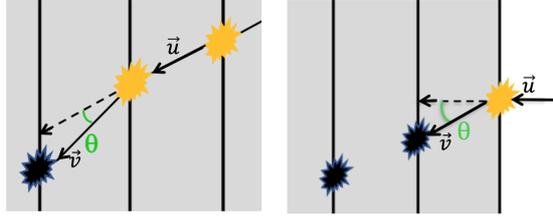


Figure 5.2: Schematic of how the angular deflection θ between two clusters located in consecutive layers is calculated for the general case (left) and when θ is computed between a seed in the first layer and a cluster in the next one, where \vec{u} is assumed to be perpendicular to the detector plane (right).

Following Highland's formula (Eq. 1.2), the expected RMS value of the MCS angle distribution, θ_0 , for a given energy is found. Then, the probability that the seed and a hit belong to the same track is calculated by means of expected scatter and the value of the angular deflection θ :

$$P(\theta_{\min} \leq \theta \leq \theta_{\max}) = \frac{1}{2\pi\theta_0^2} \int_{\theta_{\min}}^{\theta_{\max}} \exp \frac{-x^2}{2\theta_0^2} dx \quad (5.2)$$

where θ_{\min} and θ_{\max} represent the interval in which the probability of θ is calculated. The algorithm assumes that the interaction occurred in the center of the pixel, however, this may occur at any position of the pixel. Therefore, the values of θ_{\max} and θ_{\min} correspond to maximum and minimum angular difference between the extrapolated position of the previous cluster and the position at the pixel where the interaction can occur. Four different positions are evaluated (marked with yellow in Subfigure (A) in Fig. 5.3). These are the most extreme positions of the pixel and therefore also the positions where the maximum and minimum angle will be reached. The angle between \vec{u} and \vec{v} is calculated for the four cases. Finally, θ_{\min} corresponds to the smallest angle calculated (Subfigure (B) in Fig. 5.3) while θ_{\max} is the largest angle (Subfigure (C) in Fig. 5.3).

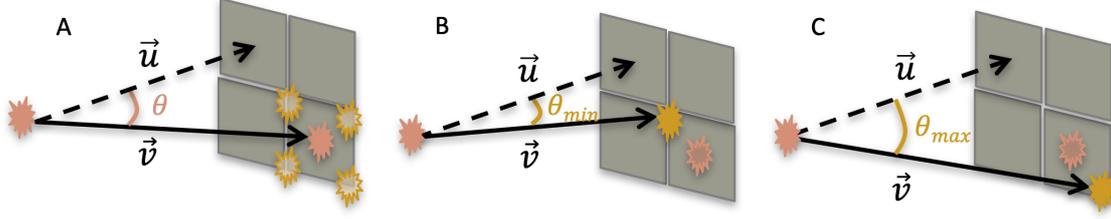


Figure 5.3: Two hits in consecutive layers are represented in pink. The four corners of the pixel in which the second cluster was detected are represented as yellow stars in (A). The angular deflection, θ , is the difference between initial direction of the first hit and the angle between first hit and second hit. The smallest value is assigned to θ_{\min} (B) while the largest one to θ_{\max} (C).

The calculation of $P(\theta_{\min} \leq \theta \leq \theta_{\max})$ is calculated at once for all seed-cluster pairs in the scenario under study. Assuming that these events are independent (i.e. the interaction position of a proton in each layer does not influence where the other protons will impact in the detector), the total probability of that scenario is computed by multiplying the probabilities of each event:

$$P_{\text{total}} = \prod^n P(\theta_{\min} \leq \theta_n \leq \theta_{\max}) \quad (5.3)$$

where n represents the number of potential clusters in the layer under study.

P_{total} is calculated for all possible scenarios and the one with the highest score determines which cluster will be added to each track.

5.1 Optimized version

The number of possible seed-cluster combinations increases with the factorial of the number of seeds. That means if 100 tracks are simulated, the number of scenarios that need to be analyzed is $100! (= 9.332622 \times 10^{157})$. Naturally, such numbers of combinations are extremely computationally expensive and renders the reconstruction process impractical.

Since many of the combinations are highly unlikely to occur, the number of scenarios to be evaluated can be reduced. One method of conveying the problem of computational requirement is to narrow down the search criteria. A search cone with an angle of $4 \cdot \theta_0$ applied to each seed in order to identify possible matches would in effect mean that possible combinations are calculated using only the most adjacent clusters.

In order to verify the effectiveness of this optimization, its efficiency is compared to the efficiency of the non-optimized version and the efficiency of the algorithm described in Section 4.2. For this purpose, 8 tracks simulated with PROCASIM are reconstructed. In order to make a fair comparison, the proton density of this simulation corresponds to the maximum number of tracks that the non-optimized version of

the proposed algorithm can efficiently reconstruct. Since the number of tracks to be reconstructed can not be increased due to this computational limitation, the distance between them in the detector is 10 times reduced (in comparison with the values normally used when the same number of tracks are simulated), in order to hinder the reconstruction of the tracks. Left panel in Fig. 5.4 shows the reconstructed tracks when the algorithm explained in Section 4.2 is applied, while right panel illustrates the result when both versions of the proposed algorithm are used to reconstruct the tracks. As can be observed, better results are achieved with the new reconstruction algorithm, even when the optimized version is used.

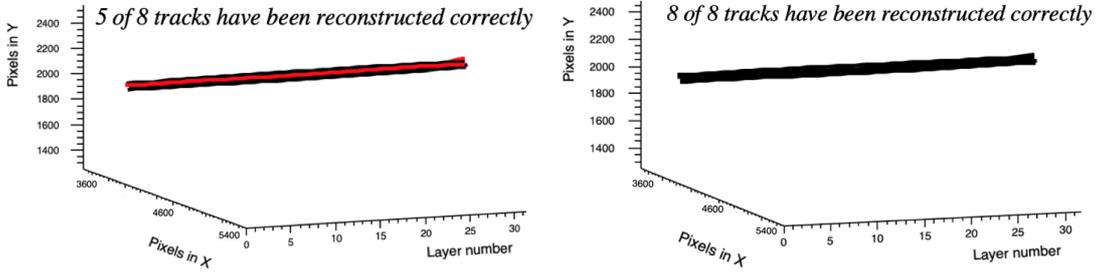


Figure 5.4: Track reconstruction result when: algorithm explained in Section 4.2 is used (left) and when most probable scenario algorithm (both original and optimized version) is applied (right). In both studies, 8 tracks are simulated using PROCASIM.

However, as the number of simulated tracks increases, the number of potential clusters also increases. Even a narrowed search cone with angle $4 \cdot \theta_0$ will eventually include too many clusters. Now, the number of possible combinations is $\prod_1^n N_{PC}$, where n is the number of seeds and N_{PC} the number of potential clusters for an specific seed. So, if two clusters are found per seed and 100 tracks are simulated, the number of combinations to be analyzed will be: $2^{100} = 1.2676506 \times 10^{30}$ and therefore still requires unacceptable high computational power.

For practical applications, further cost reduction is required. Hence, the angle of the search cone is reduced to $2 \cdot \theta_0$. If no clusters are found for that seed, this value is increased by θ_0 . This process is repeated until the seed can find at least a potential cluster. If the angle of the research cone reaches a value of $10 \cdot \theta_0$ and the seed is not able to identify any cluster, this seed will not be used for the calculation of the most likely scenario. Fig. 5.5 shows the main steps of the algorithm. Red arrow shows the path to be followed for this optimized version.

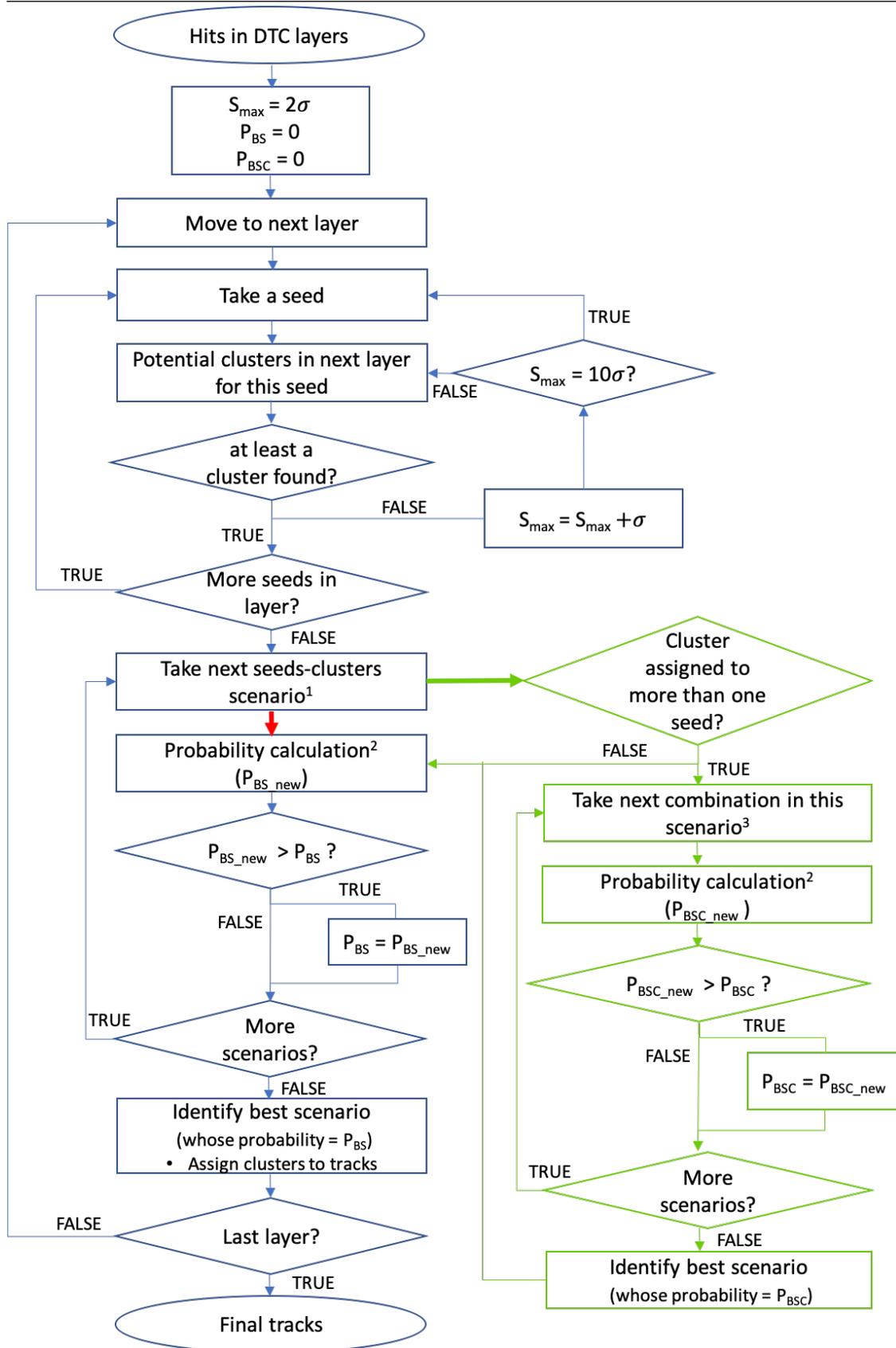


Figure 5.5: Flowchart of the most probable scenario algorithm. Red arrow shows the optimized version path (Section 5.1). Green arrow points the path to the adapted version to MC simulations (Section 5.2).

¹This step is described in Fig. 5.1.

²Following Eq. 5.3.

³This step is described in Fig. 5.9.

Fig. 5.6 shows the percentage of tracks simulated with PROCASIM with the same ID in the first and last clusters added. The fraction of tracks with the same ID in all clusters is not shown since similar results are obtained. Only $\sim 1 - 2\%$ of the tracks are wrong in the middle but regain the correct entrance points. Moreover, tracks that are reconstructed correctly but are incomplete are not represented either because the difference with the fully tracked ones is minimal.

Although in some cases higher efficiency is achieved when the energy of the proton beam is lower, overall, the difference is almost negligible.

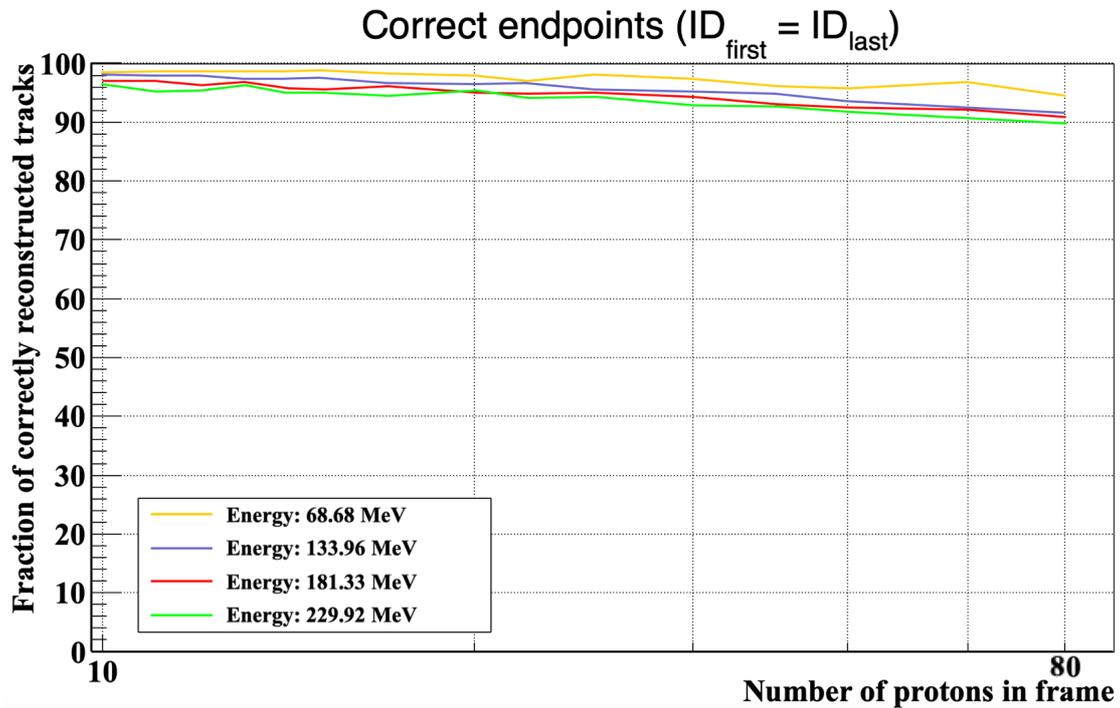


Figure 5.6: Fraction of tracks whose start- and endpoints have the same eventID for different proton densities when the most probable scenario is calculated before assigning each cluster to the respective track. Data simulated with PROCASIM.

As explained in Section 2.2.2, in PROCASIM, each particle is generated from a single point in the first layer of the calorimeter. From this starting point, a stochastic trajectory is generated taking into account only the angular deviation produced by MCS. However, sometimes, some trajectories are created very close, resulting in more than one confusing cluster at the time of the reconstruction process. This deteriorates the efficiency of the algorithm. The effect can be observed in Fig. 5.7, where two tracks (marked with green arrows) are generated so close to each other that in certain cases, the probability that a cluster belongs to the wrong track is greater than the probability that it belongs to the correct track.

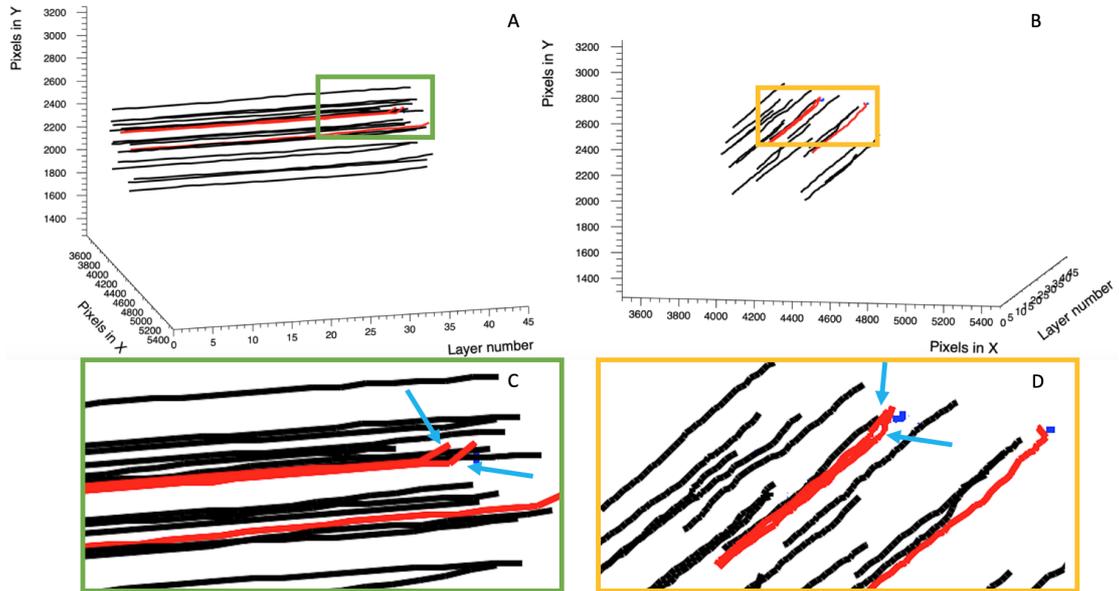


Figure 5.7: Example of a type of error in the reconstruction process due to the proximity of two tracks. Subfigures (A) and (B) show the same reconstruction result from two different points of view. Subfigures (C) and (D) show the zoomed image of the area selected in Subfigures (A) and (B) with green and yellow frames respectively. The two tracks that introduce confusion in the reconstruction process are marked with blue arrows. As can be seen, the tracks are so close to each other that it is difficult to differentiate them in most of their trajectory.

5.2 Adaptation to Monte Carlo simulations

So far, the effectiveness of the algorithm has only been demonstrated using tracks simulated with PROCASIM, where only MCS is taken into account. However, other types of physical interactions occur in real-life situations (e.g. elastic and non elastic nuclear interactions), therefore the algorithm has also been tested with MC-based simulations.

One of the main problems when trying to reconstruct tracks using the described algorithm is the existence of single hits in different layers of the detector, represented as blue dots in Fig. 5.8. They correspond to secondary particles whose origin may vary: stray photons, delta electrons, neutrons. Some of these hits (marked with red arrows) introduce errors when calculating the probabilities, since the same cluster is assigned to two or more seeds.

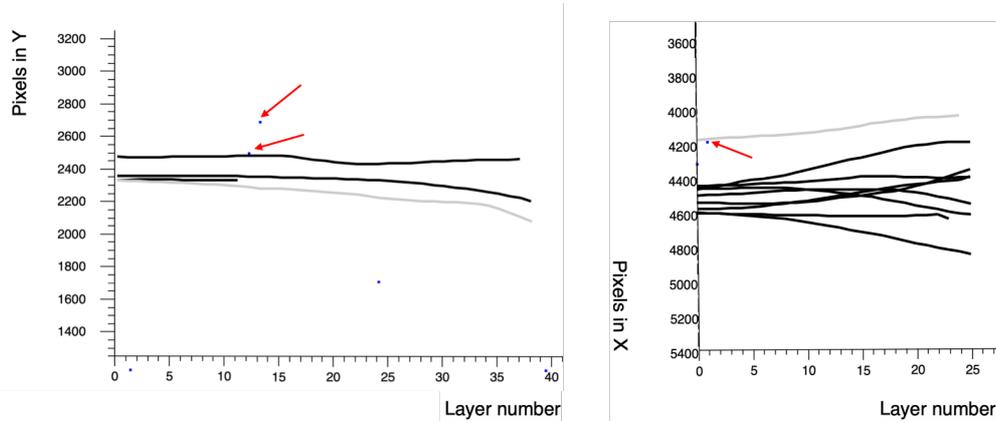


Figure 5.8: Examples of the reconstruction result when a energy pencil beam of 250 MeV is degraded by 5 cm (left) and 16 cm (right) of water. During the reconstruction process, the algorithm has to deal with single hits (marked with red arrows) that can introduce errors and cause tracks to be reconstructed erroneously.

An equal number of scenarios and seeds that share the same cluster are studied to determine the most likely seed. In each of them, the cluster is assigned only to one seed, while the rest of the seeds are not considered. Then the probability of this cluster-seed combination is calculated. Subfigure (A) in Fig. 5.9 shows an example of this situation when a cluster is assigned to two hits (one of them is already part of a track (in yellow) and a single hit (in blue)). In case more than one cluster is assigned to different seeds, all possible combinations are studied, as can be seen in Subfigure (B) in Fig. 5.9. In this example, two clusters can be assigned to two seeds each. In total four possible combinations are analyzed from an original scenario. In the end, the most likely combination is chosen. Additionally, Fig. 5.5 shows the main steps of the algorithm. The path to be followed for this new version is indicated by a green arrow.

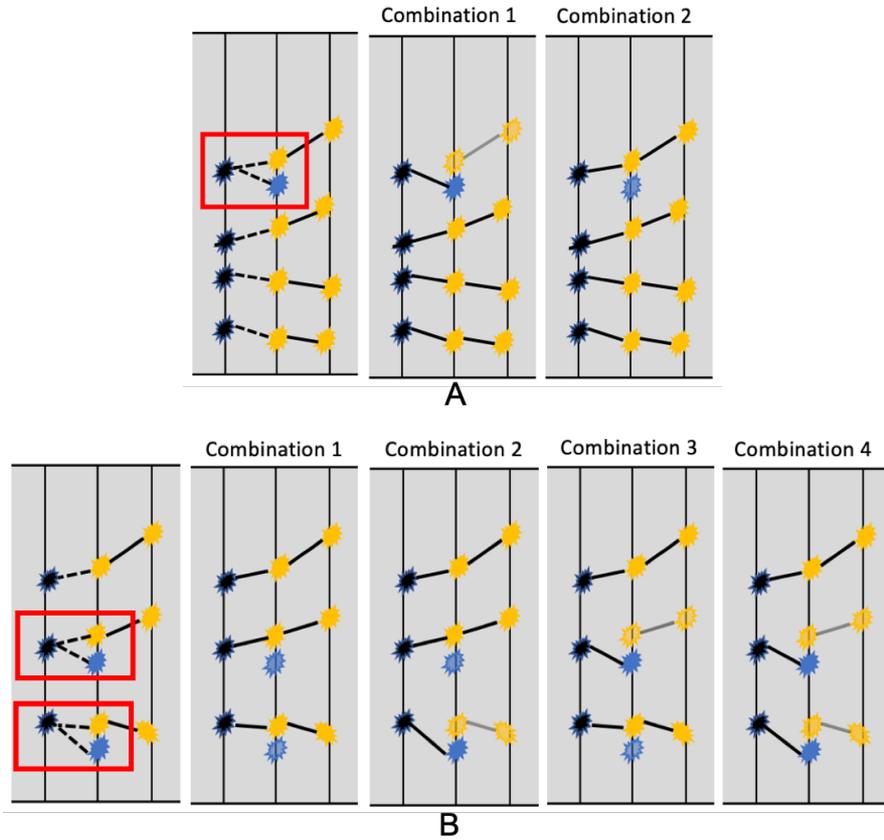


Figure 5.9: Schematic of the most probable seed-cluster selection process. In case the same cluster is assigned to multiple hits in the prior layer, different probabilities of the whole scenario (e.g. all combination between seeds-clusters) are calculated combining each time the shared cluster with one of the hits, while the rest are not linked to any cluster. In (A) a cluster (black star in red frame) is shared by two hits in the previous layer (yellow and blue stars in red frame), so the probability of two different scenarios are calculated. The same process is repeated in (B), where two clusters (black stars in red frames) match with more than one seed in the prior layer (yellow and blue stars in red frames). In this scenario, the probability of four different combinations are computed. In the end, the most likely combination is chosen.

In order to study the efficiency of this algorithm with data simulated with GATE, the number of tracks reconstructed correctly is analyzed for different proton densities. Fig. 5.10 shows the percentage of tracks that have the same eventID in the first and last entry. The fraction of tracks with the same ID in all clusters is not shown since the efficiency difference with respect to the results shown in Fig. 5.10 is less than 2%.

These results follow some trends explained in Chapter 4. Higher energy results in lower efficiency and the difference between the fully tracked and incomplete tracks become larger at higher energies. However, greater efficiency can be observed in comparison with the results shown in Chapter 4.

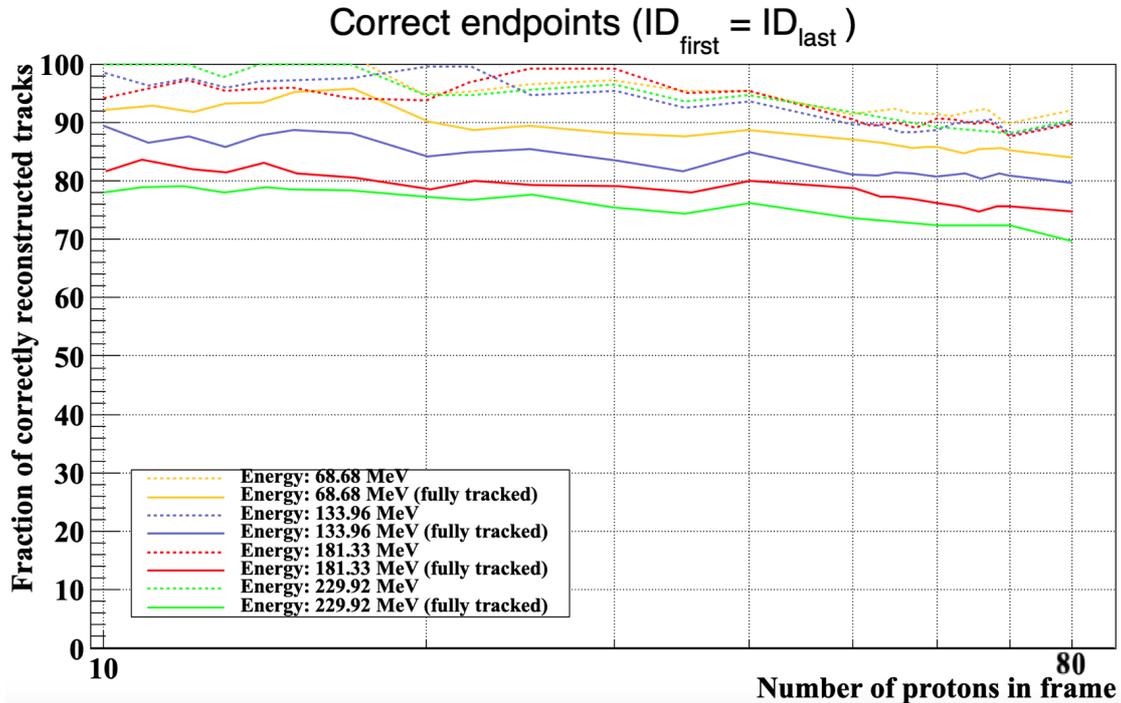


Figure 5.10: Study of the fraction of tracks whose start- and endpoints have the same eventID for different proton densities and energies. Data simulated with GATE.

5.3 Discussion and conclusions

In this chapter a new reconstruction algorithm based on the identification of the most probable scenario is discussed. As can be seen in Figures 5.6 and 5.10, the number of tracks that are reconstructed correctly is considerably greater than when the same beam intensities are reconstructed with the algorithms explained in Chapter 4.

However, one of the main disadvantages of this algorithm is the time required for the analysis and reconstruction of the tracks. As the proton density rises, the number of combinations that the algorithm has to analyze also increases. Therefore, the computational efficiency of this algorithm for large intensities is much lower compared to the other algorithms previously described. For this reason, the maximum number of tracks that can be reconstructed is ~ 80 regardless of the simulator used. The studies were carried out in a MacBook Pro 13", with 8 GB RAM and Intel Core i5 CPU at 2.7 GHz, utilizing a single core for the task.

For example, the time required for the concurrent reconstruction of 80 protons varies between 10-30 min, depending on the energy of the protons (and therefore, the number of layers). On the contrary, the reconstruction of the same number of particles with the algorithms described in previous chapter takes a few seconds.

Considering that the electronic readout time of the DTC is $5 \mu\text{s}$ and assuming that 80% of the tracks should be reconstructed correctly (this value has to be adjusted to reach a compromise between image quality / noise (due to fake tracks) and image

acquisition time), which correspond to 80 protons per readout frame for a 133.96 MeV proton beam (Fig. 5.10), the number of protons that should be reconstructed simultaneously per second is 16 millions, however, the algorithm needs ~ 15 min to reconstruct the protons measured per readout frame, which means 3×10^7 minutes to reconstruct one second of acquisition. For this reason, the time needed for the track reconstruction process is a key factor when choosing the most suitable algorithm.

All in all, although efforts have to be made in order to achieve a greater speed of reconstruction, efficiency results show a great potential, opening the door to a new track reconstruction method based on the study of the most probable scenario.

Chapter 6

Discussion and conclusions

Over the past few years, the use of proton therapy to treat cancer has increased, optimizing the dose deliver to patients. However, in order to get an accurate treatment planning, the use of a pCT is needed. To this end, the proton's residual energies can be measured using a pixel-based calorimeter behind the patient. However, due to stochastic trajectories followed by protons, an effective track reconstruction algorithm is required to handle reconstruction of multiple tracks per readout cycle.

The current proton track reconstruction algorithm presented by Pettersen [1] has some limitations, especially at higher beam intensities. Therefore, an optimization of this algorithm would ensure a better performance.

Since the calorimeter is still under development, the whole study was performed using two types of simulations: GATE and PROCASIM. MC simulations (GATE) were used to generate and track particles that undergo different physical interactions (e.g. MCS, elastic nuclear interaction, inelastic nuclear interactions...). PROCASIM was based on the generation of stochastic trajectories of protons taking into account only the angular deviation produced by MCS. Both tools are used to develop and study the performance of different proton tracking algorithms.

In order to analyze and compare the different reconstruction algorithms proposed in this thesis, a study of their efficiency was conducted. Being the efficiency the number of correctly reconstructed tracks, identified through checks against the event ID from simulations. From these results, several common trends can be observed. First of all, the higher the number of protons, the lower the probability that all hits in a given reconstructed track originate from the same primary proton. Better efficiency is reached when the number of tracks to be reconstructed per readout is lower than 100. Secondly, better results are obtained for the less energetic proton beam. The main reason are that less layers are traversed by low-energy protons and the use of a water phantom of different thicknesses in MC simulations to degrade the initial 250 MeV proton beam to the desired mean energy. This increases the width of the proton beam, facilitating the track reconstruction process.

Overall, the fraction of tracks reconstructed correctly is higher when particle trajectories are simulated with PROCASIM because the angular deviation that the protons undergo is, in general, smaller and more gradual than in the case of particles simulated with GATE. Therefore, following the method outlined in Fig. 5.2, the position of the next cluster in the subsequent layer is closer to the estimated position, increasing the probability that it is assigned to the correct track. Regarding the algorithm with the higher efficiency, studying the different combinations before assigning each cluster to the corresponding track results in a greater number of tracks reconstructed correctly for at least a beam density of up to 80 protons. This is the maximum number of tracks that the algorithm can reconstruct. With this proposal (Chapter 5), the order in which tracks are reconstructed does not influence the final result, making it a much more robust algorithm than the rest of algorithms explained in the present work. However, although efforts have been made regarding the computational efficiency of the code, it is still very computational inefficient. This renders its use relatively limited in real situations where $\sim 1 - 25$ millions of particles need to be reconstructed per second. On the other hand the algorithm with a constant S_{\max} offers greater efficiency especially up to a beam densities of 200 - 300 protons. However, it comes at the cost of robustness. For a larger number of protons, the algorithm where the value of S_{\max} varies depending on the expected scatter provides a larger fraction of correctly reconstructed tracks.

In terms of the most efficient algorithm when tracks are simulated using GATE, the concurrent study of the most probable combinations between clusters in two consecutive layer occupies the first position. However, as with the results obtained with PROCASIM, the computational inefficiency limits the number of tracks that can be reconstructed, with 80 being the maximum allowed. As for the rest of the proposals, the use of a variable value of S_{\max} depending on the expected scatter, increases the number of correctly reconstructed tracks. However, the percentage of complete tracks is minor compared to the results obtained when a constant S_{\max} is used, especially for a beam intensity less than 400 protons per readout. The main reason for this decrease is that, although most of the scatter is small-angle due to MCS, there are high-angle contributions in the first layers due to nuclear interactions.

In conclusion, although all the proposed algorithms exceed the efficiency of the current algorithm, especially at high beam intensities, the most viable algorithm seems to be the one presented in Section 4.2. In this algorithm, the value of S_{\max} is calculated based on the particle density and the pencil beam's size, and is constant throughout the reconstruction process. Although it is not very robust, since its efficiency depends on its starting point, it is the algorithm that, with results similar to others presented in this project, is the most computationally efficient, which is a key factor in quasi-online reconstruction. However, there are still limitations on the track density that can be reconstructed correctly. Therefore, the probability approach explained in Chapter 5 shows a great potential. If the algorithm is optimized, a significant improvement in efficiency could be achieved at higher beam intensities.

Chapter 7

Limitations and recommendations for future work

Although different proton track reconstruction proposals for a DTC prototype have been presented in this thesis, and a considerable improvement has been achieved, there are still some limitation. The algorithm that analyzes the most probable scenario before assigning the corresponding cluster to each track seems to give better results, however, its computational inefficiency precludes its use. Therefore, a deeper optimization of the code needs to be carried out. One option can be the execution in parallel on several CPU cores of the reconstruction track, decreasing the required time spent on the analysis. Moreover, the computational efficiency of the algorithm can be tested in more powerful computers than the one used in this study (MacBook Pro 13", with 8 GB RAM and Intel Core i5 CPU at 2.7 GHz).

Another limitation found in the reconstruction results is the large difference found, in most cases, between all tracks reconstructed correctly and those that are also complete. Therefore, an improvement strategy of the code could be to identify the cause and try to increase the percentage of fully reconstructed tracks.

During this thesis, for the sake of simplicity, the deposited energy is directly used in the track reconstruction as read out from the MC simulator. However, in a real situation, large clusters with sizes varying between 1 and 35 pixels are activated by the charge diffusion of electron-hole pairs, created by the interaction between protons and the detector [1]. The number of pixel activated by charge diffusion is proportional to the energy deposited by the proton, and some clusters can merge, complicating the reconstruction process. Future work should focus on the implementation of a charge clustering model and the consequent reconstruction process adaptation. For example, clusters should not be removed from the search pool when a track has been reconstructed because that cluster may contain more than one hit.

Additionally, one potential strategy to improve the track reconstruction algorithm is based on forward and backward reconstruction. The tracking can be performed simultaneously in both directions, track segments that match are stored while track segment that do not match can be treated with more care.

Bibliography

- [1] H. E. S. Pettersen. *A Digital Tracking Calorimeter for Proton Computed Tomography*. PhD thesis, University of Bergen, Norway, February 2018.
- [2] C. Bopp. *The proton as a dosimetric and diagnostic probe*. PhD thesis, Université de Strasbourg, 2014.
- [3] S. Lin. The utility of proton beam therapy with concurrent chemotherapy for the treatment of esophageal cancers. *Cancers*, 3:4090–101, 12 2011.
- [4] P. Piersimoni. A high-granularity digital tracking calorimeter optimized for proton ct. 2019.
- [5] W. D. Newhauser and R. Zhang. The physics of proton therapy. *Physics in Medicine and Biology*, 60(8):R155–R209, mar 2015.
- [6] H. E. S. Pettersen, I. Meric, O. Harlad Odland, H. Shafiee, J. Rambo Sølve, and D. Röhrich. Proton tracking algorithm in a pixel based range telescope for proton computed tomography. *EPJ Web of Conferences*, *In press*, 2018.
- [7] N. Krahl, F. Khellaf, J. M. Létang, S. Rit, and I. Rinaldi. A comprehensive theoretical comparison of proton imaging set-ups in terms of spatial resolution. *Physics in Medicine & Biology*, 63(13):135013, jul 2018.
- [8] M. Frandesand V. Maxim and R. Prost. Analytical inversion of the Compton transform using the full set of available projections. *Inverse Problems*, 25:095001, 08 2009.
- [9] Worldhealthorganization. <https://www.who.int/news-room/fact-sheets/detail/cancer>. Accessed: 2019-05-11.
- [10] M. Durante and J. S. Loeffler. Charged particles in radiation oncology. *Nature Publishing Group*, 12 2009.
- [11] F. Dionisi and E. Ben-Josef. The use of proton therapy in the treatment of gastrointestinal cancers: Liver. *Proceedings of the National Academy of Sciences*, 2014.

-
- [12] D. J. Indelicato et al. Incidence and dosimetric parameters of pediatric brain-stem toxicity following proton therapy. *Acta Oncologica*, 53(10):1298–1304, 2014. PMID: 25279957.
- [13] C. A. Collins-Fekete. *On particle imaging with application to particle radiotherapy*. PhD thesis, Quebec, Canada, 2017.
- [14] H. Paganetti. Range uncertainties in proton therapy and the role of monte carlo simulations. *Physics in medicine and biology*, 57(11):99–117, 2012.
- [15] M. Yang, X. R. Zhu, P. C. Park, U. Titt, R. Mohan, G. Virshup, J. E. Clayton, and L. Dong. Comprehensive analysis of proton range uncertainties related to patient stopping-power-ratio estimation using the stoichiometric calibration. *Physics in Medicine and Biology*, 57(13):4095–4115, jun 2012.
- [16] O. Jäkel and P. Reiss. The influence of metal artefacts on the range of ion beams. *Physics in medicine and biology*, 52:635–44, 03 2007.
- [17] S. Dowdell, C. Grassberger, G. C. Sharp, and H. Paganetti. Interplay effects in proton scanning for lung: a 4d monte carlo study assessing the impact of tumor and beam delivery parameters. *Physics in Medicine and Biology*, 58(12):4137–4156, may 2013.
- [18] H. Paganetti. Proton beam therapy. In *Proton Beam Therapy*, 2399-2891. IOP Publishing, 2017.
- [19] R. P. Johnson. Review of medical radiography and tomography with proton beams. *Reports on Progress in Physics*, 81(1):016701, nov 2017.
- [20] E. Bär et al. Experimental validation of two dual-energy ct methods for proton therapy using heterogeneous tissue samples. *Medical Physics*, 45(1):48–59, 2018.
- [21] N. Arbor et al. Monte carlo comparison of x-ray and proton ct for range calculations of proton therapy beams. *Physics in medicine and biology*, 60(19):7585–99, 10 2015.
- [22] P. J. Doolan, M. Testa, G. Sharp, E. H. Bentefour, G. Royle, and H. M. Lu. Patient-specific stopping power calibration for proton therapy planning based on single-detector proton radiography. *Physics in Medicine and Biology*, 60(5):1901–1917, feb 2015.
- [23] A. M. Koehler. Proton radiography. *Science*, 160(3825):303–304, 1968.
- [24] R. Schulte et al. Density resolution of proton computed tomography. *Medical Physics*, 32(4):1035–1046, 2005.

-
- [25] J. T. Taylor, C. Waltham, T. Price, N. M. Allinson, P. P. Allport, G. L. Casse, A. Kacperek, S. Manger, N. A. Smith, and I. Tsurin. A new silicon tracker for proton imaging and dosimetry. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 831:362 – 366, 2016. Proceedings of the 10th International “Hiroshima” Symposium on the Development and Application of Semiconductor Tracking Detectors.
- [26] N. Krah, V. Patera, S. Rit, A. Schiavi, and I. Rinaldi. Regularised patient-specific stopping power calibration for proton therapy planning based on proton radiographic images. *Physics in Medicine & Biology*, 64(6):065008, mar 2019.
- [27] H. Bethe. Zur theorie des durchgangs schneller korpuskularstrahlen durch materie. *Annalen der Physik*, 397(3):325–400, 1930.
- [28] G. Molière. Theorie der Streuung schneller geladener Teilchen I. Einzelstreuung am abgeschirmten Coulomb-Feld. *Zeitschrift Naturforschung Teil A*, 2:133–145, March 1947.
- [29] G. Molière. Theorie der Streuung schneller geladener Teilchen II. Mehrfach- und Vielfachstreuung. *Zeitschrift Naturforschung Teil A*, 3:78–97, February 1948.
- [30] V. L. Highland. Some Practical Remarks on Multiple Scattering. *Nucl. Instrum. Meth.*, 129:497, 1975.
- [31] M. Scaringella, M. Brianzi, M. Bruzzi, M. Bucciolini, M. Carpinelli, G.A.P. Cirrone, C. Civinini, G. Cuttone, D. Lo Presti, S. Pallotta, C. Pugliatti, N. Randazzo, F. Romano, V. Sipala, C. Stancampiano, C. Talamonti, M. Tesi, E. Vanzi, and M. Zani. The prima (proton imaging) collaboration: Development of a proton computed tomography apparatus. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 730:178 – 183, 2013. Proceedings of the 9th International Conference on Radiation Effects on Semiconductor Materials Detectors and Devices.
- [32] R. P. Johnson, V. Bashkirov, L. DeWitt, V. Giacometti, R. F. Hurley, P. Pierimoni, T. E. Plautz, H. F. . Sadrozinski, K. Schubert, R. Schulte, B. Schultze, and A. Zatserklyaniy. A fast experimental scanner for proton ct: Technical performance and first experience with phantom scans. *IEEE Transactions on Nuclear Science*, 63(1):52–60, Feb 2016.
- [33] G. Aglieri Rinella. The alpine pixel sensor chip for the upgrade of the alice inner tracking system. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 845:583 – 587, 2017. Proceedings of the Vienna Conference on Instrumentation 2016.

-
- [34] H. E. S. Pettersen, J. Alme, A. Biegun, A. van den Brink, M. Chaar, D. Fehlker, I. Meric, et al. Proton tracking in a high-granularity digital tracking calorimeter for proton ct purposes. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, page 51–61, 11 2017.
- [35] S. Agostinelli et al. Geant4—a simulation toolkit. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 506(3):250 – 303, 2003.
- [36] V. N. Ivanchenko, O. Kadri, M. Maire, and L. Urban. Geant4 models for simulation of multiple scattering. *Journal of Physics: Conference Series*, 219(3):032045, apr 2010.
- [37] R. Brun and F. 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):81 – 86, 1997. New Computing Techniques in Physics Research V.
- [38] G. E. P. Box and M. E. Muller. A note on the generation of random normal deviates. *Ann. Math. Statist.*, 29(2):610–611, 06 1958.
- [39] V. Sipala, N. Randazzo, S. Aiello, M. Bruzzi, M. Bucciolini, M. Carpinelli, G.A.P. Cirrone, C. Civinini, G. Cuttone, E. Leonora, D. Lo Presti, S. Pallotta, C. Pugliatti, M. Scaringella, C. Stancampiano, C. Talamonti, and E. Vanzi. Design and characterisation of a YAG(ce) calorimeter for proton computed tomography application. *Journal of Instrumentation*, 10(03):C03014–C03014, mar 2015.
- [40] J. T. Taylor et al. An experimental demonstration of a new type of proton computed tomography using a novel silicon tracking detector. *Medical Physics*, 43(11):6129–6136.
- [41] V. A. Bashkirov, R. W. Schulte, R. F. Hurley, R. P. Johnson, H. F.-W. Sadrozinski, A. Zatserklyaniy, T. Plautz, and V. Giacometti. Novel scintillation detector design and performance for proton radiography and computed tomography. *Medical Physics*, 43(2):664–674.
- [42] E. Rocco. Highly granular digital electromagnetic calorimeter with maps. *Nuclear and Particle Physics Proceedings*, 273-275:1090 – 1095, 2016. 37th International Conference on High Energy Physics (ICHEP).
- [43] M. Esposito, T. Anaxagoras, P.M. Evans, S. Green, S. Manolopoulos, J. Nieto-Camero, D.J. Parker, G. Poludniowski, T. Price, C. Waltham, and N.M. Allinson. CMOS active pixel sensors as energy-range detectors for proton computed tomography. *Journal of Instrumentation*, 10(06):C06001–C06001, jun 2015.

-
- [44] R. P. Johnson, V. A. Bashkirov, G. Coutrakon, V. Giacometti, P. Karbasi, N. T. Karonis, C. E. Ordoñez, M. Pankuch, H. F. W. Sadrozinski, K. E. Schubert, and R. W. Schulte. Results from a prototype proton-ct head scanner. *Physics Procedia*, 90:209 – 214, 2017. Conference on the Application of Accelerators in Research and Industry, CAARI 2016, 30 October – 4 November 2016, Ft. Worth, TX, USA.
- [45] A. Charles F. Collins, S. Brousmiche, K. N. P. Stephen, L. Beaulieu, and J. Seco. A maximum likelihood method for high resolution proton radiography/proton CT. *Physics in Medicine and Biology*, 61(23):8232–8248, nov 2016.
- [46] J. T. Taylor, P. P. Allport, G. L. Casse, N. A. Smith, I. Tsurin, N. M. Allinson, M. Esposito, A. Kacperek, J. Nieto-Camero, T. Price, and C. Waltham. Proton tracking for medical imaging and dosimetry. *Journal of Instrumentation*, 10(02):C02015–C02015, feb 2015.
- [47] K. Wong et al. The effect of tissue inhomogeneities on the accuracy of proton path reconstruction for proton computed tomography. *AIP Conference Proceedings*, 1099, 03 2009.
- [48] R. W. Schulte, S. N. Penfold, J. T. Tafas, and K. E. Schubert. A maximum likelihood proton path formalism for application in proton computed tomography. *Medical Physics*, 35(11):4849–4856.
- [49] S. Rit, G. Dedes, N. Freud, D. Sarrut, and J.M. Létang. Filtered backprojection proton ct reconstruction along most likely paths. *Medical Physics*, 40(3):031103.
- [50] G. Poludniowski, N.M. Allinson, and P.M. Evans. Proton radiography and tomography with application to proton therapy. *The British Journal of Radiology*, 88(1053):20150134, 2015. PMID: 26043157.
- [51] G. Poludniowski, N. M. Allinson, and P. M. Evans. Proton computed tomography reconstruction using a backprojection-then-filtering approach. *Physics in Medicine and Biology*, 59(24):7905–7918, nov 2014.
- [52] M. Esposito et al. Pravda: The first solid-state system for proton computed tomography. *Physica Medica*, 55:149 – 154, 2018.
- [53] D. Menichelli, M. Bruzzi, M. Bucciolini, G. Candiano, G. A. P. Cirrone, L. Capineri, C. Civinini, G. Cuttone, D. Lo Presti, L. Marrazzo, S. Pallotta, N. Randazzo, V. Sipala, C. Talamonti, S. Valentini, S. Pieri, V. Reggioli, M. Brianzi, and M. Tesi. Characterization of a silicon strip detector and a yag:ce calorimeter for a proton computed radiography apparatus. *IEEE Transactions on Nuclear Science*, 57(1):8–16, Feb 2010.
- [54] I. Rinaldi, S. Brons, O. Jäkel, B. Voss, and K. Parodi. Experimental investigations on carbon ion scanning radiography using a range telescope. *Physics in Medicine and Biology*, 59(12):3041–3057, may 2014.

-
- [55] T. Price, M. Esposito, G. Poludniowski, J. Taylor, C. Waltham, D.J. Parker, S. Green, S. Manolopoulos, N.M. Allinson, T. Anaxagoras, P. Evans, and J. Nieto-Camero. Expected proton signal sizes in the PRAVDA range telescope for proton computed tomography. *Journal of Instrumentation*, 10(05):P05013–P05013, may 2015.
- [56] Y. Saraya, T. Izumikawa, J. Goto, T. Kawasaki, and T. Kimura. Study of spatial resolution of proton computed tomography using a silicon strip detector. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 735:485 – 489, 2014.
- [57] R. Glattauer. *Track Reconstruction in the Forward Region of the Detector ILD at the Electron-Positron Linear Collider ILC*. PhD thesis, TU Vienna, 2012.
- [58] A. Strandlie and R. Fruhwirth. Track and vertex reconstruction: From classical to adaptive methods. *Rev. Mod. Phys.*, 82:1419–1458, 2010.
- [59] J. Princen, J. Illingworth, and J. Kittler. A formal definition of the hough transform: Properties and relationships. *Journal of Mathematical Imaging and Vision*, 1(2):153–168, Jul 1992.
- [60] R. Mankel. Pattern recognition and event reconstruction in particle physics experiments. *Reports on Progress in Physics*, 67:553, 03 2004.
- [61] S. Neuhaus, S. Skambraks, and C. Kiesling. Track vertex reconstruction with neural networks at the first level trigger of belle ii. *EPJ Web Conf.*, 150:00009, 2017.
- [62] M. Delcourt. The cms tracker upgrade for the high luminosity lhc. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 2018.
- [63] F. Siklér. A combination of analysis techniques for efficient track reconstruction of high multiplicity events in silicon detectors. *Eur. Phys. J.*, A54(6):113, 2018.
- [64] R. Aggleton et al. An FPGA based track finder for the l1 trigger of the CMS experiment at the high luminosity LHC. *Journal of Instrumentation*, 12(12):P12019–P12019, dec 2017.
- [65] G. Traini et al. Design of a new tracking device for on-line beam range monitor in carbon therapy. *Physica Medica*, 34:18 – 27, 2017.
- [66] Performance of the ATLAS Silicon Pattern Recognition Algorithm in Data and Simulation at $\sqrt{s} = 7$ TeV. Technical Report ATLAS-CONF-2010-072, CERN, Geneva, Jul 2010.
- [67] The Optimization of ATLAS Track Reconstruction in Dense Environments. Technical Report ATL-PHYS-PUB-2015-006, CERN, Geneva, Mar 2015.

-
- [68] A. Georges et al. A neural network clustering algorithm for the ATLAS silicon pixel detector. *JINST*, 9:P09009, 2014.
- [69] V. Lacuesta. Track and vertex reconstruction in the atlas experiment. *Journal of Instrumentation*, 8:C02035, 02 2013.
- [70] S. Amrouche et al. Track reconstruction at lhc as a collaborative data challenge use case with ramp. *EPJ Web Conf.*, 150:00015, 2017.
- [71] H. Grote. Pattern recognition in high-energy physics. *Reports on Progress in Physics*, 50(4):473–500, apr 1987.
- [72] N. Chernov, A. Glazov, I. Kisel, E. Konotopskaya, S. Korenchenko, and G. Ososkov. Track and vertex reconstruction in discrete detectors using chebyshev metrics. *Computer Physics Communications*, 74(2):217 – 227, 1993.
- [73] F. James. Fitting Tracks in Wire Chambers Using the Chebyshev Norm Instead of Least Squares. *Nucl. Instrum. Meth.*, 211:145, 1983.
- [74] B. G. M. Vandeginste. Robust regression and outlier detection. *Journal of Chemometrics*, 2(4):299–300, 1988.
- [75] G. Cerati et al. Parallelized kalman-filter-based reconstruction of particle tracks on many-core architectures. *Journal of Physics: Conference Series*, 1085:042016, sep 2018.
- [76] Wikipedia contributors. Kalman filter — Wikipedia, the free encyclopedia, 2019. [Online; accessed 30-January-2019].
- [77] R. Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering (ASME)*, 82D:35–45, 01 1960.
- [78] B. Jones and B. Tompkins. A physicist’s guide to kalman filters. 09 1998.
- [79] R. Frühwirth and A. Strandlie. Track fitting with ambiguities and noise: A study of elastic tracking and nonlinear filters. *Computer Physics Communications*, 120(2):197 – 214, 1999.
- [80] R. Frühwirth. Application of Kalman filtering to track and vertex fitting. *Nucl. Instrum. Meth.*, A262:444–450, 1987.
- [81] S. Chatrchyan et al. Description and performance of track and primary-vertex reconstruction with the CMS tracker. *JINST*, 9(10):P10009, 2014.
- [82] V. Khachatryan et al. CMS Tracking Performance Results from early LHC Operation. *Eur. Phys. J.*, C70:1165–1192, 2010.

-
- [83] T. Cornelissen, M. Elsing, S. Fleischmann, W. Liebig, E. Moyse, and A. Salzburger. Concepts, Design and Implementation of the ATLAS New Tracking (NEWT). Technical Report ATL-SOFT-PUB-2007-007. ATL-COM-SOFT-2007-002, CERN, Geneva, Mar 2007.
- [84] C. Roland. Track reconstruction in heavy ion collisions with the cms silicon tracker. *Nuclear Instruments Methods in Physics Research Section A-accelerators Spectrometers Detectors and Associated Equipment - NUCL INSTRUM METH PHYS RES A*, 566:123–126, 10 2006.
- [85] A. M. Sirunyan et al. Particle-flow reconstruction and global event description with the CMS detector. *Journal of Instrumentation*, 12(10):P10003–P10003, oct 2017.
- [86] R. M. van der Eijk. *Track reconstruction in the LHCb experiment*. PhD thesis, Amsterdam U., 2002.
- [87] F. Pantaleo, A. Schmidt, V. Innocente, B. Hegner, A. Pfeiffer, and A. Meyer. *New Track Seeding Techniques for the CMS Experiment*. PhD thesis, 2017.
- [88] T. Lu, S. Lei, J. Zang, J. Chang, and J. Wu. Study of track reconstruction for dampe. *Chinese Astronomy and Astrophysics*, 41(3):455 – 470, 2017.
- [89] E. Krebs. Application of a kalman filter and a deterministic annealing filter for track reconstruction in the hades experiment. Master’s thesis, Goethe-Universität, March 2012.
- [90] S. Farrell et al. Novel deep learning methods for track reconstruction. In *4th International Workshop Connecting The Dots 2018 (CTD2018) Seattle, Washington, USA, March 20-22, 2018*, 2018.
- [91] G. Apollinari, O. Brüning, T. Nakamoto, and L. Rossi. High Luminosity Large Hadron Collider HL-LHC. *CERN Yellow Report*, (5):1–19, 2015.
- [92] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 521:436–44, 05 2015.
- [93] Wikipedia contributors. Artificial neural network — Wikipedia, the free encyclopedia. https://en.wikipedia.org/w/index.php?title=Artificial_neural_network&oldid=884120674, 2019. [Online; accessed 23 – February – 2019].
- [94] S. Farrell et al. The hep.trkx project: deep neural networks for hl-lhc online and offline tracking. *EPJ Web Conf.*, 150:00003, 2017.
- [95] B. Denby. Neural networks and cellular automata in experimental high energy physics. *Computer Physics Communications*, 49(3):429 – 448, 1988.

-
- [96] C. Peterson. Track finding with neural networks. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 279(3):537 – 545, 1989.
- [97] T. Bilka et al. The track finding algorithm of the belle ii vertex detectors. *EPJ Web Conf.*, 150:00007, 2017.
- [98] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.
- [99] Wikipedia contributors. Convolutional neural network — Wikipedia, the free encyclopedia. https://en.wikipedia.org/w/index.php?title=Convolutional_neural_network&oldid=884489938, 2019. [Online; accessed 23 – February – 2019].
- [100] J. M. Newby et al. Convolutional neural networks automate detection for tracking of submicron-scale particles in 2d and 3d. *Proceedings of the National Academy of Sciences*, 115(36):9026–9031, 2018.
- [101] W. Adam, B. Mangano, T. Speer, and T. Todorov. Track reconstruction in the CMS tracker. 2005.
- [102] T. Speer, R. Frühwirth, P. Vanlaer, and W. Waltenberger. Robust vertex fitters. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 566(1):149 – 152, 2006. TIME 2005.
- [103] F. Ragusa and L. Rolandi. Tracking at LHC. *New Journal of Physics*, 9(9):336–336, sep 2007.
- [104] G. Kitagawa. Non-gaussian seasonal adjustment. *Computers Mathematics with Applications*, 18(6):503 – 514, 1989.
- [105] G. Kitagawa. The two-filter formula for smoothing and an implementation of the gaussian-sum smoother. 2004.
- [106] R. Frühwirth and A. Strandlie. Application of adaptive filters to track finding. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 559(1):162 – 166, 2006. Proceedings of the X International Workshop on Advanced Computing and Analysis Techniques in Physics Research.
- [107] E. Chabanat and N. Estre. Deterministic annealing for vertex finding at CMS. In *Computing in high energy physics and nuclear physics. Proceedings, Conference, CHEP’04, Interlaken, Switzerland, September 27-October 1, 2004*, pages 287–290, 2005.

-
- [108] M. Ohlsson, C. Peterson, and A. L. Yuille. Track finding with deformable templates — the elastic arms approach. *Computer Physics Communications*, 71(1):77 – 98, 1992.
- [109] M. Lindström. Track reconstruction in the atlas detector using elastic arms. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 357(1):129 – 149, 1995.
- [110] G. Aad et al. ATLAS pixel detector electronics and sensors. *Journal of Instrumentation*, 3(07):P07007–P07007, jul 2008.
- [111] M. Gyulassy and M. Harlander. Elastic tracking and neural network algorithms for complex pattern recognition. *Computer Physics Communications*, 66(1):31 – 46, 1991.
- [112] V. Blobel, C. Kleinwort, and F. Meier. Fast alignment of a complex tracking detector using advanced track models. *Comput. Phys. Commun.*, 182:1760–1763, 2011.
- [113] C. Kleinwort. General broken lines as advanced track fitting method. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 673:107 – 110, 2012.
- [114] V. Blobel. A new fast track-fit algorithm based on broken lines. *Nucl. Instrum. Meth.*, A566:14–17, 2006.
- [115] E. Bartz et al. Fpga-based tracklet approach to level-1 track finding at cms for the hl-lhc. *EPJ Web Conf.*, 150:00016, 2017.
- [116] T. Gehrke, R. Gallas, O. Jäkel, and M. Martišíková. Proof of principle of helium-beam radiography using silicon pixel detectors for energy deposition measurement, identification, and tracking of single ions. *Medical Physics*, 45, 12 2017.
- [117] W. Adam, R. Frühwirth, A. Strandlie, and T. Todorov. Reconstruction of electrons with the gaussian-sum filter in the CMS tracker at the LHC. *Journal of Physics G: Nuclear and Particle Physics*, 31(9):N9–N20, jul 2005.