THE UNIVERSITY OF
WARWICK

Library Declaration and Deposit Agreement

1. STUDENT DETAILS

Please complete the following:

Full name: ....ANDREW JAMES BENNIESTON....

University ID number: ....0409139....

2. THESIS DEPOSIT

2.1 I understand that under my registration at the University, I am required to deposit my thesis with the University in BOTH hard copy and in digital format. The digital version should normally be saved as a single pdf file.

2.2 The hard copy will be housed in the University Library. The digital version will be deposited in the University's Institutional Repository (WRAP). Unless otherwise indicated (see 2.3 below) this will be made openly accessible on the Internet and will be supplied to the British Library to be made available online via its Electronic Theses Online Service (EThOS) service.
[At present, theses submitted for a Master's degree by Research (MA, MSc, LLM, MS or MMedSci) are not being deposited in WRAP and not being made available via EthOS. This may change in future.]

2.3 In exceptional circumstances, the Chair of the Board of Graduate Studies may grant permission for an embargo to be placed on public access to the hard copy thesis for a limited period. It is also possible to apply separately for an embargo on the digital version. (Further information is available in the Guide to Examinations for Higher Degrees by Research.)

2.4 *If you are depositing a thesis for a Master's degree by Research, please complete section (a) below.* For all other research degrees, please complete both sections (a) and (b) below:

(a) Hard Copy

I hereby deposit a hard copy of my thesis in the University Library to be made publicly available to readers (please delete as appropriate) ~~EITHER~~ immediately ~~OR after an embargo period of .... months/years as agreed by the Chair of the Board of Graduate Studies~~.

I agree that my thesis may be photocopied. YES / ~~NO~~ (Please delete as appropriate)

(b) Digital Copy

I hereby deposit a digital copy of my thesis to be held in WRAP and made available via EThOS.

Please choose one of the following options:

EITHER My thesis can be made publicly available online. YES / ~~NO~~ (Please delete as appropriate)

OR My thesis can be made publicly available only after.....[date] (Please give date)
YES / NO (Please delete as appropriate)

OR My full thesis cannot be made publicly available online but I am submitting a separately identified additional, abridged version that can be made available online.
YES / NO (Please delete as appropriate)

OR My thesis cannot be made publicly available online. YES / NO (Please delete as appropriate)

3. GRANTING OF NON-EXCLUSIVE RIGHTS

Whether I deposit my Work personally or through an assistant or other agent, I agree to the following:

Rights granted to the University of Warwick and the British Library and the user of the thesis through this agreement are non-exclusive. I retain all rights in the thesis in its present version or future versions. I agree that the institutional repository administrators and the British Library or their agents may, without changing content, digitise and migrate the thesis to any medium or format for the purpose of future preservation and accessibility.

4. DECLARATIONS

(a) I DECLARE THAT:

- I am the author and owner of the copyright in the thesis and/or I have the authority of the authors and owners of the copyright in the thesis to make this agreement. Reproduction of any part of this thesis for teaching or in academic or other forms of publication is subject to the normal limitations on the use of copyrighted materials and to the proper and full acknowledgement of its source.

- The digital version of the thesis I am supplying is the same version as the final, hard-bound copy submitted in completion of my degree, once any minor corrections have been completed.

- I have exercised reasonable care to ensure that the thesis is original, and does not to the best of my knowledge break any UK law or other Intellectual Property Right, or contain any confidential material.

- I understand that, through the medium of the Internet, files will be available to automated agents, and may be searched and copied by, for example, text mining and plagiarism detection software.

(b) IF I HAVE AGREED (in Section 2 above) TO MAKE MY THESIS PUBLICLY AVAILABLE DIGITALLY, I ALSO DECLARE THAT:

- I grant the University of Warwick and the British Library a licence to make available on the Internet the thesis in digitised format through the Institutional Repository and through the British Library via the EThOS service.

- If my thesis does include any substantial subsidiary material owned by third-party copyright holders, I have sought and obtained permission to include it in any version of my thesis available in digital format and that this permission encompasses the rights that I have granted to the University of Warwick and to the British Library.

5. LEGAL INFRINGEMENTS

I understand that neither the University of Warwick nor the British Library have any obligation to take legal action on behalf of myself, or other rights holders, in the event of infringement of intellectual property rights, breach of contract or of any other right, in the thesis.

Please sign this agreement and return it to the Graduate School Office when you submit your thesis.

Student's signature: ..[redacted]... Date: ....2014 – 02 – 17.......

Reconstruction Techniques for Fine-Grained Neutrino Detectors

by

Andrew J. Bennieston

Thesis

Submitted to the University of Warwick

for the degree of

Doctor of Philosophy

Department of Physics

October 2013

THE UNIVERSITY OF
WARWICK

Contents

List of Tables **v**

List of Figures **vi**

Acknowledgements **ix**

Declarations **xi**

Abstract **xiii**

Abbreviations **xiv**

Chapter 1 Introduction **1**

Chapter 2 Neutrino Physics **3**

2.1 History 3

2.2 $\mathcal{CP}$ Violation 5

2.3 Neutrino Oscillations 6

2.4 The PMNS Matrix 9

2.4.1 Solar Neutrinos and θ_{12} 10

2.4.2 Atmospheric Neutrinos and θ_{23} 11

2.4.3 Reactor Neutrinos and θ_{13} 11

2.5 Open Questions in Neutrino Physics 14

Chapter 3 Detector Physics **15**

3.1 Particle Detectors 15

3.2 Physics Motivation 16

3.3 Liquid Argon Time Projection Chambers 17

3.3.1 Production of Ionisation Charge 17

3.3.2 Drifting Charge 18

3.3.3 Wire Readout 18
3.3.4 Thick Gas Electron Multipliers 19
3.4 Point Light Source Reconstruction 20
3.5 The Lamu Simulation 22
3.5.1 Readout and Digitization 24
3.5.2 Characterisation & Testing of the Lamu Simulation 25
3.6 The TrackGen Toy Track Simulation 26
3.7 Neutrino Event Generation with Genie 28
3.8 Neutrino Experiments Using Liquid Argon TPCs 28
3.8.1 The Fermilab Liquid Argon Experiments 29
3.8.2 Laguna LBNO 32
3.8.3 Icarus T600 32

Chapter 4 The Latte Framework 34

4.1 Introduction 34
4.2 Nearest Neighbour Search using KDTree 34
4.3 Charge Weighting 35
4.4 Density-based Clustering 35
4.4.1 DBSCAN 35
4.4.2 OPTICS 36
4.5 Feature Detection 37
4.5.1 Two-dimensional Feature Detection 38
4.5.2 Three-dimensional Feature Detection 38
4.6 Feature Masking 40
4.7 Track Merging 42
4.8 Truth Information 43
4.9 Range Cuts 44
4.10 Latte Control 45
4.10.1 Event Objects 45
4.10.2 Services & Event Wrappers 46
4.10.3 Pipelines 47
4.10.4 Event Loops 47

Chapter 5 A Cellular Automaton for Track Reconstruction 50

5.1 Cellular Automata 50
5.2 Cellular Automata & Track Finding 51
5.3 A 3D Cellular Automaton for Track Finding 52
5.3.1 Terminology 53

5.3.2 Preprocessing . . . 54
5.3.3 Cell Generation . . . 54
5.3.4 Forward Run . . . 55
5.3.5 Reverse Run . . . 57
5.3.6 Postprocessing . . . 57
5.3.7 Optimisation of CA Parameters . . . 58
5.4 Performance of the Cellular Automaton on Toy Monte Carlo Events 58
5.4.1 Raw Cellular Automaton Output . . . 59
5.4.2 Performance of the Cellular Automaton with Merging . . . 60
5.4.3 Performance After a Range Cut . . . 62
5.5 Performance of the Cellular Automaton on Genie Monte Carlo Events 65
5.5.1 Charged Current $\nu_\mu \rightarrow \mu + p$ (CCQE) . . . 65
5.5.2 Charged Current $\nu_\mu \rightarrow \mu + p + \pi^+$ (CC1π) . . . 70
5.6 Conclusions . . . 73

Chapter 6 Track Fitting 77
6.1 Introduction . . . 77
6.2 The Icarus Kalman Filter . . . 77
6.3 The Latte Kalman Filter . . . 80
6.4 Tuning the Kalman Filter . . . 82
6.5 Momentum Measurements of CCQE μ tracks . . . 83
6.5.1 Constrained Momentum Measurements . . . 87
6.6 Conclusions . . . 91

Chapter 7 Particle Identification 92
7.1 Introduction . . . 92
7.2 Muon Track Identification . . . 93
7.2.1 770 MeV $\nu_\mu \rightarrow \mu + p$ (CCQE) Interactions . . . 93
7.2.2 770 MeV $\nu_\mu \rightarrow \mu + p + \pi^+$ (CC1π) Interactions . . . 94
7.2.3 4.5 GeV $\nu_\mu \rightarrow \mu + p$ (CCQE) Interactions . . . 96
7.2.4 4.5 GeV $\nu_\mu \rightarrow \mu + p + \pi^+$ (CC1π) Interactions . . . 96
7.3 Angular Distributions . . . 99
7.3.1 770 MeV $\nu_\mu \rightarrow \mu + p$ (CCQE) Interactions . . . 99
7.3.2 770 MeV $\nu_\mu \rightarrow \mu + p + \pi^+$ (CC1π) Interactions . . . 99
7.3.3 4.5 GeV $\nu_\mu \rightarrow \mu + p$ (CCQE) Interactions . . . 101
7.3.4 4.5 GeV $\nu_\mu \rightarrow \mu + p + \pi^+$ (CC1π) Interactions . . . 101
7.4 Energy Loss (dE/dx) . . . 105
7.4.1 770 MeV $\nu_\mu \rightarrow \mu + p$ (CCQE) Interactions . . . 106

7.4.2 770 MeV $\nu_\mu \to \mu + p + \pi^+$ (CC1π) Interactions 106
7.4.3 4.5 GeV $\nu_\mu \to \mu + p$ (CCQE) Interactions 107
7.4.4 4.5 GeV $\nu_\mu \to \mu + p + \pi^+$ (CC1π) Interactions 108
7.5 Conclusions 108

Chapter 8 Analysis of Neutrino Interactions 111
8.1 Introduction 111
8.2 Charged Current $\nu_\mu \to \mu + p$ at 770 MeV 111
8.2.1 Event Selection Efficiency 111
8.2.2 Muon Energy Reconstruction 113
8.3 Conclusions 114

Chapter 9 Conclusion 117

Appendix A Other Algorithms 121
A.1 KDTree Track Finding 121
A.2 The Hough Transform 123
A.3 Graph Clustering Algorithms 124

Appendix B Continued Use of the Latte Framework 126
B.1 Cellular Automaton 126
B.2 Local Principal Curves 128
B.3 Pandora Particle Flow Analysis 128

Bibliography 129

List of Tables

5.1 CA parameter set for toy track processing 59
5.2 Optimisation of Cellular Automaton reconstruction parameters . . . 67
5.3 CA parameter set for reconstruction of CCQE events 68
5.4 CA parameter set for reconstruction of CC1π events 70

7.1 Composition of tracks after 1000 hit cut on 770 MeV CCQE events . 93
7.2 Composition of tracks after 5000 hit cut on 4.5 GeV CCQE events . 97
7.3 Composition of tracks after 5000 hit cut on 4.5 GeV CC1π events . . 97
7.4 Composition of tracks after cut on $\hat{p}_x$ for 4.5 GeV CCQE events . . . 101
7.5 Definition of symbols used in Bethe-Bloch equation 105
7.6 Composition of tracks after cut on dE/dx for 770 MeV CCQE events 107
7.7 Summary of cuts with efficiencies and purities 110

B.1 CA reconstruction parameters for CCQE events 126
B.2 CCQE reconstruction efficiency 127
B.3 CC1π reconstruction efficiency 127

List of Figures

2.1 Allowed region for θ_{12} and Δm^2_{21} 10

2.2 Allowed region for θ_{23} and Δm^2_{31} 11

2.3 Comparison of measurements of $\sin^2 2\theta_{13}$ 13

3.1 Schematic diagram of the operation of a time-projection chamber . . 17

3.2 Simulation of light intensity recorded by a sparse array of photodetectors 21

3.3 Iterative point light source position reconstruction 22

3.4 Uncertainty on position resolution for several sparse array detector densities 23

3.5 Detector radius required for full containment as a function of energy 26

3.6 Ratio of energy deposited to kinetic energy, by particle species . . . 27

3.7 Charged current neutrino interactions producing a $\mu + p$ final state 29

4.1 Diagram of the effect of charge weighting on hit position 36

4.2 Reachability plot from the OPTICS clustering algorithm 37

4.3 Feature response for a typical neutrino event 39

4.4 Performance of the feature detection service in 3D 41

4.5 Track road merging algorithm operating on clustered hits 43

4.6 Perpendicular distance from a point to a line in 3D 44

4.7 Pipeline of reconstruction and analysis tasks, with an event loop . . 48

5.1 Example of state evolution in *Conway's Game of Life* 51

5.2 Illustration of the cell generation procedure in the CA algorithm . . 55

5.3 Initial and final states of a CA for track finding 56

5.4 Track count as a function of angle for raw CA operating on toy MC events 60

5.5 Sample CA reconstruction of toy event with 42° opening angle . . . 61

5.6 Multiple reconstructed clusters for a single straight line 61
5.7 Track count as a function of angle for CA with merging operating on toy MC events 62
5.8 Track count as a function of angle for CA with merging and range cut operating on toy MC events 63
5.9 Efficiency for finding 2 tracks for CA with merging and range cut operating on toy MC events 64
5.10 Hit efficiency for CA operating on long muon and proton tracks . . . 68
5.11 Track purity for CA operating on long muon and proton tracks . . . 69
5.12 Clusters found by the CA in $\mu + p$ events 71
5.13 Number of reconstructed tracks in CCQE events 72
5.14 Hit efficiency for CCQE events reconstructed with a CA 72
5.15 Track purity for CCQE events reconstructed with a CA 73
5.16 Clusters found by the CA in a $\mu + p + \pi^+$ event 74
5.17 Number of reconstructed tracks in CC1π events 74
5.18 Hit efficiency for CC1π events reconstructed with a CA 75
5.19 Track purity for CC1π events reconstructed with a CA 75

6.1 Reconstructed momenta for various Kalman filter parameters 83
6.2 Momentum residuals for various Kalman filter parameters 84
6.3 True and reconstructed muon momentum distributions at 770 MeV . 85
6.4 True and reconstructed muon momentum distributions at 4.5 GeV . 86
6.5 True and reconstructed muon momentum distributions at 770 MeV (constrained) 88
6.6 Residuals for constrained momentum reconstruction at $E_\nu = 770\,\mathrm{MeV}$ 89
6.7 True and reconstructed muon momentum distributions at 4.5 GeV (constrained) 90

7.1 Track length distribution for μ, p and e^- from 770 MeV neutrinos (CCQE) 94
7.2 Track length distribution for μ, p and π^+ from 770 MeV neutrinos (CC1π) 95
7.3 Track length distributions for μ, p and π^+ from 4.5 GeV neutrinos (CCQE) 96
7.4 Track length distribution for μ, p and π^+ from 4.5 GeV neutrinos (CC1π) 98
7.5 Angular distribution of μ, p at 770 MeV 100
7.6 Angular distribution of μ, p and π^+ at 770 MeV 102

7.7 Angular distribution of μ, p at 4.5 GeV 103
7.8 Angular distribution of μ, p and π^+ at 4.5 GeV 104
7.9 dE/dx distribution for μ, p from 770 MeV ν_μ 106
7.10 dE/dx distribution for μ, p and π^+ from 770 MeV ν_μ 107
7.11 dE/dx distribution for μ, p from 4.5 GeV ν_μ 108
7.12 dE/dx distribution for μ, p and π^+ from 4.5 GeV ν_μ 109

8.1 Relationship between number of hits and energy 113
8.2 Relationship between track length and muon energy 115

A.1 Direction cosines around a corner between two tracks 122
A.2 Hough transform applied to two-track event 124

Acknowledgements

This Ph.D. thesis represents almost four years of research, and many hours of writing, which would not have been possible without the help, guidance, advice and reassurance of many people. It is in this section that I thank most of them, and hope that the ones I omit realise that I do so purely out of forgetfulness.

I must first thank my supervisor, Yorck Ramachers, for providing the opportunity to make real progress in a field of particle physics that was severely under-resourced, as well as for his advice and guidance over the four years that have gone into the work presented here. The discussions, which ranged from short and direct to long and wandering, were nonetheless always interesting and useful, and I would not have made it to this point without his support.

In addition, I would like to thank Gary Barker and Steven Boyd for their unending questions, many of which helped to highlight major issues that must be resolved or explained in order to proceed. Their sharp eyes are responsible for noticing several issues that may otherwise have been left unexplored. I must also thank Ben Morgan for his efforts in supporting the research presented here, most notably by writing and maintaining the Geant4-based *Lamu* simulation central to much of the analysis in this thesis.

Although not directly involved in my research, I must acknowledge the roles of Phill Litchfield and Tom Latham, both of whom provided advice, discussion and entertainment when needed. They are both extremely competent physicists, and my Ph.D. experience was enriched by the many discussions with them.

I would like to thank my friends and colleagues, Leigh Whitehead, Mark

Whitehead, Martin Haigh, Eugenia Puccio, Nicola McConkey, Callum Lister (who painstakingly proof-read much of this thesis!) and Daniel Scully for providing an office environment that never got dull, and for being the source of much insightful conversation about particle physics and the wider world. In addition, I would like to acknowledge the work of Ben Newell, a summer research student who laid the foundations for the cellular automaton that was later adapted and became a central point of my Ph.D., and who engaged thoroughly in the spirit of the particle physics group.

Finally, I must thank my family for their continued support throughout my academic endeavours.

Declarations

This thesis discusses the application of known algorithms to areas of particle physics for which they are not normally applied, or for detector data which does not readily lend itself to existing automated reconstruction techniques. For this reason, the original material of this thesis is distributed throughout, and interspersed with historical or technical overviews of the algorithms involved. These declarations aim to disambiguate the original work from the surrounding discussion.

In chapter 2, the bulk of the discussion summarises the current status of neutrino physics, and is appropriately referenced. The presentation of an averaged result for the value of $\sin^2 2\theta_{13}$, based on the results from the RENO and Daya Bay experiments, is original work by me, although the prediction made by Harrison and Scott is, of course, not.

In chapter 3, the simulation used by Rutter and Richards to obtain an algorithm for iterative reconstruction of point light sources was written by myself. These results were published in [1]. The characterisation plots for the *Lamu* simulation are also my own work, though the simulation itself was written by Ben Morgan. The *TrackGen* toy simulation was written by me, with some of the event generator modules originating from Ben Morgan. The sources of information on detector physics are referenced throughout.

In chapter 4, the bulk of the work described originated from me, with the exception of the two-dimensional feature detection, which is attributed to Ben Morgan. The three-dimensional feature detection characterisation study is my own work. Some data structures such as the KDTree are used directly from the SciPy library. The remaining work is my own.

The cellular automaton presented in chapter 5 is my own work, based on ideas from Ben Newell, who was in turn influenced by the reconstruction algorithms used for the SciBoone detector. Ben Newell created a two-dimensional prototype, which underwent a year-long metamorphosis into the current three-dimensional version. None of the original code remains. A paper describing the development and characterisation of the cellular automaton has been accepted for publication in Eur. Phys. J. C [2], based on the work I carried out and presented here.

The Kalman filter of chapter 6 is based heavily on that presented by the Icarus collaboration, and the Python implementation was written by Yorck Ramachers, based on the SciPy implementation of a generic Kalman filter. The analysis of its performance on muon tracks is entirely my own.

The particle identification work of chapter 7 is entirely my own, though I acknowledge the use of the *Lamu* simulation written by Ben Morgan (as used throughout this thesis). The analysis presented in chapter 8 is also my own work.

The investigations described in appendix A were carried out by myself, while the work described in appendix B was performed by others in the group at Warwick, making use of the algorithms and framework I developed.

Abstract

The use of Liquid Argon Time-Projection Chambers in neutrino physics is looking increasingly certain as the field moves to larger, finer-grained detectors capable of delivering the physics reach required for the next generation of experiments studying neutrino oscillations and $\mathcal{CP}$ violation in the lepton sector.

This thesis explores reconstruction procedures for use in a Liquid Argon Time-Projection Chamber (LAr TPC). Fully automated reconstruction of neutrino events in these environments has not been successfully demonstrated previously, although several collaborations across the world are working towards this goal. A number of algorithms and techniques are discussed, and their applicability to the field of reconstruction in fine-grained detector environments is assessed.

The techniques presented here are fully automated and characterised to the maximum extent possible, and may be combined to produce a software reconstruction chain that is free from human intervention. In addition to these algorithms, a framework for running chained reconstruction tasks is presented and demonstrated to work in conjunction with the algorithms developed.

Muon identification is also presented, using cuts justified from the truth information available from simulations. The algorithms and cuts together are used to analyse simulated neutrino events throughout the thesis, focusing on charged current muon neutrino interactions at energies of 0.77 GeV and 4.5 GeV, and considering interactions producing $\mu + p$ (referred to as CCQE) or $\mu + p + \pi^+$ (referred to as CC1π) final states.

Abbreviations

CA	Cellular Automaton
CARLA	Cellular Automaton for Reconstruction in Liquid Argon
CCD	Charge Coupled Device
CCQE	Charged Current Quasi-Elastic
CC1π	Charged Current interactions producing a $\mu + p + \pi$ final state
DBSCAN	Density-Based Spatial Clustering of Applications with Noise
FSM	Finite State Machine
KDTree	k-dimensional tree
LAr	Liquid Argon
LAr TPC	Liquid Argon Time-Projection Chamber
MC	Monte Carlo
MWPC	Multi-Wire Proportional Chamber
OPTICS	Ordering Points To Identify the Clustering Structure
PCB	Printed Circuit Board
PID	Particle Identification
TGEM	Thick Gas Electron Multiplier
TPC	Time-Projection Chamber

1 Introduction

This thesis explores reconstruction procedures for use in a LAr TPC. Fully automated reconstruction of neutrino events in these environments has not been successfully demonstrated previously, although several collaborations across the world are working towards this goal. This thesis discusses a number of algorithms and techniques, and assesses their applicability to the field of reconstruction in fine-grained detector environments.

These environments present a number of challenges, including multiple scattering occurring throughout the active detector volume. Furthermore, in neutrino experiments, the primary vertices of events are not well-localised to a single interaction point. This means that algorithms must exhibit some degree of translational invariance, i.e. they must be insensitive to the precise starting point of the event.

Many of the techniques presented existed as algorithms in other fields of science, mostly the machine vision and learning disciplines of computer science, while others are more traditional particle physics techniques that were adapted for the unique environment provided by LAr TPCs. Each technique is described in terms of its theoretical or computational basis before the implementation details are explained and the results of applying the technique to LAr TPC data are presented.

The ultimate goal is to present options for a fully automated reconstruction chain, going from detector readout at one end to physics information at the other. This chain should require no manual intervention[1] and should be free of arbitrarily defined cuts or thresholds. In every case, where a parameter, cut value or threshold is required, it should be justified in terms of the data and the performance of the algorithm.

In addition to the algorithms and techniques themselves, this thesis also

[1] Many existing “bubble chamber”-like experiments use some degree of human interaction to scan events and pick out starting points for less sophisticated computer algorithms.

presents a framework known as *Latte*, in which fully automated reconstruction and analysis tasks may be performed, making use of a pipeline architecture to simplify the flow of data from one algorithm to the next, and automating tasks such as reading event data from disk, looping over events, and writing results back to disk. The algorithms presented here all work in conjunction with this framework, and each algorithm implemented contains a control module for use with Latte.

The emphasis throughout this thesis is on small, self-contained algorithms that are verified against truth data to the maximum extent possible. These can be used as building blocks to establish a reconstruction chain that has known properties.

The structure of this thesis is to present the current status of neutrino physics in chapter 2 before explaining the relevance of LAr TPCs to the global experimental neutrino programme in chapter 3, as well as looking at the details of LAr TPCs from a detector physics perspective. In chapter 4, the Latte framework is described in detail, along with a number of the smaller algorithms and services it provides. While these algorithms are grouped together here, they are in no way less important than those that are the subject of an entire chapter.

Chapter 5 describes a clustering algorithm which uses a cellular automaton to find track-like objects in heterogeneous event structures (i.e. those containing tracks and additional, unrelated hits from detector noise, co-developing showers, etc.), while chapter 6 describes a Kalman filter designed to obtain momentum estimates of muon tracks. Chapter 7 describes particle identification techniques of relevance for extracting muon tracks from a reconstructed event, and chapter 8 attempts to put these building blocks together into an end-to-end analysis.

Finally, conclusions are drawn in chapter 9, summarising the results from the other chapters.

2 Neutrino Physics

2.1 History

The neutrino was first proposed by Wolfgang Pauli in December 1930 as a radical solution to the problem of energy conservation in beta decay. It had been observed that the electron from beta decay had a continuous energy distribution, which could only be consistent with the law of conservation of energy if a third particle was also emitted. This particle, which Pauli called a *neutron*, was incorporated into Enrico Fermi's 1934 theory of beta decay as the *neutrino* [3, 4]. The neutrino was postulated to be (almost) massless and electrically neutral. Its role in beta decay meant it must carry spin $\frac{1}{2}$. The neutrino remained a theoretical ghost until it was conclusively discovered in 1956 by Frederick Reines and Clyde Cowan [5], in an experiment which agreed with the predicted interaction cross-section of $6.3 \times 10^{-44}\,\mathrm{cm}^2$.

In 1964, John Bahcall published a solar model which predicted the neutrino flux from the Sun [6]. This model was put to the test by Raymond Davis, Jr. in what is now known as the Homestake experiment [7]; a carefully designed radiochemical experiment which measured the solar neutrino flux by counting the decays of ^{37}Ar atoms which were produced in charged current neutrino interactions on ^{37}Cl nuclei. After a period of 20 years of data-taking, theoretical and experimental refinements, the Homestake experiment concluded that there was a discrepancy between experiment and theory; the rate measured experimentally was approximately one third the predicted rate. This became known as the *solar neutrino problem.*

Meanwhile, it was discovered that the neutrino that appears in pion decay ($\pi^\pm \to \mu^\pm + \nu$) differs somehow from that produced in β decay [8]; the neutrinos from pion decay always produced muons in a charged current interaction, while those from β decay produced electrons. This distinction between neutrino *flavours* led Pontecorvo to predict mixing (*oscillation*) between neutrino flavours in 1958 [9], and in particular $\nu_e \leftrightarrow \nu_\mu$ oscillations in 1968 [10].

The Kamiokande experiment further reinforced the solar neutrino problem, with an experimental result showing that only about half of the expected neutrino flux was observed with a detection mechanism that involved imaging the Čerenkov light produced when electrons from neutrino interactions pass through water [11]. The Kamiokande experiment was originally designed to search for proton decay, and in studying their backgrounds they discovered another problem with neutrinos; the *atmospheric neutrino problem*, in which the measured flux of neutrinos from cosmic ray interactions in the atmosphere ($p \to \pi \to \mu + \nu \to \nu + e$) was only about 60% of that expected.

With an observed deficit in both electron (ν_e) and muon (ν_μ) neutrinos, the oscillation predicted by Pontecorvo provided a likely explanation, but it was only in 2002 that this was confirmed. The Sudbury Neutrino Observatory (SNO) measured the flux of ^{8}B solar neutrinos [12] with a water Čerenkov detector filled with heavy water (D_2O). SNO was sensitive to charged current (CC), elastic scattering (ES) and neutral current (NC) neutrino interactions of the form:

$$\begin{aligned} CC: &\quad \nu_e + d \to e^- + p + p \\ ES: &\quad \nu + e^- \to \nu + e^- \\ NC: &\quad \nu + d \to \nu + p + n \end{aligned}$$

The Kamiokande experiment was sensitive to the CC and ES interactions, but not to the NC interaction. It was this additional sensitivity that allowed SNO to measure not only the ν_e flux, but also the *total* neutrino flux, independent of flavour. While the CC flux was found to be low, by an amount consistent with Kamiokande and Homestake, the total rate of interactions from all flavours of neutrino, measured through the NC channel, implied a neutrino flux which was precisely that predicted by the solar model, and John Bahcall's work was finally validated, over 30 years after the solar neutrino problem was discovered.

The SNO results provided a clear indication that neutrino flavour oscillations were taking place: the pure ν_e from the Sun were undergoing flavour change, appearing at Earth as ν_μ and ν_τ. The theory of neutrino oscillations—expanded by Maki, Nakagawa and Sakata in 1962 [13]—requires that the neutrinos have mass, and introduces a 3 × 3 unitary mixing matrix (the PMNS matrix) analogous to the

CKM matrix[1] for quark flavour mixing (see chapter 2.3).

It is now known that the dominant mechanism for the reduction of ν_e in the solar neutrino flux is a resonant interaction known as the MSW[2] effect, which occurs in the electron-rich solar environment. Once the neutrinos leave the Sun, their propagation through the vacuum to Earth makes no substantial change. The MSW effect is due to charged current elastic forward scattering of neutrinos in environments where there is a high density of the corresponding flavoured charged lepton [14].

2.2 $\mathcal{CP}$ Violation

It is thought that in the very early universe, immediately after the big bang, matter and antimatter were created in equal amounts. The visible universe today is dominated by matter, a situation that requires some asymmetry in the way matter and antimatter interact [15].

This asymmetry is known as $\mathcal{CP}$ violation, where $\mathcal{CP}$ is the combined operation of the charge conjugation operator $\mathcal{C}$ (which swaps particles for antiparticles, and vice versa), and the parity operator $\mathcal{P}$ (which flips the signs of all spatial coordinates). $\mathcal{CP}$ violation is one of the three Sakharov conditions [16] required to generate the measured baryon asymmetry of the universe:

1. Existence of a baryon number violating process.
2. Existence of $\mathcal{C}$ and $\mathcal{CP}$ violation.
3. Interactions outside of thermal equilibrium.

The first condition is required in order for baryons to be produced at all. Under the assumption that such a mechanism exists, the second condition introduces an asymmetry which favours the production of baryons over that of antibaryons. This must occur outside of thermal equilibrium, such that the creation processes and annihilation processes have different rates.

$\mathcal{CP}$ violation has been observed in the quark sector [17], but the level to which it has been observed is insufficient to produce the asymmetry we see in the universe today. Neutrino oscillations open up the possibility for $\mathcal{CP}$ violation to occur in the lepton sector, producing a lepton asymmetry which may then be converted to a baryon asymmetry [18].

[1] After Cabibbo, Kobayashi and Maskawa.

[2] After Mikheyev, Smirnov and Wolfenstein.

2.3 Neutrino Oscillations

The mixing of three massive neutrinos can be parametrised with a 3 × 3 unitary[3] matrix U, known as the PMNS matrix.[4] The matrix U relates the three neutrino flavour states ν_α to the three mass states ν_i. The flavour state $|\nu_\alpha\rangle$ can be expressed as a superposition of mass states $|\nu_i\rangle$ through components of U [19]:

$$|\nu_\alpha\rangle = \sum_i U^*_{\alpha i}|\nu_i\rangle \tag{2.1}$$

Given this relationship between a flavour state $|\nu_\alpha\rangle$ and the mass states $|\nu_i\rangle$, it is possible to determine the probability that a neutrino produced in state $|\nu_\alpha\rangle$ is observed at some later time to be in state $|\nu_\beta\rangle$. A full treatment is too long to describe here[5] so, as is traditional, several simplifications will be made. The first is to assume a plane-wave form for the neutrino states; while this is unphysical, it allows for the trivial development of the expression for the oscillation probability. It is then necessary to manually impose the requirement that the neutrino is localised in space and propagates at a fixed speed, so that the system evolves purely as a function of the distance travelled by the neutrino, x. Then:

$$|\nu(x,t)\rangle = e^{-ip\cdot x}|\nu(0)\rangle \tag{2.2}$$

The expression $p \cdot x = Et - \mathbf{p}\cdot\mathbf{x}$ is dependent on the mass of the neutrino involved, so it must be expanded in the mass basis as follows:

$$|\nu(x)\rangle = \sum_i A_i e^{-ip_i\cdot x}|\nu_i\rangle \tag{2.3}$$

The amplitude A_i is the amplitude of the mass state $|\nu_i\rangle$ at $x = 0$. Neutrinos are always produced as flavour eigenstates, so the amplitudes of the mass states are given by transforming using the PMNS matrix U:

$$A_i = \langle\nu_i|\nu_\alpha\rangle = U^*_{\alpha i} \tag{2.4}$$

After the neutrino has propagated some distance x (during which time we deal with the mass states), it is detected by observing the flavour state. This requires

[3] $U^\dagger U = \mathbb{1}$

[4] After Pontecorvo, Maki, Nakagawa and Sakata.

[5] See, for instance, [19] for a complete treatment using wave-packets.

a second transformation:

$$\langle \nu_\beta | = \sum_j U_{\beta j} \langle \nu_j | \tag{2.5}$$

The oscillation amplitude is obtained by putting together the two transformations, along with the propagation term:[6]

$$\begin{aligned} \langle \nu_\beta(x) | \nu_\alpha(0) \rangle &= \sum_j \sum_i U_{\beta j} \langle \nu_j | U^*_{\alpha i} e^{-ip_i \cdot x} | \nu_i \rangle \\ &= \sum_i U_{\beta i} U^*_{\alpha i} e^{-ip_i \cdot x} \end{aligned} \tag{2.6}$$

The phase $\phi_i = p_i \cdot x$ is problematic, since a plane wave has well-defined momentum and energy, and therefore mass through $E^2 = p^2 + m^2$, but we measure flavour states, not mass states[7] so we must introduce some uncertainty into the energy or the momentum. In principle, we must integrate over the range of allowed momenta, but in the spirit of simplification, a widely adopted strategy is to force all neutrino mass states to have the same momentum $\mathbf{p}$. This introduces small variations in energy (through the mass differences) and saves the oscillation probability. The phase is now given by:

$$\phi_i = E_i t - \mathbf{p} \cdot \mathbf{x} \tag{2.7}$$

Under the assumption that the neutrino is highly relativistic, i.e. the mass states $m_i \ll E$, it is possible to expand:

$$\begin{aligned} E_i &= \sqrt{|\mathbf{p}|^2 + m_i^2} \\ &\approx |\mathbf{p}| \left(1 + \frac{m_i^2}{2|\mathbf{p}|^2} \right) \end{aligned} \tag{2.8}$$

In a highly relativistic scenario, the distance L from source to detector is approximately the travel time t, and the energy E is approximately the magnitude of the momentum, $|\mathbf{p}|$. Making these substitutions yields:

$$\begin{aligned} \phi_i &= |\mathbf{p}| \left(1 + \frac{m_i^2}{2|\mathbf{p}|^2} \right) L - |\mathbf{p}| L \\ &= \frac{m_i^2 L}{2E} \end{aligned} \tag{2.9}$$

[6] Note that $\langle \nu_j | \nu_i \rangle = \delta_{ij}$, i.e. the mass states are orthogonal and are eigenstates of the Hamiltonian.

[7] Actually, knowledge of the mass states involved destroys the oscillation probability, see [19] for details.

The oscillation probability amplitude can then be written as:

$$\langle \nu_\beta(L)|\nu_\alpha(0)\rangle = \sum_i U_{\beta i}U^*_{\alpha i} \exp\left(-i\frac{m_i^2 L}{2E}\right) \tag{2.10}$$

with a corresponding probability:[8]

$$\begin{aligned} P(\nu_\alpha \to \nu_\beta) &= |\langle \nu_\beta|\nu_\alpha\rangle|^2 \qquad\qquad (2.11)\\ &= \sum_{ij} U_{\beta i}U^*_{\alpha i}U^*_{\beta j}U_{\alpha j} \exp\left(-i\frac{(m_i^2 - m_j^2)L}{2E}\right) \end{aligned}$$

Writing $\Delta m^2_{ij} = m_i^2 - m_j^2$ gives:[9]

$$\begin{aligned} P(\nu_\alpha \to \nu_\beta) =& \sum_i U_{\beta i}U^*_{\alpha i} \sum_j U^*_{\beta j}U_{\alpha j} \\ &+ \sum_{ij} U_{\beta i}U^*_{\alpha i}U^*_{\beta j}U_{\alpha j} \left[\exp\left(-i\frac{\Delta m^2_{ij}L}{2E}\right) - 1\right] \end{aligned} \tag{2.12}$$

The first term in equation (2.11) reduces to $\delta_{\alpha\beta}$ since U is unitary. The second term can be rewritten with the understanding that terms with $j > i$ are the complex conjugate of terms with $j < i$, and that the phase difference is 0 when $j = i$. We now obtain:

$$P(\nu_\alpha \to \nu_\beta) = \delta_{\alpha\beta} + 2\sum_{i>j} U_{\beta i}U^*_{\alpha i}U^*_{\beta i}U_{\alpha i}\left(e^{-i\Delta m^2_{ij}L/2E} - 1\right) \tag{2.13}$$

The term inside parentheses can be split into real and imaginary parts, giving (after a little effort):

$$\begin{aligned} P(\nu_\alpha \to \nu_\beta) = \delta_{\alpha\beta}& \\ &- 4\sum_{i>j} \mathfrak{Re}\left(U_{\beta i}U^*_{\alpha i}U^*_{\beta j}U_{\alpha j}\right)\sin^2\left(\frac{\Delta m^2_{ij}L}{4E}\right) \\ &+ 2\sum_{i>j} \mathfrak{Im}\left(U_{\beta i}U^*_{\alpha i}U^*_{\beta j}U_{\alpha j}\right)\sin\left(\frac{\Delta m^2_{ij}L}{2E}\right) \end{aligned} \tag{2.14}$$

The first term of equation (2.14) states that in the absence of oscillations, we would detect the same flavour that we started with. The second term (which takes

[8] The "distance" labels (L) and (0) are omitted from this point forward.

[9] The rearrangement here is for convenience; the expression remains the same, but now we have two clearly identifiable terms.

the real parts of the PMNS matrix elements, and involves a $\sin^2$ term) is $\mathcal{CP}$ even, behaving identically for particles and antiparticles (for which the oscillation probability is found by taking the complex conjugate of each U term). This $\mathcal{CP}$ even term is responsible for neutrino oscillations. The third term is $\mathcal{CP}$ odd, and disappears completely if U is real.[10] The third term also vanishes if we are considering *survival*, i.e. $P(\nu_\alpha \to \nu_\alpha)$, for which the *survival probability* is:

$$P(\nu_\alpha \to \nu_\alpha) = 1 - 4\sum_{i>j} |U_{\alpha i}|^2 |U_{\alpha j}|^2 \sin^2\left(\frac{\Delta m_{ij}^2 L}{4E}\right) \tag{2.15}$$

2.4 The PMNS Matrix

The PMNS matrix U can be parametrised by three Euler angles and (in general) six phases. For the case of three massive Dirac neutrinos ($\nu_i \neq \bar{\nu}_i$), five of the phases can be absorbed by the particle fields. Three absorbed phases correspond to unobservable rotations of the neutrino flavour eigenstates, two correspond to relative phases of the mass eigenstates. This leaves just one $\mathcal{CP}$ violating Dirac phase. For Majorana neutrinos ($\nu_i = \bar{\nu}_i$) there are two additional Majorana phases, which cannot be absorbed.[11] In the usual formulation [21], the matrix $U \equiv VP$ (where P encapsulates the Majorana phases) is expressed as follows, where $c_{ij} \equiv \cos\theta_{ij}$ and $s_{ij} \equiv \sin\theta_{ij}$ represent the cosines and sines of the mixing angles.

$$V \equiv \begin{pmatrix} 1 & 0 & 0 \\ 0 & c_{23} & s_{23} \\ 0 & -s_{23} & c_{23} \end{pmatrix} \begin{pmatrix} c_{13} & 0 & s_{13}e^{-i\delta} \\ 0 & 1 & 0 \\ -s_{13}e^{i\delta} & 0 & c_{13} \end{pmatrix} \begin{pmatrix} c_{12} & s_{12} & 0 \\ -s_{12} & c_{12} & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{2.16}$$

$$P \equiv \mathrm{diag}\left(1,\ \exp\left[\frac{i\alpha_{21}}{2}\right],\ \exp\left[\frac{i\alpha_{31}}{2}\right]\right) \tag{2.17}$$

In the expression for V above, each sub-matrix represents a rotation corresponding (roughly) to mixing between two neutrinos. This was used historically as a simplification for oscillation probability calculations, and each sub-matrix was assigned a name based on the neutrino source for an experiment which most readily provided sensitivity: “solar” for θ_{12}, “atmospheric” for θ_{23}, and “reactor” for θ_{13}.

[10] In the special case of two neutrino flavours, U must be real. This restriction is lifted for three flavours.

[11] Majorana neutrinos, neutrino mass mechanisms, and lepton-number violating processes such as neutrinoless double-β decay will not be described further in this thesis. For more details, consult a good book on neutrino physics [14, 20].

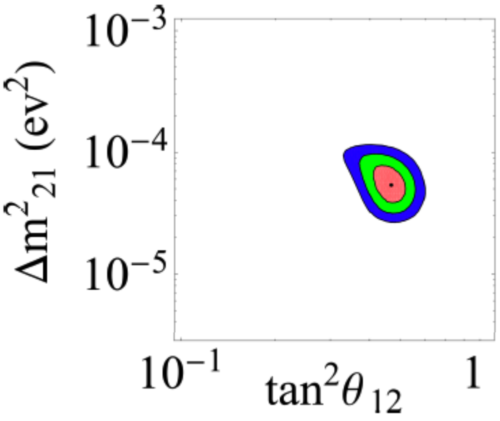


Figure 2.1: The current knowledge of the values of θ_{12} and Δm^2_{21} represented as a plot of Δm^2_{21} against $\tan^2\theta_{12}$. The contours give the 68.27%, 95.45% and 99.74% confidence level allowed regions for a global solar neutrino analysis. Figure adapted from [26].

The latest oscillation results are analysed in the context of a full three-neutrino mixing scenario, where it is more convenient to represent the matrix V in a combined form:

$$V \equiv \begin{pmatrix} c_{12}c_{13} & s_{12}c_{13} & s_{13}e^{-i\delta} \\ -s_{12}c_{23} - c_{12}s_{23}s_{13}e^{i\delta} & c_{13}c_{23} - s_{12}s_{23}s_{13}e^{i\delta} & s_{23}c_{13} \\ s_{12}s_{23} - c_{12}c_{23}s_{13}e^{i\delta} & -c_{12}s_{23} - s_{12}c_{23}s_{13}e^{i\delta} & c_{23}c_{13} \end{pmatrix} \tag{2.18}$$

2.4.1 Solar Neutrinos and θ_{12}

The best measurements of the angle θ_{12} and the mass splitting Δm^2_{21} come from solar neutrino experiments including the Homestake (Cl), SAGE [22], GALLEX [23] and GNO [24] (all Ga) radiochemical experiments, as well as from the Super-Kamiokande and SNO (H_2O and D_2O, respectively) water Čerenkov detectors.

From a global three-flavour oscillation analysis, taking into account data from all contributing experiments, the values for the "solar" parameters θ_{12} and Δm^2_{21} are shown in figure 2.1, with contours highlighting the allowed region of the parameter space. The best-fit values are [25]:

$$\sin^2\theta_{12} = 0.318^{+0.019}_{-0.016}, \qquad \Delta m^2_{21} = 7.59^{+0.23}_{-0.18} \times 10^{-5}\,\mathrm{eV}^2 \tag{2.19}$$

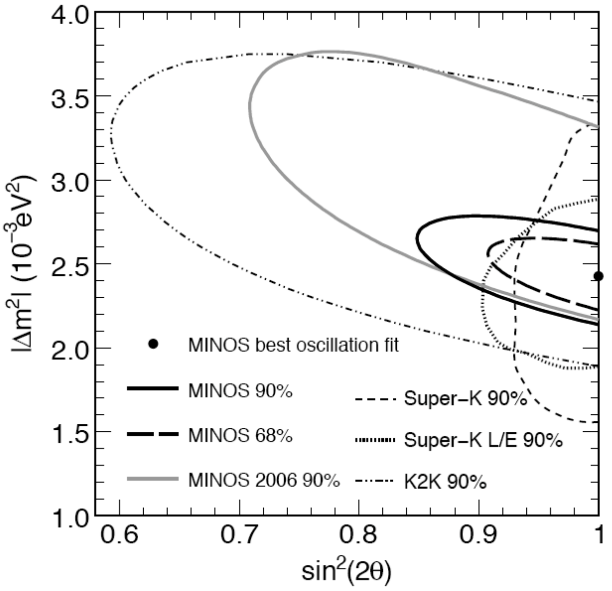


Figure 2.2: The current knowledge of the values of θ_{23} and Δm^2_{31} as measured by the MINOS, Super-Kamiokande and K2K experiments, with allowed regions at 68% and 90% confidence levels. Figure taken from [21].

2.4.2 Atmospheric Neutrinos and θ_{23}

Measurements of θ_{23} and the mass splitting Δm^2_{31} were obtained from experiments measuring atmospheric neutrino oscillations from Super-Kamiokande [27], combined with results from the K2K [28] and MINOS [29] long-baseline accelerator neutrino experiments. Figure 2.2 represents the allowed regions based on each experiment's results, along with the MINOS best fit value. The best-fit values are [25]:

$$\sin^2\theta_{23} = 0.50^{+0.07}_{-0.06}, \qquad \left|\Delta m^2_{32}\right| \approx \left|\Delta m^2_{31}\right| = 2.40^{+0.12}_{-0.11} \times 10^{-3}\,\mathrm{eV}^2 \tag{2.20}$$

2.4.3 Reactor Neutrinos and θ_{13}

From the PMNS matrix, equation (2.18), it is clear that the $\mathcal{CP}$ violating phase δ is always associated with a $\sin\theta_{13}$ term. This means that $\mathcal{CP}$ violation in the lepton sector requires a non-zero θ_{13}. The T2K [30] experiment published indications that θ_{13} was non-zero in 2011, but the measurement was claimed in 2012 by the Daya Bay reactor neutrino experiment. The Daya Bay collaboration published a result for non-zero θ_{13} with 5.2σ significance [31], quoting a measured value of:

$$\sin^2 2\theta_{13} = 0.092 \pm 0.016(\mathrm{stat}) \pm 0.005(\mathrm{sys}) \tag{2.21}$$

Within a month of the Daya Bay result, the RENO reactor neutrino experiment also published a measurement of $\sin^2 2\theta_{13}$ [32]:

$$\sin^2 2\theta_{13} = 0.103 \pm 0.013(\text{stat}) \pm 0.011(\text{sys}) \tag{2.22}$$

Since both the Daya Bay and RENO results are from rate-only analyses, i.e. looking for a deficit of $\bar{\nu}_e$ in the far detector, compared to the flux predicted based on measurements in the near detectors, and in the absence of oscillations, it is reasonably straightforward to combine the two results. The combination process involves averaging the values, weighted by their combined errors (weight $w = 1/\sigma^2$). In order for this to work, the errors should ideally be Gaussian distributed, and uncorrelated. While this is true for the statistical errors, it cannot be trivially assumed for the systematics, which may exhibit some correlation between the two experiments. In the case of the Daya Bay and RENO experiments, however, many of the sources of systematic error are cancelled in the final result due to their use of both near and far detectors. The addition of near detectors allows for flux measurement as well as the cancellation of systematic errors common to both detectors, by taking the ratio of the neutrino flux seen by the near and far detectors. On this basis, it is safe to combine the results of both experiments on the understanding that the remaining systematic errors are uncorrelated between the experiments themselves.

Ideally, such a combination would take the predicted and measured fluxes in each reactor–detector pair (across both experiments), and use a χ^2 minimisation, weighted by the statistical and systematic uncertainties presented for each detector. In the absence of this detailed information, the best one can do is to assume Gaussian, uncorrelated errors, and proceed accordingly. This assumption is somewhat backed up by the parabolic shape of the χ^2 versus $\sin^2 2\theta_{13}$ curves presented by the experiments [31, 32].

It turns out that the combined statistical and systematic errors for the two experiments are the same:

$$\begin{aligned} \sigma_{\text{Daya Bay}} &= \sqrt{0.016^2 + 0.005^2} = 0.017 \\ \sigma_{\text{RENO}} &= \sqrt{0.013^2 + 0.011^2} = 0.017 \end{aligned} \tag{2.23}$$

This simplifies the weighted average to a simple mean:

$$\left\langle \sin^2 2\theta_{13} \right\rangle = \frac{0.092 + 0.103}{2} = 0.098 \tag{2.24}$$

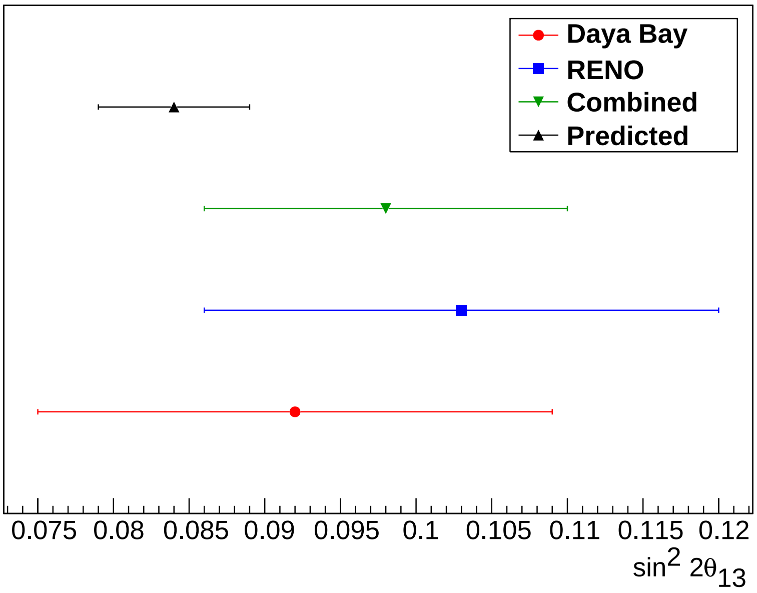


Figure 2.3: Comparison of measurements of $\sin^2 2\theta_{13}$ from the Daya Bay experiment (red), the RENO experiment (blue), a combination of the two results (green) and prediction by Harrison and Scott (black).

The error on this combined quantity is $\sigma = \sqrt{1/\sum w}$, which reduces to:

$$\sigma_{\text{Combined}} = \frac{\sigma}{\sqrt{2}} = 0.012 \tag{2.25}$$

The combined result is therefore:

$$\left\langle \sin^2 2\theta_{13} \right\rangle = 0.098 \pm 0.012 \tag{2.26}$$

A more thorough combination, bringing in the T2K and MINOS results on θ_{13}, is beyond the scope of this thesis, though such an analysis would allow the production of a confidence region plot showing the parameter space for θ_{13} and the $\mathcal{CP}$ violating phase δ (the reactor experiments are measuring $\bar{\nu}_e$ disappearance, which is not sensitive to δ; a ν_e appearance measurement is required).

It is worth noting that in 2004, a prediction compatible with this value of $\sin^2 2\theta_{13}$ was made by Harrison and Scott, by relating $\sin\theta_{13}$ to the two mass-squared differences, Δm^2_{21} and Δm^2_{31} [33]. Their prediction, motivated by the results available at the time, states:

$$\sin\theta_{13} = \sqrt{\frac{2\Delta m^2_{21}}{3\Delta m^2_{31}}} \tag{2.27}$$

Converting this to a statement about $\sin^2 2\theta_{13}$, and neglecting terms with Δm^4 (or, equivalently, $\sin^4\theta$) yields:

$$\sin^2 2\theta_{13} = \frac{8\Delta m^2_{21}}{3\Delta m^2_{31}} \tag{2.28}$$

Substituting in the global best-fit values for the two mass splittings, and propagating through the errors, one obtains:

$$\sin^2 2\theta_{13,\,\mathrm{predicted}} = 0.084 \pm 0.005 \tag{2.29}$$

The two experimental results, along with the combined result and the predicted value, are shown in figure 2.3.

2.5 Open Questions in Neutrino Physics

With a non-zero θ_{13} firmly established, the experimental focus will now shift to the determination of the $\mathcal{CP}$ violating phase δ. $\mathcal{CP}$ violation in the lepton sector manifests as a difference between the vacuum oscillation probabilities for neutrinos and for antineutrinos, and could form part of an attractive mechanism for generating the observed baryon asymmetry of the universe through processes such as leptogenesis [18].

Since the sign of the mass splitting Δm^2_{31} is not known, there remains a question of *hierarchy*; that is, whether there are two light neutrino mass states and one heavier (the so-called *normal* hierarchy), or two heavy and one light (the *inverted* hierarchy).

The possible Majorana nature of neutrinos is currently being probed by neutrinoless double-β decay experiments, including SuperNEMO [34] and part of the physics programme for the SNO+ upgrade [35] to the SNO project.

3

Detector Physics

3.1 Particle Detectors

Particle detectors for high-energy physics experiments typically include a number of components, each designed to perform one specific task. A common design, based on a central interaction region (i.e. in collider experiments) uses a fine-grained tracker close to the interaction region, with calorimeters and coarse trackers further out. Muon chambers around the outside provide a signal if a particle penetrates the calorimeters and leaves the detector entirely. In this scenario, detector elements are designed to perform either tracking or calorimetry. A tracker provides details of the passage of ionising particles through a detector, usually in the form of spatial points or *hits*. Hits are combined using a track reconstruction algorithm, yielding parameters which estimate the path a particle took (usually a straight line, or a curve or helix in a magnetic field). Tracks may be back-projected to estimate the interaction vertex, and curvature in a magnetic field provides a measurement of particle momentum [36].

The history of tracking detectors began with the bubble chamber, in which each event consists of images of the bubbles produced when ionising particles pass through a detector, recorded onto photographic plates. The signal readout was performed by manually scanning each photographic plate. Wire chambers and drift chambers also provide this tracking capability, and they are preferred since the readout is electronic, and can be automated. In a drift chamber, ionising particles release electrons from the gas that fills the chamber. These electrons drift in an electric field (provided by field wires) towards a signal wire where the charge is read out. The position resolution of such a detector depends on the wire pitch.

The current generation of particle physics experiments also makes use of Silicon trackers, which use the properties of Silicon semiconductors to trap ionisation charge into pixels which can be read out electronically. These can provide extremely high resolution (on the order of μm) but are also rather expensive.

Calorimeters deal with the determination of the kinetic energy of a particle (or of an electromagnetic or hadronic shower produced by an initial particle). The energy losses are measured by a detector medium which produces light or charge signals proportional to the energy deposited. With a sufficiently large calorimeter, particles of interest will come to a stop completely within the calorimeter, and the total kinetic energy can be measured. Liquid Argon is currently used in the calorimeters of a number of experiments, including the electromagnetic calorimeter of the ATLAS [37] experiment. Liquid Argon is usually used to build sampling calorimeters, in which layers containing liquid Argon are alternated with layers of a high-density absorber such as Lead. This technology provides calorimetry in conjunction with high stopping power. The scintillation light from the liquid Argon is generally wavelength shifted in order that the peak wavelength coincides with the peak photon detection efficiency of a photosensitive device. Liquid Argon calorimetry is therefore a demonstrated and proven technology [38].

3.2 Physics Motivation

Many of the fundamental results of experimental neutrino physics have been provided by large underground neutrino detectors such as SNO and Super-Kamiokande. These couple a large target mass with the shielding that several hundred metres of rock provides against many sources of background events (cosmic rays, for instance). The next generation of experiments will require an order of magnitude increase in detector size to reach new physics goals.

A large, high-mass detector on the order of 100 kton to 1 Mton would provide sensitivity for a number of new scientific opportunities, including a precision measurement of the neutrino mixing angle θ_{13} as well as sensitivity to the $\mathcal{CP}$ violating phase δ (and hence to leptonic $\mathcal{CP}$ violation measurements). In addition to this conventional neutrino physics programme, such a detector would be able to study neutrinos from supernovae, permitting studies of the mechanisms of the explosions, as well as having a sensitivity to proton decay with lifetimes of up to 10^{35} years, a number of significance to several theoretical models [39].

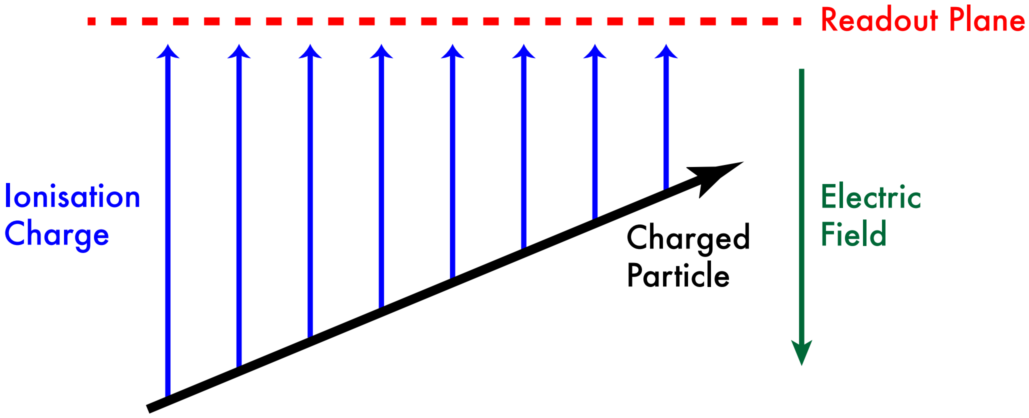


Figure 3.1: Schematic diagram of the operation of a time-projection chamber. An incident ionising particle produces ionisation charge, which drifts in an applied electric field to a readout plane. The readout plane provides a two-dimensional (x, y) position measurement, and the depth z can be determined from the drift time t taken for charge to arrive at the readout plane.

3.3 Liquid Argon Time Projection Chambers

Ionising radiation moving through Liquid Argon (LAr) generates both ionisation charge and scintillation light signals, though the energy resolution of the scintillation light is typically poor compared to that of the ionisation charge, since only a small number of the emitted photons can usually be detected [38]. Liquid Argon Time-Projection Chambers were first proposed by Carlo Rubbia in 1977 [40], when he noted that LAr has several useful properties as a detector medium, including high density, long electron drift times, and relatively high abundance making it cheap to obtain in large quantities.

The ionisation charge produced by a primary ionising particle can be drifted through an applied electric field to a readout plane (anode), see figure 3.1.

3.3.1 Production of Ionisation Charge

The amount of ionisation charge Q that drifts to a readout plane is smaller than that initially produced by the passage of an ionising particle Q_0, since some of the electron-ion pairs undergo recombination processes, i.e. $Q = \mathcal{R}Q_0$ with some *quenching factor* $\mathcal{R}$. The recombination is described empirically by Birk's Law [41]:

$$Q = \frac{Q_0}{1 + k_E/\epsilon} \tag{3.1}$$

where k_E is a recombination constant, to be fitted to data, and ϵ is the applied electric field strength. In terms of the stopping power dE/dx, this is written:

$$Q = \frac{Q_0}{1 + k_Q\, dE/dx} \tag{3.2}$$

with $k_Q(\epsilon) = k_E/\epsilon$.

The Icarus experiment [42] measured the quenching factor $\mathcal{R}$ as a function of the applied field and dE/dx, finding $\mathcal{R} = 0.71$ for dE/dx from minimum ionising particles up to $\sim 30\,\mathrm{MeV\,g^{-1}\,cm^2}$ in an applied field of $0.5\,\mathrm{kV\,cm^{-1}}$. They found, however, that their data was best described with an additional normalisation constant A:

$$Q = A\frac{Q_0}{1 + k_E/\epsilon\, dE/dx} \tag{3.3}$$

with $A = 0.800 \pm 0.003$ and $k_E = 0.0486 \pm 0.0006\,\mathrm{kV\,cm^{-1}}$. This updated quenching factor has been applied in the Warwick *Lamu* simulation package for liquid Argon detectors (see chapter 3.5 for further details).

3.3.2 Drifting Charge

Charge transport through a medium such as LAr is governed by the mobility μ of charge carriers (electrons) [38]. The drift velocity $\mathbf{v}$ is:

$$\mathbf{v} = \mu\mathbf{E} \tag{3.4}$$

where μ is constant for low electric fields $\mathbf{E}$. An electron is considered to be *quasifree* when the mobility in the absence of an electric field, μ_0, is greater than $10\,\mathrm{cm^2\,V^{-1}\,s^{-1}}$. For LAr, where $\mu_0 = 625 \pm 15\,\mathrm{cm^2\,V^{-1}\,s^{-1}}$ [38], the drift velocity is large and drift lengths are limited primarily by impurities in the Argon, which capture the charge.

Another constraint on the drift length is the reduction in spatial resolution due to *transverse diffusion* of drifting charge. As the charge drifts from the point of creation to the readout plane, diffusion perpendicular to the direction of motion acts to spread out the electron cloud. For long drift lengths, this reduces the xy resolution which can be achieved at the readout plane.

3.3.3 Wire Readout

In most current and planned TPCs the readout plane consists of a grid of wires, attached to charge amplifiers and subsequent electronics to read the charge signal that collects on each wire. Coincidence of signals on x and y wires provides a two-dimensional view of charge arriving at the readout plane. The Icarus T600 [43] detector, with a 500 ton fiducial mass, demonstrated the success of wire plane readout on a small scale. Projects including MODULAr [44] plan to carry the concept to larger detectors by grouping many Icarus-style modules together. Alternative plans by collaborations such as LANNDD [45] use a single LAr volume, internally segmented by readout planes.

Whilst the wires are inexpensive, large wire plane detectors require many electronics (readout) channels, each of which carries significant cost. For a 100 kton detector, this cost is substantial, and can be reduced only by using longer wires (which requires higher tension in the wires themselves, and presents an engineering challenge) or by increasing the inter-wire spacing, thus reducing the spatial resolution of the detector. Another option would be to increase the drift length, but this also results in degraded spatial resolution due to transverse diffusion of charge as it drifts.

3.3.4 Thick Gas Electron Multipliers

A Thick Gas Electron Multiplier (TGEM) is a Printed Circuit Board (PCB) of thickness on the order of 1 mm, with copper cladding on both sides. Holes are drilled into the PCB at regular intervals, with diameters on the order of 1 mm and spaced by < 1 mm. The holes may be further shaped by etching e.g. to smooth rough edges from the drilling process. An electric potential is applied across the two Copper electrodes, creating a strong field in the holes and extending into the region around them. Electrons arriving at the TGEM are focused into the holes by this field. The high field strength ($> 1.5\,\mathrm{kV\,mm^{-1}}$) within the holes causes rapid acceleration of the electrons.

If the TGEM is in gaseous Argon, the accelerated electrons cause ionisation of Argon in the hole, freeing more electrons in an avalanche process. In this way, the TGEM acts as a charge multiplier. In liquid Argon, the mean free path of electrons is much shorter than in gas, and the electrons do not gain sufficient energy to ionise an Argon atom (the ionisation energy of Argon is 23.4 eV [38]). Some of the electrons do, however, gain enough energy to excite electrons in Argon atoms (11.55 eV threshold [46]), which subsequently emit photons and allow for optical readout of the TGEM [47].

The light produced by excitation processes in the holes of a TGEM immersed in liquid Argon could be imaged and a reconstruction procedure applied to obtain the (x, y) position of the hole(s) in which light was produced. As an alternative to wire plane readout, this optical readout procedure could have a much lower cost, due primarily to a reduction in the number of readout channels required. One approach would be to image a large area (on the order of $1\,\mathrm{m}^2$) with a single Charge Coupled Device (CCD) camera. This scenario is complicated by the necessity to focus the light onto the CCD, and the time resolution which could be achieved is limited by the frame rate of the CCD. In order for the time resolution to match that of the xy plane, a CCD that operates at frequencies of $\sim 1\,\mathrm{MHz}$ (and with high efficiency in the cold environment of LAr) would be required. Such devices do not yet exist.

The second option is to use a sparse array of Silicon photomultipliers. These devices have high quantum efficiency and provide single photon sensitivity. Using a grid of silicon photomultipliers, it is possible to determine the position of a light source based on the relative intensities recorded at several detectors (see chapter 3.4).

3.4 Point Light Source Reconstruction

In order that the position resolution and associated measurement uncertainty associated with reconstructing point light sources on a plane could be estimated, I wrote a simulation that produces a *flux map* for a detector plane some distance Δz from a plane of point light sources. This detector flux map is then used to calculate the integrated light intensity seen by a photosensitive device with a given area, active area and quantum efficiency. Such device areas can be positioned on the detector plane, and the output from the simulation then provides a list of intensities at each detector from a given distribution of sources. A visualisation of this output is show in figure 3.2.

This simulation was used by G. Rutter and M. Richards to develop an iterative algorithm for position reconstruction [1]. The algorithm requires an initial estimate of the position of a source, which can be obtained from a global approach such as that used to reconstruct position in Anger cameras [48]. Once this seed position is established, the closest 2×2 square of photosensors surrounding the source is chosen and the reconstructed position iteratively refined by assuming that light from the source propagates freely (i.e. spherically). The ratio of intensities seen by detectors D_1 and D_2 is given by [1]:

$$\alpha = \frac{I_1}{I_2} = \frac{r_2^2}{r_1^2} = \frac{x_2^2 + y_2^2 + h^2}{x_1^2 + y_1^2 + h^2} \tag{3.5}$$

where the height h is fixed and the coordinates are distances to the detectors D_1

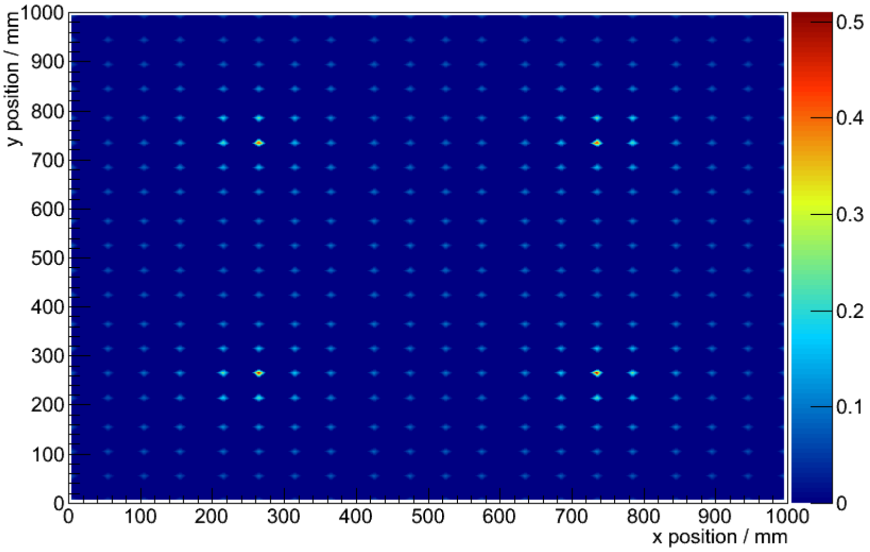


Figure 3.2: Simulation of light intensity recorded by a sparse array of photodetectors from four light sources, normalised to the intensity of the sources. The sources are located at $(0.25, 0.25)$ m, $(0.25, 0.75)$ m, $(0.75, 0.25)$ m and $(0.75, 0.75)$ m.

and D_2. A similar ratio β is constructed for detectors D_1 and D_3. The detector which is shared between the two ratios (D_1) is the one which recorded the highest light intensity. The radius r is from source to detector.

Since the inter-detector spacing is fixed, the value of x_1 fixes the value of x_2, and similarly for the y values. By fixing x values, the y values can be recomputed using equation (3.5), providing an improved estimate of the y position of the source. This process can be repeated with the y values fixed to improve the x estimate, iteratively converging on the true source position until the difference in coordinates between iterations becomes small compared with other experimental uncertainties (or to a predefined tolerance), see figure 3.3. The total uncertainty resulting from this procedure is shown as a function of the distance between source and detector planes in figure 3.4, where the detectors are arranged in a square grid over a $1\,\mathrm{m}^2$ area, with the quoted number of detectors per row, and the reconstruction is of a single point light source. This provides an optimisation procedure for determining the detector coverage required to obtain the desired uncertainty on position resolution.

The spatial resolution is limited by statistical uncertainty, which gives rise to the curved shape seen in figure 3.4. A small distance between the source and detector planes yields small effective photodetector areas, reduction in the number of photons

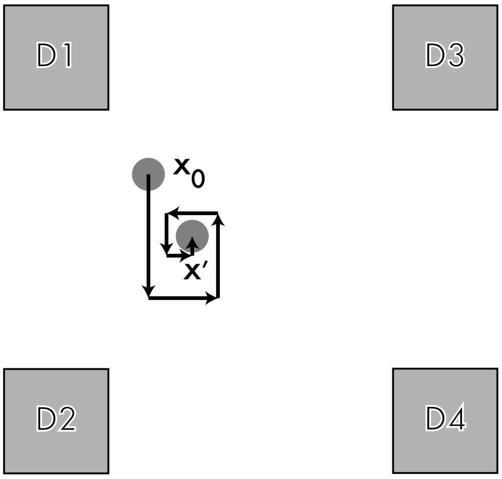


Figure 3.3: Iterative reconstruction of the position of a point light source using the ratio of light intensities captured by the surrounding photodetectors. Each step updates either the x or y coordinate by constraining the y or x coordinate and using the known spacing between detectors.

received from the source, and a corresponding deterioration in resolution. A large distance results in reduced photon flux over the active area of the photodetector, and again reduces the spatial resolution. These effects generate the curves displayed in the figure, which can be viewed as an optimisation of the spacing between the source and detector planes, for a given array geometry.

The advantage of using this iterative procedure over simply fitting the intensity measurements as for an Anger camera is that the iteration can continue until the uncertainty on the position of the light source falls below some threshold. Fitting provides a source position with some associated uncertainty, but the iterative procedure can be used to reduce this uncertainty to a chosen acceptable level, matching the desired position resolution of the detector. If the photodetector grid layout does not physically provide sufficient resolution, continued iteration of the algorithm will eventually provide no improvement to the knowledge of the position of the light source, and this is the basis for the optimisation described above.

3.5 The Lamu Simulation

Lamu is a Geant4 [49] simulation for liquid Argon volumes which consist of cylinders of configurable radius and height. It includes full modelling of physics pro-

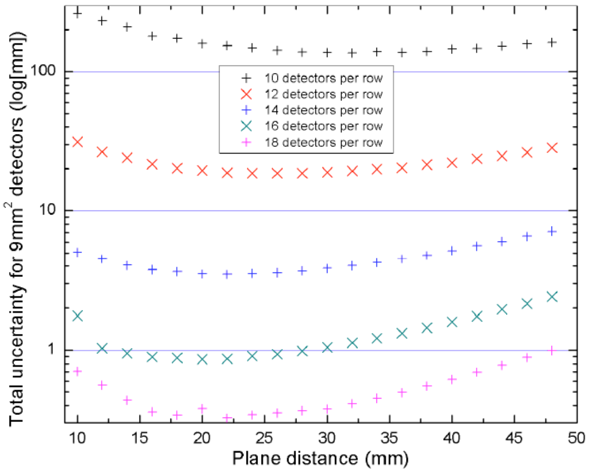


Figure 3.4: The uncertainty on the reconstructed position of a point light source resulting from the iterative procedure discussed in the text, as a function of source–detector plane spacing, for five different sparse array densities (detector–detector spacings). Figure taken from [1].

cesses according to the QGSP__BIC__HP[1] physics list, with all electromagnetic and hadronic processes enabled. The energy loss of particles is modelled taking into account the quenching described in chapter 3.3.1, and the LAr volume is divided into $(3 \times 3 \times 3)\,\mathrm{mm}^3$ cubic voxels.[2] Charged particles are tracked through these voxels to zero energy or until they leave the detector, and the energy deposited in each voxel leads to a mapping between voxel coordinates and the total energy deposited within that voxel. The output from the simulation is this mapping, without any attempt to model drifting charge, readout technologies or subsequent electronics, all of which are dependent on experiment-specific details. A LAr specific quenching factor is used to scale the energy deposited in each voxel by the passage of a charged particle through it. This factor is described in chapter 3.3.1. The result is a three-dimensional point cloud corresponding to the energy deposition in the detector volume by a particular event. Events are stored in ROOT [50] files for subsequent analysis.

[1] QGSP__BIC__HP is a standard Geant4 physics list, which applies a string model for the interactions of high energy hadrons. In addition to the basic string model (QGSP), the list uses binary cascade modelling (BIC) for primary protons and neutrons below 10 GeV and high-precision data (HP) for low energy neutrons. See `http://geant4.cern.ch/support/proc_mod_catalog/physics_lists/referencePL.shtml` for details.

[2] The 3 mm voxel side length was chosen to match the wire pitch of existing LAr TPC detectors, including the Icarus [43] detector.

3.5.1 Readout and Digitization

Lamu does not currently simulate readout and digitization electronics or the effects of projection of hits onto readout planes. This is done because the simulation is not specific to any readout technology. The output from Lamu is currently at the level of voxellised hits in 3D, taking into account the resolution that real detector readout options would provide.

Projection effects could be taken into account in the simulation, but would need to include finite timing measurements for the drift of charge through the TPC volume in order to simulate the imperfect projection that would be seen in a real TPC. Cross-talk, digitization and multiplicity of hits on readout planes are not taken into account in Lamu since they would be specific to a readout technology.

While these effects have a negative impact on the resolution of the final 3D reconstructed hits, the Lamu simulation attempts to emulate the combination of readout effects with its 3 mm voxel size.

Depending on the readout technology used, different effects would have to be considered in order to accurately simulate the signal processing and digitization steps. A number of these are discussed below:

- Readout from induction (i.e. non-charge collecting) planes is complicated by low signal to noise ratio compared to collection planes. Since the planes are typically in different orientations, this introduces asymmetry in the spatial uncertainties, complicating subsequent 3D hit reconstruction.

- In many wire-plane detectors, the first instrumented plane is not electrostatically shielded from the drift volume by a non-instrumented plane (or Frisch grid). This increases the noise on the first induction plane.

- Pure two-dimensional readout schemes, including strip readout or two wire planes, suffer systematically from ghost hits when two or more hits arrive simultaneously. This poses a challenge for the readout of dense hit structures such as those obtained from electromagnetic showers. A third readout plane breaks the degeneracy, resulting in improved hit reconstruction.

- Hit reconstruction should take into account the possibility of broken wires, which result in an effective local reduction in the number of readout planes, and may reintroduce the two-dimensional degeneracy issues.

- Cross-talk between wires or strips in a readout plane can result in noisy or ghost signals. This is a problem for events with high hit multiplicity, where the charge measurement becomes unreliable, although the position information should remain valid.

3.5.2 Characterisation & Testing of the Lamu Simulation

In order to characterise the Lamu simulation, a number of tests were performed using simulated single particles produced by the Geant4 General Particle Source (GPS) generator [51].

In the first test, muons of fixed kinetic energy were started at the centre of a cylindrical detector volume, with radius R and height $H = 2R$. The radius was initially set to 1 m and 100 events were generated. The radially outermost hit in each event was found, and the number of events with such hits lying within 10 cm of the detector wall recorded. The radius was increased in 1 m intervals until greater than 95% of the events were fully contained, i.e. the outermost hit was more than 10 cm from the wall. This study was performed at energies from 100 MeV to 5000 MeV in 100 MeV intervals, and the results are shown in figure 3.5.

The purpose of this study was to determine the maximum range of a muon of a given energy in liquid Argon, enabling the remainder of the work in this thesis to be done in the context of a simulation with geometry chosen so as to fully contain a muon produced at the centre of the detector. For example, all ν_μ charged current events with an initial neutrino energy of 4500 MeV were simulated in a detector of 25 m radius, in order to ensure that the resulting muons were fully contained.

The second test relates to the ionisation charge quenching factor built into the Lamu simulation. For this test, particles of several species were tracked through the detector with a fixed initial kinetic energy, and the total energy deposited was recorded by summing the energy deposited in each voxel. The results of this study are shown in figure 3.6, giving the ratio of energy deposited in the simulated LAr volume to the initial kinetic energy of the particle. This is plotted separately for muons, protons, electrons, and charged pions.

For each plot, the calculation of energy deposited took into account only hits produced by a particle of the species under consideration. For muons, the ratio is fairly consistent with the quenching formula since a muon will behave much the same irrespective of energy (within the range considered). For protons and charged pions, the higher energy particles are more likely to undergo hadronic interactions resulting in particles of different species. In this way, the energy deposited by protons drops as the proton energy increases, since most of the energy is carried away by products of hadronic interactions. The same applies for pions.

In the case of electrons, as the energy increases, the chance of an electromag-

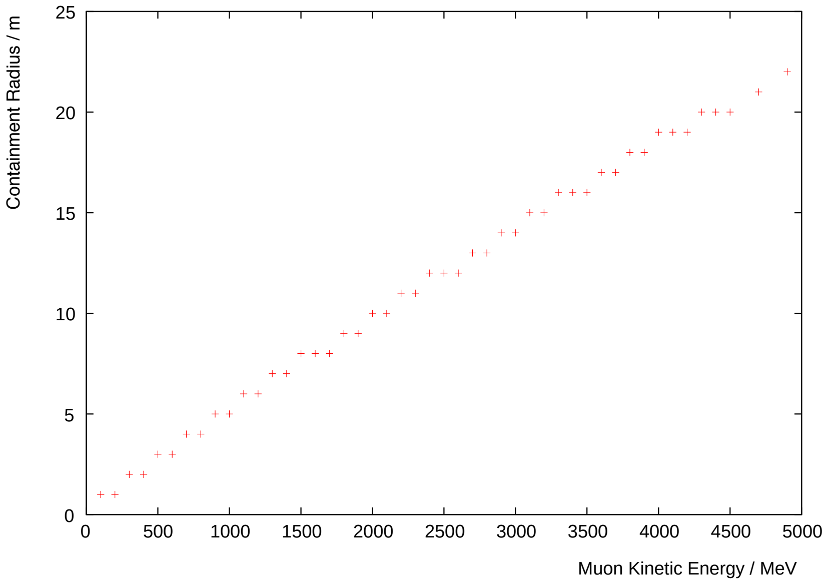
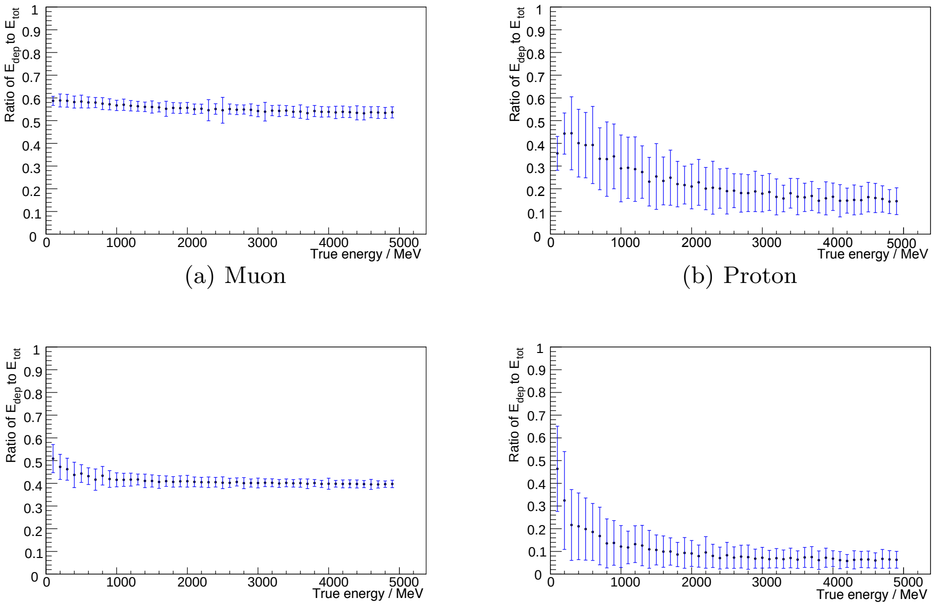


Figure 3.5: Minimum detector radius required in order to fully contain muon tracks, as a function of the muon kinetic energy. This information is used to ensure that subsequent studies have fully contained primary particles.

netic shower occurring also increases in analogy to the hadronic showers of protons and charged pions. However, the ratio curve plotted in figure 3.6(c) does not show a significant drop since the (charged) products of an electromagnetic shower are themselves electrons, and all energy deposited by electrons was counted towards the deposition from the primary particle.

3.6 The TrackGen Toy Track Simulation

In addition to the full physics simulation provided by Lamu (see chapter 3.5), it is useful to be able to test algorithms for event reconstruction using simplified “toy” tracks. The TrackGen simulation is a Python program which produces events containing one or more straight line segments, tracked through cubic voxels with “energy” deposition $L \cdot dE/dx$ where L is the length of the track segment through a voxel and dE/dx is a constant ‘energy loss’ per unit length for the track concerned. The calculated ‘energy’ is deposited at the centre of each voxel. There is no attempt to model physics processes such as Bethe-Bloch energy loss or scattering. Similarly, no detector properties are taken into account. This produces very clean (but still



(a) Muon

(b) Proton

(c) Electron

(d) Charged Pion

Figure 3.6: Ratio of energy deposited in the Lamu simulation to the initial kinetic energy of the particle, by particle species (errors are statistical). See text for discussion.

voxellised) events which, in principle, can be used to test the baseline efficiency of a reconstruction algorithm without dealing with a combination of algorithm effects and physics effects.

The core of the TrackGen simulation deals with calculating the intersection of straight lines with a set of voxels, determining the segment length through a voxel, and keeping track of the energy deposits. It is possible to set the voxel size (as a parameter to the TrackGen program) but only cubic voxels are supported, i.e. a single side length is specified, rather than the more general three that would be required for arbitrary cuboid voxels. In addition to the core tracker, several event generator modules produce different sets of straight lines, corresponding to different event topologies of interest. The topologies of most relevance to the reconstruction of neutrino events are:

1. Single straight line; corresponding to a muon track without scattering.
2. Single line with kink; corresponding to a muon track with a single large scatter.
3. Two lines originating at a vertex, with fixed opening angle; corresponding to an interaction vertex and e.g. $\mu + p$ final state.

4. Multiple lines originating at a vertex; corresponding to a higher-energy ν interaction producing multiple final state particles.

The TrackGen simulation has event generator modules for each of these topologies, producing events consisting of one or more straight line segments, each tracked through a voxellised space. Output is to an SQLite3 [52] database file with a simple table structure.

3.7 Neutrino Event Generation with Genie

The Genie [53] neutrino event generator is used to generate sets of final state particles (including momenta) from the interactions of neutrinos of a given flavour on Argon nuclei at a variety of energies. These final state particles are fed into the Lamu simulation as input trajectories and tracked through a LAr volume.

Genie allows for the selection of interaction type, e.g. Charged Current Quasi-Elastic (CCQE) scattering, but it is recommended that users limit themselves to selecting only charged current or neutral current interactions, since there is not a one-to-one mapping between physical interaction and final state topology, and physicists are mostly concerned with final state topologies. For instance, when one talks about a CCQE event, one typically thinks of a $\mu + p$ final state. The reality is that a $\mu + p$ final state can occur in a number of ways, with CCQE interactions a major contributor. Furthermore, a CCQE interaction can produce other final states, particularly if the resulting nucleon undergoes further interactions in the nucleus before emerging (see figure 3.7 for illustration). For this reason, the events generated for this thesis using Genie were made by specifying only charged or neutral current interactions, without imposing limitations on the subtype of the interaction. The results were subsequently filtered to “cherrypick” the topologies of interest, and a note made of the fraction of the total events generated which passed the filter. In the remainder of this thesis, the term CCQE is used, for convenience, to mean “*charged current interactions producing a $\mu + p$ final state*”.

3.8 Neutrino Experiments Using Liquid Argon TPCs

A number of existing and planned experiments use a LAr TPC. This section summarises the main experiments, including notable prototypes or future planned experiments, describing the technology used in the detector, the status of any software framework, and the issues faced by those experiments.

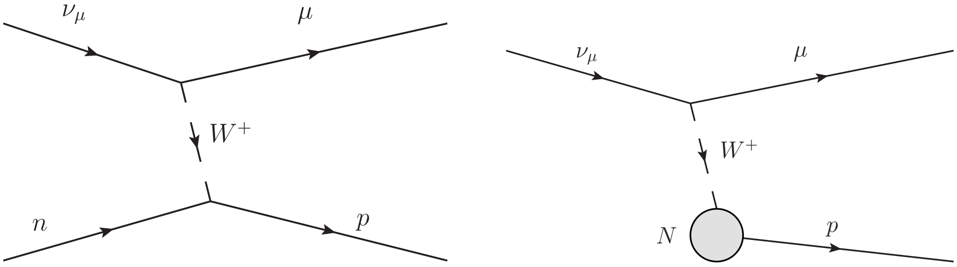


Figure 3.7: The difference between true charged current quasi-elastic (CCQE) interactions (left), and charged current interactions producing a $\mu + p$ final state (right). The left-hand diagram is included in the possibilities for the right-hand diagram, but in a more general sense any interaction could occur in the nucleus, and we see only the products that leave. Genie allows us to select events based on these products, leaving the details of the interaction that produced them to be decided by Genie itself.

3.8.1 The Fermilab Liquid Argon Experiments

A number of proposed and running experiments at Fermilab use LAr TPCs. There is a software framework called *LArSoft*, which will be shared by the ArgoNeuT [54], MicroBooNE [55] and LBNE [56] experiments.

The LArSoft Framework

LArSoft aims to be a complete set of reconstruction, simulation and analysis tools, shared by the liquid Argon neutrino experiments based at Fermilab. It accepts data from GENIE and Geant simulations, as well as real data from the ArgoNeuT experiment. The development goals are to provide wire calibration, hit finding and clustering, endpoint finding, 3D tracking, 3D shower finding, vertex finding and calorimetry [57], but at present LArSoft does not provide a completely automated reconstruction chain. The sections below on each Fermilab experiment contain further details of the use of the LArSoft framework for their reconstruction tasks.

ArgoNeuT

ArgoNeuT [54] is a small LAr TPC exposed to the Fermilab NuMI (Neutrinos at the Main Injector) beam. ArgoNeuT is the first phase of Fermilab's path towards using a massive LAr TPC in the LBNE experiment.

The detector itself consists of a vacuum-insulated cryostat holding 550 litres of liquid Argon, with a TPC active area of $40 \times 47 \times 90\,\text{cm}^3$, or approximately

170 litres. The maximum drift length is 47 cm and there is a uniform electric field of $500\,\mathrm{V\,cm^{-1}}$ between the cathode and the anode, which consists of three wire planes with 4 mm wire pitch in each plane and a 4 mm spacing between planes. Liquid Argon is circulated through a closed-loop cooling and filtration system which removes electronegative impurities.

The first (*shield*) wire plane is not instrumented, and acts to shield the outer two planes from induced signals created by drifting electrons in the main TPC volume. The second (*induction*) plane consists of 240 wires at $+60°$ to the beam axis. Drifting electrons induce a signal in these wires after passing the shield plane. The third (*collection*) plane consists of 240 wires at $-60°$ to the beam axis, and collects the electrons at the end of their drift.

At each beam spill, 2×240 signals are collected from the wires, pre-amplified, filtered and digitized. A wire-plane *hit* is characterised by the peak signal amplitude and the coordinates in the (*wire–time*) plane. Hits from both planes are reconstructed into space coordinates associated with the tracks of ionising particles through the drift volume and associated with the energy deposited at those coordinates.

The DBSCAN algorithm (described in chapter 4.4.1) is used to cluster hits into tracks, using an elliptical neighbourhood based on the geometry of the wire planes. Lines are found in each 2D plane of clustered hits using the Hough transform (described in appendix A.2), and those lines with similar slopes and connecting endpoints are merged together on a plane by plane basis. At this stage, candidate interaction points are found by looking at the longest 2D clusters in each plane. Some confusion is caused by δ rays, which must be filtered out.

Tracks found in each plane are combined into 3D by matching the drift coordinates of each hit and requiring that the hits coincide within a time window allowing for the drift time between planes. Geometric data is provided in the form of direction cosines and the effective length of the track portion that was exposed to a single wire. This allows for the computation of dE/dx information, which is corrected using Birk's model for electron recombination.

MicroBooNE

The MicroBooNE [55] experiment has three primary objectives; to resolve the source of the MiniBooNE low energy excess,[3] to measure a variety of neutrino cross-sections, and to provide technical experience in the construction and operation of a large LAr TPC in preparation for LBNE and future experiments.

[3] MiniBooNE saw a 3σ excess of events producing low energy e^- or γ final states [58].

The detector will consist of a $2.325 \times 2.5604 \times 10.368\,\mathrm{m}^3$ liquid Argon volume with the longest dimension along the beam. The readout will consist of three instrumented wire planes on a 3 mm wire pitch and with 3 mm separation. A drift field of $500\,\mathrm{V\,cm}^{-1}$ will be applied across the $61.8\,\mathrm{m}^3$ drift volume (approximately 86 tonnes of liquid Argon). Pre-amplification and pulse shaping will be performed inside the cryostat with cold electronics, before further amplification stages outside the cryostat, leading to the eventual digitization of the pulses.

The offline software for MicroBooNE uses the LArSoft toolkit, and includes a simulation using Geant4. The first stages of reconstruction calibrate the wire signals and localise them into spatial hits. These are grouped into clusters, then further grouped into *prongs*, which represent either tracks or showers. Vertex finding proceeds by looking for prongs originating from a common point. The clustering is performed in two dimensions (wire-plane projections) using DBSCAN and edge or vertex finding algorithms based on the Harris function (see chapter 4.5.1). Track finding is performed in 2D using the Hough transform, with matched lines being combined into 3D tracks. LArSoft does not currently provide a way to perform full three-dimensional track reconstruction without first using two-dimensional projections.

MicroBooNE faces a number of challenges during the construction and operation, including the readout of a large number of wire channels in cryogenic environments, maintaining liquid Argon purity in a large detector without resorting to vacuum processes, ensuring that wires do not break inside the large cryostat volume, and demonstrating successful simulation and reconstruction of events in a large LAr TPC.

LBNE

The LBNE [56] collaboration aim to build a long baseline neutrino oscillation experiment using a large LAr TPC as the far detector. With a 1300 km baseline and an intense neutrino beam providing neutrino energies between 0.5 GeV and 5.0 GeV, the aim is to make precision measurements of θ_{13} and to determine the $\mathcal{CP}$ violating phase δ_{CP} using a combination of traditional near detectors at Fermilab, and a 10 kton LAr TPC far detector.

The LAr TPC will consist of two cryostats built side-by-side, with their long axes parallel to the beam, and separated by a wall which is centered on the beam centre. Each cryostat will be $13.9 \times 14 \times 25.3\,\mathrm{m}^3$ and have a fiducial mass of 5 kton or a total mass of 6.7 kton (giving 13.5 kton for the entire far detector).

The TPC itself will consist of alternating rows of anode and cathode planes with a uniform electric field of 500 V cm^{-1} in the drift spaces. The maximum drift distance (i.e. the separation of the planes) is 2.3 m, and the anode will consist of three instrumented wire planes (two induction and one collection) and a fourth, uninstrumented, plane to shield the others from electrostatic discharge and improve signal-to-noise ratio on the first induction plane. The planes will be oriented vertically and at ±45° to the vertical, improving resolution along the beam compared to the ±60° configuration of ArgoNeuT and MicroBooNE, and will have a 4.5 mm wire pitch. With 2650 wires in each anode plane, there will be a total of 153600 readout channels per cryostat. Readout electronics will be in the liquid Argon, and will include pre-amplification, pulse shaping and digitization.

LBNE will use the LArSoft framework and take advantage of the reconstruction procedures demonstrated by both ArgoNeuT and MicroBooNE.

3.8.2 Laguna LBNO

Laguna-LBNO [59] is a planned 2300 km long baseline neutrino oscillation experiment in Europe, utilising a beam from CERN, with a far detector consisting of a 20 kton LAr TPC and a magnetised Iron calorimeter. Its physics goals include measuring δ_{CP} by directly observing CP violation, and attempting to measure the neutrino mass hierarchy. The far detector is to be built deep underground (approximately 1440 m, or 4000 m water equivalent), which also allows for the study of a number of low event rate phenomena. The LAr TPC will use Large Electron Multiplier technology for its readout planes, most likely employing a strip readout similar to that described in [60]. The drift field will be between 500 V cm^{-1} and 1000 V cm^{-1}. One challenge faced by the experiment will be achieving sufficiently high liquid Argon purity to allow for very long drift lengths in a single, non-evacuated tank.

Reconstruction will be provided by the QSCAN framework, which was originally developed for the Icarus 50L 15T and T600 experiments. It provides noise reduction and hit finding, as well as clustering and track reconstruction in two-dimensional planes, using the Hough transform. Three-dimensional reconstruction is once again achieved by combining information from multiple two-dimensional planes.

3.8.3 Icarus T600

The Icarus T600 detector [61] is a 600 ton LAr TPC operating at the Gran Sasso National Laboratory in the CNGS (CERN to Gran Sasso) neutrino beam. It has three readout wire planes, at $0^\circ, +60^\circ$ and -60° to the vertical, all with a 3 mm wire pitch.

The first stage of reconstruction consists of converting individual wire signals into hits by recording their spatial position and calorimetric information. These are clustered into 2D structures in each wire plane, which are categorised as tracks or showers. The DBSCAN algorithm has been proposed for this clustering. Three-dimensional reconstruction is performed by matching clusters in at least two 2D planes, there is no fully three-dimensional reconstruction at present.

4 The Latte Framework

4.1 Introduction

The Latte framework forms the software basis for almost all of the reconstruction algorithms described in this thesis. It acts as both a set of utilities and a unifying central control mechanism. This chapter discusses several of the algorithms provided, ending with an overview of the control structures of the Latte framework itself. Some components of Latte form a significant contribution to the work in this thesis, and are discussed separately in subsequent chapters.

4.2 Nearest Neighbour Search using KDTree

The k-dimensional tree (KDTree) is a data structure for partitioning a k-dimensional space by representing the points as nodes in a binary tree. Using a KDTree it is possible to perform nearest-neighbour search in $O(\log N)$ time [62],[1] compared with the brute-force search complexity of $O(N^2)$. An introductory tutorial on k-dimensional trees appears in [63]. The KDTree is not a clustering algorithm, but its rapid near-neighbour search forms a central part of several clustering algorithms, as well as being used for the charge weighting procedure (see chapter 4.3) and the cell generation (chapter 5.3.3) stage of the Cellular Automaton track reconstruction procedure.

The implementation currently used is that of the *SciPy* [64] library of Python code for scientific computing.

[1] N is the number of data points to be partitioned. The KDTree itself is built in $O(kN \log N)$ time.

4.3 Charge Weighting

Data from a LAr TPC is assumed to be in the form of (x, y, z) voxels with an associated charge deposit Q. The voxel shape is determined entirely by the readout mechanism; a 2D readout plane determines the xy resolution while the readout sampling rate determines the z resolution as expected for a Time-Projection Chamber (TPC). This structure is retained in the Geant4 simulation through the use of cubic voxels with a side length of 1 mm. In order to transform this data into a representation more closely resembling the true passage of ionising radiation through the detector, a charge weighting procedure is applied. This procedure adjusts the spatial coordinates of each charge deposit (voxel) by taking the charge-weighted average of the spatial coordinates of all hits within a sphere of some radius (default: 2 mm) centered on that voxel:

$$\mathbf{x}' = \frac{\sum\limits_i \mathbf{x}_i Q_i}{\sum\limits_i Q_i} \tag{4.1}$$

The neighbouring hits are discovered using the KDTree algorithm described in chapter 4.2.

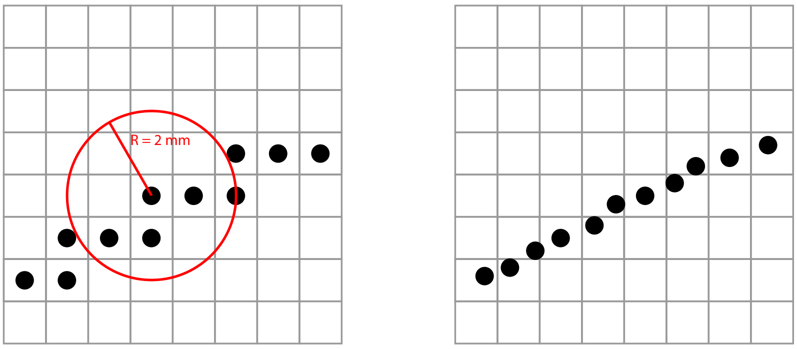
Charge weighting[2] in this way moves the position of each charge deposit closer to the local charge cloud, bringing track-like charge deposits closer to a straight line (see figure 4.1). For each point, the sum is taken over the unweighted positions of neighbours and a new dataset is generated, containing the weighted positions. This results in a stable, deterministic output.

4.4 Density-based Clustering

4.4.1 DBSCAN

The Density-Based Spatial Clustering of Applications with Noise (DBSCAN) algorithm was proposed in 1996 as a means for clustering spatial data based on the varying densities of point clouds [65]. The algorithm is characterised by its requirement that, for the neighbourhood ϵ around a given point in the cluster, the number of points N in ϵ must exceed some threshold value N_{min}. In this manner, the clustering is determined by identifying areas of high point density. Areas of low point

[2] This is not, strictly speaking, the usual definition of charge weighting, in which a set of hits is reduced to a smaller set of charge-weighted hits, but more of a smoothing process to reduce the geometric effects of the readout system. Hits are retained here primarily to facilitate accessing truth information.



(a) Hits as central points of voxels

(b) Hits shifted from voxel centres

Figure 4.1: Schematic diagram of the effect of the charge weighting procedure on hit positions. In (a) each hit is positioned at the centre of a voxel. In (b) the hit positions have been adjusted based on the charge-weighted positions of surrounding hits within a 2 mm radius; this shifts them from the voxel centres and out of a regular grid structure.

density (i.e. any points not clustered) are identified as *noise* and clustered together as such.

The DBSCAN algorithm has been applied to spatial data obtained from a LAr TPC by the ArgoNeuT experiment [66] to identify clusters in two 2D views, which are subsequently recombined into 3D based on wire readout timing information.

DBSCAN has two configurable parameters; ϵ, the radius around a given point within which neighbours must lie, and N_{min}, the minimum number of those neighbours for a point to be considered part of a dense cluster. While these parameters can be optimised, they remain global, and DBSCAN will not typically identify regions of changing density and cluster accordingly. This tends to result in a DBSCAN clustering which wraps around the vertex of charged-current neutrino events (for example).

4.4.2 OPTICS

The Ordering Points To Identify the Clustering Structure (OPTICS) algorithm [67] extends DBSCAN by performing a *hierarchical clustering*; an ordering operation which is equivalent to returning the density-based clusterings associated with a broad range of values of the parameter ϵ. For example, given a fixed value of N_{min}, clusters obtained with a small ϵ (that is, high density clusters) are completely con-

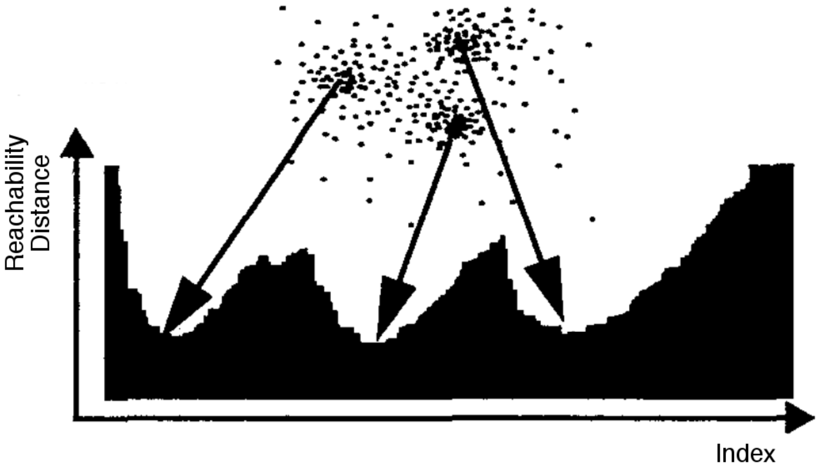


Figure 4.2: An example of a reachability plot and the clusters found on one scale. The reachability plot allows clusters on any scale to be found by choosing an appropriate reachability distance threshold (y axis value). Figure adapted from [67].

tained within the clusters obtained for a larger ϵ (lower density). OPTICS exploits this relationship to provide information about the clustering on all ϵ scales, allowing clusters to be extracted from spatial data with varying densities.

During the execution of the OPTICS algorithm, an array is built up in which the points from the input data set are sorted according to proximity and density. Points which are close to each other in the input data set will be close together in the array, which is one-dimensional. OPTICS produces a reachability plot with the indices into this array on the x axis and a reachability distance on the y axis. An example of a reachability plot is shown in figure 4.2.

It is this reachability plot that gives OPTICS the ability to simultaneously cluster on all density scales. Each index has an associated reachability distance, and clusters are represented by valleys in the plot, separated by peaks of high reachability distance. Depending on the clustering scale desired, the plot can be scanned for peaks above a certain threshold. Between each threshold index, the points contained are associated to a single cluster.

4.5 Feature Detection

Feature detection refers to the procedure of locating interest points within an image or event. Latte provides two-dimensional feature detection using the method described in [68]. Three-dimensional feature detection is currently based on running the two-dimensional feature finder in multiple projections and combining the results.

4.5.1 Two-dimensional Feature Detection

Two-dimensional feature detection uses the *structure tensor* S, which is a second-moment matrix defined in terms of the intensity variation of a two-dimensional image: [68]

$$S(x,y) = g(\sigma_s) * \begin{bmatrix} I_x(x,y)^2 & I_x(x,y)I_y(x,y) \\ I_x(x,y)I_y(x,y) & I_y(x,y)^2 \end{bmatrix} \tag{4.2}$$

where $*$ represents convolution, and $I_x(x,y)$ is the partial derivative of the image with respect to x (similarly for $I_y(x,y)$). A Gaussian window $g(\sigma_s)$ is defined as

$$g(\sigma) = exp\left(\frac{-(x^2+y^2)}{2\sigma^2}\right) \tag{4.3}$$

and is used for averaging, as well as noise-reduction.

The eigenvectors and eigenvalues of $S(x,y)$ describe *local* intensity variations within the image. Functions of $S(x,y)$ can therefore be constructed such that their local maxima correspond to features of a desired type. For example, the 'V'-like corner of a particle decay vertex will have large intensity changes in all directions. Since it is this type of feature that we are interested in, the Förstner-Noble response function $R_N(x,y)$ is used within Latte:

$$R_N(x,y) = \begin{cases} \frac{\det S}{\operatorname{Tr} S} & \text{if } \operatorname{Tr} S > 0 \\ 0 & \text{if } \operatorname{Tr} S = 0 \end{cases} \tag{4.4}$$

This function has local maxima at corner-like features, so the coordinates of these structures correspond to interest points such as decay vertices, interaction points and track endpoints; an example is shown in figure 4.3. Note that the feature detection service is used to provide the rough spatial location of a feature that warrants further investigation, and should not be used for precision operations such as vertex fitting.

4.5.2 Three-dimensional Feature Detection

Three-dimensional feature detection follows the algorithm outlined below, which uses multiple passes over two-dimensional projections.

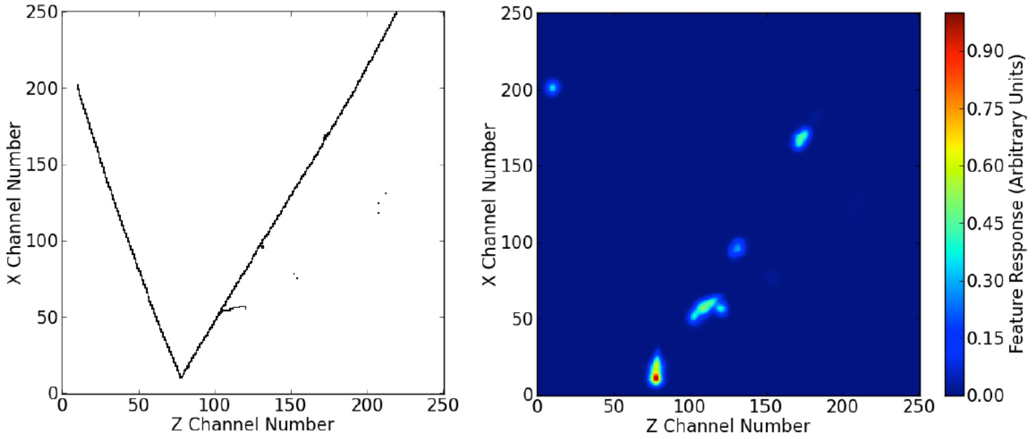


Figure 4.3: A typical neutrino event (left) and the feature response (right) as determined by the feature detection algorithm. The initial vertex and track endpoints are clearly visible, as well as several sites along the right (muon) track where delta electrons were produced.

1. Run 2D feature detection in each of xy, xz and yz projections.

2. Match interest points from each 2D run by requiring a shared coordinate (within some tolerance), e.g. features at $(1, 0)$ in xy and $(1, 3)$ in xz would be matched.

3. For each such matching, make a new 3D feature using the information from both points, e.g. for the features above, create a feature at $(1, 0, 3)$ in xyz.

4. Merge together features in 3D which are within some radial tolerance of each other. This deals with the potential duplicates which may be produced when a feature is visible in all three projections.

5. Return a list of 3D features which survive the previous steps.

Three-dimensional feature detection implemented in this way is imperfect, since we throw away information from one dimension in each projection, only to recombine the results later. A better approach would be to extend the mathematical definition of the structure tensor, S, to 3D (which is trivial) and define a new response function R_N which yields a scalar having large values for corner-like features, and small values elsewhere (which is not trivial).

The extension of the structure tensor to 3D simply involves adding additional spatial derivatives, e.g.

$$S(x, y, z) = g(\sigma_s) * \begin{bmatrix} I_x^2 & I_x I_y & I_x I_z \\ I_x I_y & I_y^2 & I_y I_z \\ I_x I_z & I_y I_z & I_z^2 \end{bmatrix} \tag{4.5}$$

where the I_i are functions of (x, y, z) and all other quantities are identical to those in equation (4.2).

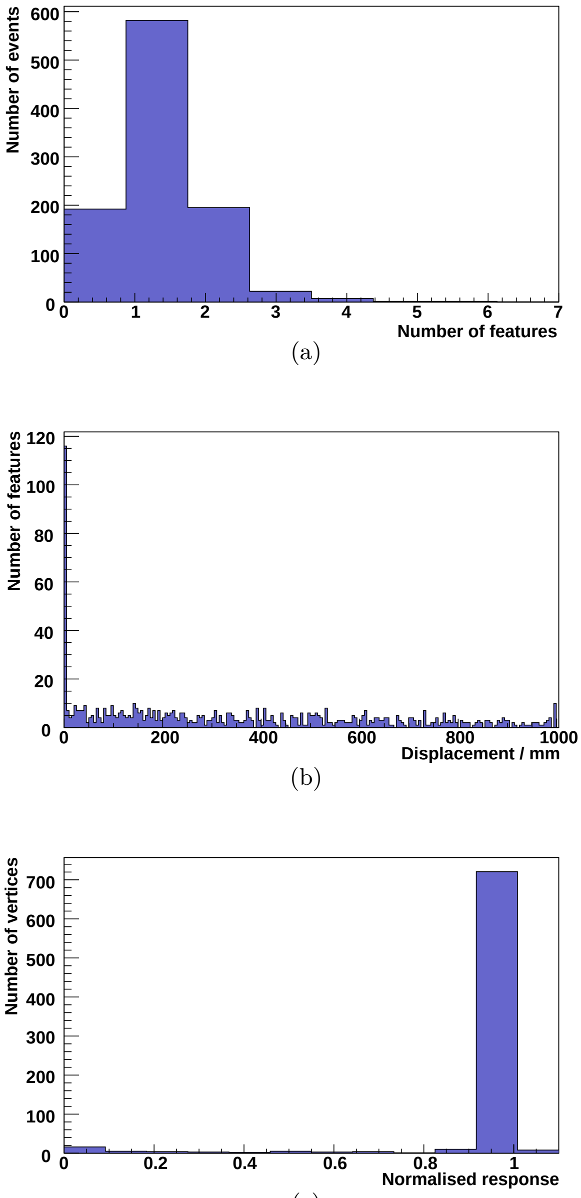
The difficulty lies in constructing a function R to take the place of the Förstner-Noble response function R_N in 2D. Such a function must result in a scalar whose maxima are localised to the features of interest. Extending one of the existing 2D corner-detection functions to 3D would require significant work in the field of computer image recognition, and is beyond the scope of this thesis.

Despite this limitation, the three-dimensional feature response (using two-dimensional projections) performs remarkably well. Figure 4.4 shows three performance metrics for feature detection applied to 1000 CCQE neutrino interactions resulting in a $\mu + p$ final state. The total number of features found in an event is typically small, with between zero and two features found in most events. For the features found, the displacement from the true interaction vertex is typically small, though this has a long tail all the way to 1000 mm, which is the radius of the detector used in this simulation. Finally, the normalised response at the true vertex is typically close to 1, where normalised response is the response at a given feature, divided by the largest response value in the entire event. These characteristics apply even though delta electrons are a large source of noise, having typically high feature response values and appearing throughout the spatial extent of the event.

4.6 Feature Masking

Latte provides a service to *mask out* (remove) hits within a spherical region around a feature. The feature masking service presents a new view of the data, which does not include hits positioned less than some radius away from a central point. Each such point can be the location of a feature as determined by the feature detection algorithms of chapter 4.5, or in a special case the point $(0, 0, 0)$, referred to as *origin masking*.

It is intended that the feature masking service is used after applying feature detection as described in chapter 4.5. Any feature above a threshold value could be used as a masking site to block out a small radius (typically less than 5 mm) around



(a)

(b)

(c)

Figure 4.4: Performance of three-dimensional feature detection applied to CCQE events. (a) A small number of features are found in most events. (b) The distance between a feature found and the true event vertex is typically small (i.e. features are reasonably well located) but may be anywhere up to the radius of the detector, due to the noise introduced by delta electrons. (c) The vertex itself typically provides the strongest response of all features in the event.

that feature. This would effectively punch holes in the event around each vertex and make the task of clustering hits into tracks more straightforward.

This service is useful because it allows algorithms to run on modified views of the event data, which often produces better results. One example would be in a clustering algorithm such as DBSCAN (see 4.4.1), where the local point density affects the clustering. In this case, introducing a gap which corresponds to a feature such as a vertex results in a better clustering of tracks.

4.7 Track Merging

Merging is performed using a cylindrical track road algorithm. Track candidates containing more than 100 hits are treated as key tracks for merging. A straight line in 3D is computed for each key track by the procedure below:

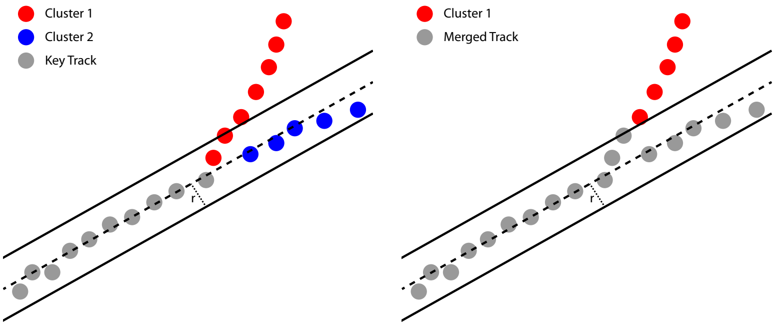
- Determine a unit vector along the line of the track by using principal components analysis and choosing the component with the largest eigenvalue. This becomes the vector $\mathbf{n}$ along the line of the track.
- Determine a point on the line by computing the centroid coordinate of the hits corresponding to the track. This becomes the vector $\mathbf{p}$.

Each track is then represented by an infinite line in 3D, defined by equation 4.6 where the parameter λ is a scalar representing distance along the line from the point $\mathbf{p}$ at which the point $\mathbf{l}$ is found.

$$\mathbf{l} = \mathbf{p} + \lambda \mathbf{n} \tag{4.6}$$

The key tracks are iterated over, beginning with the longest, and hits from other (non-key) tracks are considered for merging into the key track if the distance d from the hit to the line corresponding to the key track is less than some radius r. This defines a cylinder of radius r around the key track, inside which any hits from other tracks are merged into the key track. The cylinder extends indefinitely either end of the key track; in practice this is not a problem due to the sparse nature of events resulting from neutrino interactions. Figure 4.5 illustrates this merging process at a point where a track has been split into two pieces and a third cluster has been identified in the area.

The distance from a hit to the line is defined by taking two points $\mathbf{x}_1, \mathbf{x}_2$ on the line (by picking two values of λ) and, with $\mathbf{x}_0$ as the coordinates of a hit, the



(a) Unmerged tracks (b) Merged tracks

Figure 4.5: The track road cylinder merging algorithm operating on the output of the CA. (a) shows the initial state with a key track and two smaller clusters. The cylinder drawn around the key track encloses all of cluster 2, but only two hits from cluster 1. (b) shows the resulting merged tracks; hits within the cylinder have been merged such that cluster 2 disappears altogether. Cluster 1 remains but has fewer hits than before.

distance is given by equation 4.7 (see also figure 4.6).

$$d = \frac{|(\mathbf{x}_2 - \mathbf{x}_1) \times (\mathbf{x}_1 - \mathbf{x}_0)|}{|\mathbf{x}_2 - \mathbf{x}_1|} \tag{4.7}$$

Hits which are merged into a key track are removed from their original track. Tracks which become empty as a result of this procedure are discarded. This procedure may leave some very small track fragments, which can be cleaned up by applying a range cut requiring any final state track to have some minimum number of hits. Currently, this range cut is set at 20 hits, corresponding to a straight-line track approximately 20 mm long, or a minimum ionising particle with 4.2 MeV kinetic energy.[3]

4.8 Truth Information

When working with simulated events, it is often important to be able to access the *truth information*, i.e. the information which was used to generate those events. The Latte framework provides two simulation packages. The first, *PyTrackGen*, generates events consisting of one or more straight line tracks, and is designed to provide simplified representations of common event topologies. The second, *Lamu*,

[3] A minimum ionising particle deposits 2.1 MeV cm^{-1} in liquid Argon [38].

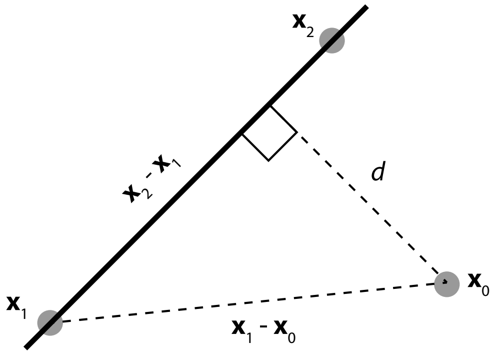


Figure 4.6: Illustration of the perpendicular distance d from the point $\mathbf{x}_0$ to the line between the points $\mathbf{x}_1$ and $\mathbf{x}_2$ according to equation 4.7.

provides a full Geant4 physics simulation, and can be seeded with events from the Genie event generator, or from manually constructed particle sources.

Both kinds of simulation write the truth information corresponding to lines or particles into their output formats, and the Latte framework provides a mechanism to access this information, query it for particle properties, and even make decisions based on it.

It is possible, for instance, to select only hits produced by electrons, or to obtain the true energy of the primary muon from a neutrino interaction. Such quantities are often compared against reconstructed quantities to provide a measure of the success of an algorithm.

4.9 Range Cuts

Latte provides a mechanism to filter event objects based on the number of hits they contain. For track-like objects, this corresponds approximately to the range of the particle. Range cuts can be applied based on the truth information (e.g. to select events where a proton track travelled more than some minimum reconstructible distance), or on reconstructed track objects (e.g. to cut out short track stubs, or to identify long tracks which may correspond to muons).

The range cuts are provided as either filters or wrappers. The filters allow for the selection of events based on track ranges recorded in truth information, while the wrappers present a reduced view of the event, in which the short tracks are not visible.

4.10 Latte Control

Latte Control is a mechanism for applying a sequence of algorithms to a set of events. In its simplest form, it allows the user to set up a *pipeline* composed of service algorithms which are run in sequence on each event stored in a ROOT [50] file.

4.10.1 Event Objects

A Latte *Event* is essentially a Python dictionary[4] with string keys mapped to arbitrary values. The *LamuRun* file reader produces a single key called 'raw', which maps to another dictionary containing keys for accessing the raw, simulated data. An *Event object* in Latte Control is a fairly simple wrapper around this plain dictionary.

The wrapper exists primarily to abstract the idea of an event, so that the bulk of the Latte Control framework can deal with the abstract concept of an *event-like object*. An event-like object is any Python object that provides all of the methods in the `latte.control.event.Event` class. This is useful because it allows us to wrap one event object inside another, thus modifying the behaviour of the event without having to change the original.

The base class, `latte.control.event.Event`, has the following functionality:

- A constructor, used internally by event loops (see chapter 4.10.4). This is not usually called by a user of the framework.
- `getFilename()`, which retrieves the file from which the event was read.
- `getEventID()`, which returns the integer ID of the event within a file.
- `getCurrentHitSelection()`, which returns a list of hits.
- `getCurrentTracks()`, which returns a list of clusters (tracks).
- `getKeys()`, which returns a list of keys defined in the event dictionary.
- `hasKey(key)`, which returns `True` if the key exists, `False` otherwise.

[4] A set of key–value pairs. Some languages call them associative arrays, or maps.

- `setKeyValue(key, value)`, which sets the key `key` to value `value`.
- `addToKeyValue(key, value)`, which adds `value` to key `key`, creating it if it does not already exist.
- `getKeyValue(key)`, which retrieves the value associated with the `key`.
- `getUnderlyingEvent()`, which returns the event-like object that this object wraps (or `self` if it is the innermost layer).
- `hasUnderlyingEvent()`, which returns `True` if this event object wraps another, `False` otherwise.

Most of the methods above allow for access to the keys and values of the dictionary itself. A wrapper class can modify these methods, and by doing so modify the details of how the dictionary is accessed. This allows for great flexibility in defining alternative views of events, for instance, where the set of hits presented differs from the set in the real event (e.g. by application of a range cut, or selection of hits from certain particle types).

In practice, the two most frequently altered methods of the event objects are `getCurrentHitSelection()` and `getCurrentTracks()`. Most algorithms in Latte use these two methods to access the hits or clusters they will work on. In order to provide a modified set of hits or clusters, one simply needs to write an event wrapper to perform the necessary tasks. Event wrappers are discussed in the next section; this is the mechanism by which the range cuts and other modifying services in Latte work, without losing the original event data.

4.10.2 Services & Event Wrappers

Algorithms made available in the Latte framework can be used with Latte Control in one of two ways. For general purpose reconstruction algorithms, Latte Control provides a corresponding *Service*, which accepts an event object and performs some task. For example, the *Merging* service provides an interface to the track merging algorithms provided in Latte and, when called, runs them on the active track selection. Services usually provide algorithms which are processor-intensive and add new information to the event, for example the allocation of hits to clusters, or the merging of multiple clusters based on certain criteria. Services typically augment an event object with the additional information they are able to provide.

In contrast, there are many cases where the goal of an algorithm is to present a different *view* of the same data. One such example is the charge weighting algorithm, which shifts the position of each hit. Such algorithms are typically brought into the Latte Control framework as *event wrappers*, which intercept attempts to access certain data that are already associated with an event, and present a modified version of that data. In this way, the original data is preserved, and can be retrieved by simply *unwrapping* the event. Each event wrapper provides a *wrapping service* which is used to apply the wrapper at an arbitrary point in the pipeline.

4.10.3 Pipelines

Each event is processed by a *pipeline*, which is simply a sequence of service objects that are called in turn. Some services perform complex tasks such as clustering, while others wrap or unwrap an event, thus changing the views of data available for subsequent services. As events propagate through the pipeline, reconstructed information is added to them by the appropriate services, and may be written to disk (for example) at various points.

Since a Pipeline appears (to an outside observer) as just another callable object which accepts events, it is possible to put pipelines within pipelines, for example to group related services into logical *tasks* (such as grouping clustering and merging of tracks into a *tracking* task), or to provide decision making, using a service which chooses to execute one of several pipelines based on the value (or existence) of certain data in the event. A pipeline may contain an unbounded number of tasks, each of which may be another pipeline; there is no limit to the length or complexity of tasks that can be performed with this mechanism.

This flexibility means that the pipeline structure of Latte Control is capable of performing complicated analyses without needing further core support. A user can simply stick together the components they need, in the correct order, then leverage the power of Latte Control to run their analysis job over the available data.

4.10.4 Event Loops

At the highest level, Latte Control provides an event loop which is capable of reading a range of events from a Root file and passing each event in turn through a pipeline of Latte Control services. In this manner, it is possible to easily write reconstruction scripts which configure a number of services, stick them together in an arbitrary order into a pipeline, and have that pipeline applied to each event in a file. This provides a powerful mechanism for writing physics analysis scripts which make use of the Latte-provided services as well as intermediate or final *analysis* services, using the data available in an event to measure physically useful quantities. Since the objects representing pipeline segments persist between events, they can also be used

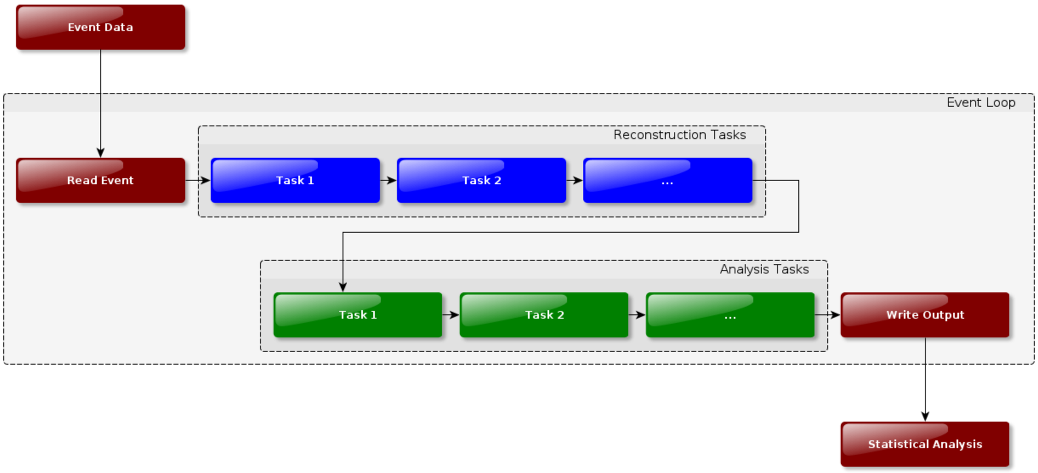


Figure 4.7: Illustration of the passage of event data through a pipeline with an event loop. The data file is loaded by the Latte framework and the event loop iterates over each event in turn. A pipeline is constructed to run sequential reconstruction or analysis tasks. The event object is created at the beginning of the pipeline, and at each step it may be modified or queried as appropriate. The reconstructed data can be written to a separate file, and statistical analyses using information from multiple events can be performed by gathering data using a persistent element in the pipeline, with some code to run after all events have been processed to produce statistical output.

to build up statistics over an entire set of events. When the event loop finishes, the analysis object can be queried for the set of statistical data it has collected. Figure 4.7 illustrates the passage of an event through a pipeline.

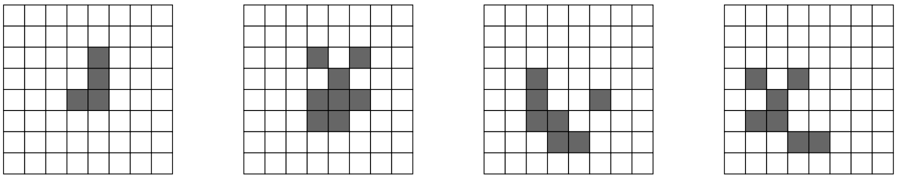
5 A Cellular Automaton for Track Reconstruction

5.1 Cellular Automata

A Cellular Automaton (CA) is a Finite State Machine (FSM) which usually consists of a regular grid of cells in a finite number of dimensions, each of which exists in one of a finite number of states. A neighbourhood is defined to be some region around any given cell, and the next generation is formed by updating the state of each cell according to a fixed rule which takes into account the current cell state as well as the state of its neighbours. Typically, this rule is identical for each cell and at each time step. The update rule is usually applied simultaneously to the whole grid.

The concept was originally devised by John von Neumann [69] in the 1940s as a method of creating self-reproducing machines using a CA with 29 states. Cellular automata became popular in the 1970s with the development of the *Game of Life* by John Conway (see, for example, [70]). In *Life*, cells exist in one of two states (live or dead) and the neighbourhood is defined to be the 8 cells surrounding a central cell on a square grid. The update rules are simple but result in varied and complex behaviour. Figure 5.1 illustrates these rules for the first four states of a simple initial pattern.

1. Any live cell with < 2 live neighbours dies (under-population).
2. Any live cell with 2 or 3 live neighbours lives on.
3. Any live cell with > 3 live neighbours dies (over-population).
4. Any dead cell with exactly 2 live neighbours becomes live (reproduction).



(a) Step 1 (b) Step 2 (c) Step 3 (d) Step 4

Figure 5.1: The first four states of a CA following the rules of *Conway's Game of Life*. Grey squares represent live cells, white squares represent dead cells. The initial pattern in (a) produces increasingly complex states, e.g. (b) to (d), as the system evolves according to a few simple rules. The state of a cell depends only on the states of its immediate neighbours in the preceding step, allowing the state to be computed in parallel, using only local information.

5.2 Cellular Automata & Track Finding

The use of a CA for tracking in high-energy physics was proposed in [71] for filtering tracks in Multi-Wire Proportional Chamber (MWPC) detectors. They define a cluster of continuous hits to be a living cell and an empty region to be a dead cell. In order to support interrupted tracks due to detector inefficiencies, they also propose phantom cells, which may also be either living or dead, giving their CA four states.

Neighbours are determined by the range of admissible angles at which a track may run through the cluster of hits corresponding to any given cell. The update rules create phantoms if there are living neighbours on either side which are compatible with the track model, and destroy cells if there are too few (< 2) or too many (> 4) neighbours. The first part is designed to cope with gaps in a track due to detector inefficiencies, while the last part filters out noise hits. The CA is iterated until a stable configuration is reached, at which point the remaining real live cells correspond to the hits of filtered tracks. A similar procedure is also proposed in [72].

An important development was made in [73] where it was proposed to make a CA in which the cells were straight line segments linking hits or clusters in adjacent layers of a detector (or skipping at most one layer, to account for detector inefficiencies). The neighbours of a cell are considered to be those cells which share a hit or cluster at one end and have some angle $\phi < \phi_{\text{max}}$ between the line segments.

An integer number is assigned as the state of each cell, associated with the cell's position along the track. Initially, all cells get the state 1 and at each step of evolution the update rule looks at neighbours in preceding layers and increases the position value by 1 if there is a neighbouring cell with the same state. Evolution stops when no neighbouring cells have the same state. As usual, time evolves in discrete steps with cell updates occurring simultaneously.

After this CA is run the final state gives the positions of all segments of all track candidates. The track candidates are formed by starting with the highest valued segments and adding the neighbour with the previous position value, all the way back to position value 1. This approach not only filters out noise but also *clusters* the track candidates into distinct clusters.

A further example of the use of line segments for the cells of a CA with similar update rules, considering neighbours to be leftward segments sharing a common space point and having some angle no greater than a maximum *breaking angle* between adjacent segments is considered for the HERA-B vertex detector in [74]. The approach is extended to 3D tracks in a layered scintillator bar detector with horizontal and vertical layers in [75], where the CA is run in 2D over the xz-plane and the yz-plane independently, and tracks are combined into a three-dimensional reconstruction using z positions and timing information.

5.3 A 3D Cellular Automaton for Track Finding

The Cellular Automaton for Reconstruction in Liquid Argon (CARLA) algorithm is composed of several procedures, each of which is described in detail. The algorithm runs through each of the following stages in turn:

1. Preprocessing
2. Cell generation
3. Forward run
4. Reverse run
5. Postprocessing

The pre- and post-processing stages are specific to the form of data used; preprocessing typically includes a charge weighting process followed by a re-scaling to reduce the number of input hits, while postprocessing typically involves merging track segments and filtering out short fragments. In the following algorithm descriptions, the term *leftward* means having lower coordinate value along the beam axis, which is defined to be the x-axis in the simulation. Similarly, *rightward* means having higher coordinate value along the beam axis.

5.3.1 Terminology

The following description of the cellular automaton reconstruction algorithm makes extensive use of the terms *forward*, *backward*, *left* and *right*. For the most part, these terms originated in the papers describing earlier implementations of cellular automata for tracking, described earlier in this chapter. Precise definitions in the context of the algorithm used here are given below. Note that the algorithm presented is, to the maximum extent possible, fully three-dimensionally aware and that, in particular, the concept of left or right is relative, even though it is required in order to avoid ambiguity about how to proceed in any given step. The algorithm cannot be described without defining the terms, but the terms cannot be properly defined without describing the algorithm; the definitions here should be taken together with the complete description that follows.

Forward The *forward* run is the process by which the cellular automaton gives each cell a weight value, updating it until a stable state is reached (i.e. no weights change from one pass to the next). This process occurs across the entire 3D space, and may even be performed in parallel.

Reverse, or Backward The *reverse* run is the process by which the cellular automaton follows a chain of cells whose weights decrease from an arbitrary integer to 1, descending by 1 weight unit at a time. This process can occur in any direction in 3D space; the *reverse* part refers to traversing a chain of cells from a high number to a low number.

Left and Right The term *left*, or *leftward*, describes the relative position of one hit or cell when compared with another hit or cell. The positions are taken relative to a single direction vector in 3D, which is usually chosen to be along one of the axes of the detector. A typical choice is for this vector to be along the beam, with *left* upstream and *right* downstream, but the algorithm does not require this to be the case. All that is required is a relative sense of direction, consistent across the entire detector, but applied locally when comparing nearby points or cells.

5.3.2 Preprocessing

Charge Weighting

A charge weighted smoothing procedure is applied to the raw hits before further processing occurs. This procedure is described in chapter 4.3.

Scaling

The run time of the CA depends strongly on the hit multiplicity (which in turn determines the cell multiplicity). In order to reduce the run time, a re-scaling from 1 mm to 3 mm voxels is implemented after the charge weighting procedure. This re-scaling groups together hits within the same 3 mm × 3 mm × 3 mm region and represents them as a single cluster at the charge-weighted centroid position of those hits (calculated as for the charge weighting, with equation 4.1), but now containing the accumulated charge of all the hits within it.

5.3.3 Cell Generation

Cell generation is the process by which cells are made from pairs of hits. A cell consists of a leftward and rightward hit pair, and represents the line between the two hits, pointing from the leftward hit to the rightward hit. In principle, each possible pairing of points within a fixed radius could produce a cell, but for speed and performance reasons it is better to build the cells more selectively. The algorithm below does this in a manner which produces the required cells with very little overhead.

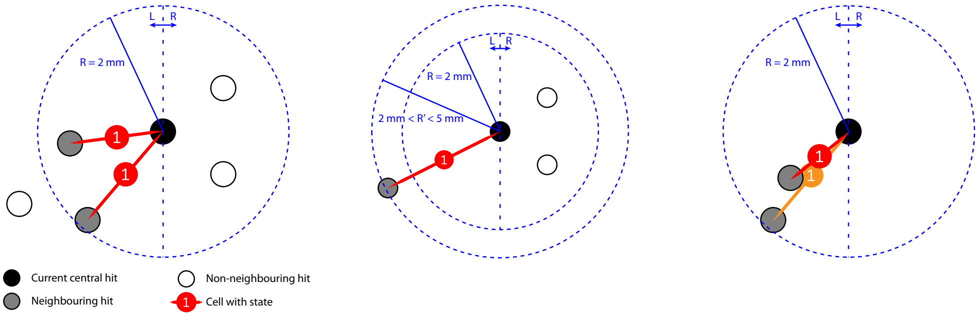
For each hit in the event:

1. Build a list of neighbouring hits within a 2 mm radius; in this implementation a KDTree is used for this near-neighbour search.

2. Filter the neighbour list to select only those neighbours which are *upstream* with respect to the beam direction, i.e. select only leftward neighbours.

3. Generate a cell for each remaining leftward neighbour, pointing from that hit to the current central hit (from left to right); see figure 5.2(a).

4. If no leftward hits were found, i.e. no cells were generated in the previous step, expand the search radius by 0.05 mm and repeat the procedure above until either a cell is generated or the maximum search radius of 5.0 mm is reached; see figure 5.2(b).



(a) Normal operation (b) Expanding radius (c) Filtering

Figure 5.2: Illustration of the steps in the cell generation procedure for the CA: (a) creation of cells between a central point and any leftward neighbours within a 2 mm radius. (b) adaptive search radius expansion, which gradually expands the radius in 0.05 mm increments until either a leftward neighbour is found or a maximum radius of 5 mm is reached. (c) removal of cells if a shorter cell exists with similar slope; in this case, the red cell is retained while the longer orange cell is removed.

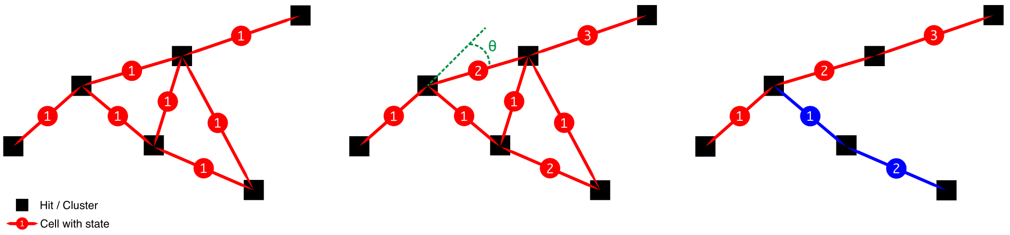
5. If multiple cells were generated, filter out longer copies of cells with the same slope, using a comparison tolerance of 0.10 on the slope components.[1] This step avoids making long cells which "jump" over several shorter cells, which could lead to multiple interleaved tracks in the final reconstruction; see figure 5.2(c).

5.3.4 Forward Run

The forward run is responsible for the evolution of the CA from its initial state until a stable final state is reached. The forward step algorithm is run until the state remains unchanged, that is, until no cells needed to be updated in a step.

Before the first forward step, a mapping from cell to list of leftward neigbours is built. Cell A is a leftward neighbour of cell B if they share a central point (cell A's right point is identical to cell B's left point) and if the angle θ made between the two cell vectors is less than some threshold value θ_{max}. This mapping is used in every forward step, providing a speed boost over searching for leftward neighbours

[1] The value 0.10 is chosen geometrically. Consider a 3×3 grid of unit cells; a line running maximally through the diagonal has slope $\alpha = \tan^{-1}(3/3) = 0.98$, while a line which just touches the top right cell has slope $\alpha = \tan^{-1}(2/3) = 0.78$, a difference of 0.20. The discrete, voxellised nature of the detector means that it is very difficult to distinguish between two lines with slopes less than half of this value because they hit almost identical voxels, and charge weighting smooths out the small differences that do exist. A value of 0.10 is therefore used as the comparison tolerance for slopes.



(a) Initial state (b) Final state (c) Output clustering

Figure 5.3: Illustration of the effects of the forward run of a CA on a set of cells: (a) initial state with all cell values set to 1. (b) final state where cell values have been updated as per the forward step algorithm. Cell values now reflect the position of a cell in a straight line track. Cells which have no leftward neighbours consistent with the breaking angle θ retain the value 1. (c) track clusters are formed by following chains from high-valued to low-valued cells, while cells which are not part of a track are filtered out.

every time, since the number of cells does not change, and the cell properties are fixed (with the exception of cell state).

The forward step algorithm proceeds as follows (see also the illustration in figure 5.3):

1. For each cell with one or more leftward neighbours in the same state, mark the cell for update.

2. Update all marked cells by increasing the value of their state by 1.

3. Return the number of updated cells (if 0, this signals termination of the forward run).

It is important not to update any cell state before all cells have been tested to see whether they must be updated. This prevents problems where a cell which would have been updated (because it had a neighbour with the same state) does not get updated because its neighbour was updated first. The forward step must appear to occur simultaneously across all cells; this opens up opportunities for parallelisation since each cell can be tested and marked for update independently. However, since this part of the algorithm is relatively quick for small numbers of cells, the current implementation does not do this in parallel.

5.3.5 Reverse Run

The reverse run extracts track clustering information from the final state of the CA. The mapping of cells to a list of their leftward neighbours is retained, again for efficiency. Initially, all cells are considered as input to the reverse run. The reverse step algorithm is performed until no cells remain in the input list.

The reverse step algorithm proceeds as follows:

1. Find the input cell with the highest valued state. Set as current cell.
2. Create a new track candidate and add the current cell to it.
3. While the current cell has value > 1:
   a) Find a leftward neighbouring cell with state value 1 less than the current cell state.
   b) If multiple candidate cells exist, pick the one which makes the smallest angle with the current cell.
   c) Add this cell to the track candidate, and mark it as the new current cell.
4. Remove allocated cells from the input list.

Each reverse step follows a sequence of cells from a high-valued cell state to a cell with state 1, building a complete track candidate in the process. The reverse step algorithm runs until all track candidates have been extracted. Track candidates which contain only 1 cell are rejected as noise. The result of applying the reverse run algorithm is illustrated in figure 5.3(c). Finally, the cells in each track candidate are unpacked to leave each track as a list of hits.

5.3.6 Postprocessing

The CA algorithm is considered to be complete at this point. Postprocessing stages are used to improve the results from the CA based on its known properties.

The first stage involves uniquely assigning hits to tracks. The CA makes unique assignments of cells to tracks, but since a hit can appear as an endpoint of multiple cells, it is possible that hits may appear multiple times across multiple tracks. The current strategy for unique hit assignment is to keep each hit with the longest track it is a member of, and remove it from any shorter tracks.

The second stage of postprocessing merges track segments together if they are compatible with being part of the same straight line. This is necessary because the CA tends to break tracks up at any site of complexity, including the primary vertex, decay vertices and delta electron production sites. In particular for μ tracks, it is useful to stitch together these segments to produce a single longer track. The merging algorithm is described in detail in chapter 4.7

5.3.7 Optimisation of CA Parameters

The algorithm presented is a general purpose reconstruction procedure for track-like objects in fine-grained detectors. In order to apply it to the data from a particular experiment or simulation, the parameters must be optimised somewhat to the environment in which it will be operating. One such optimisation is described in [2], in which the parameters are tuned to maximise the product of hit-level efficiency times cluster-level purity. Hit-level efficiency is defined to be the number of hits visible in the CA output as a fraction of the total number of hits input, and is a measure of how well the CA retains data. Cluster-level purity is a measue of the contamination of each cluster, i.e. what proportion of the hits in a cluster originate from the same real track, i.e. a measure of the clustering capabilities of the algorithm. By maximising this product it was found that a wide range of CA parameters provide comparable results. Further optimisation then depends on the desired output from the CA, as well as experimental considerations such as the physics goals and operating energies.

Another optimisation is presented in chapter 5.5.1, and in particular table 5.2. Here, the number of output tracks after the application of a merging procedure is used as a measure of the ability of the CA to produce the kind of clustering expected for the input data, which consists of two-track events with long tracks. Any use of the CA in a reconstruction chain requires optimisation of the parameters for the desired clustering, taking into account detector and experimental properties.

5.4 Performance of the Cellular Automaton on Toy Monte Carlo Events

The CA was initially tested on a set of straight line tracks from a toy Monte Carlo (MC) generator which deposits charge from straight line segments in voxels of 1 mm × 1 mm × 1 mm. The tracks are generated such that they have lengths between 200 mm and 300 mm. Both tracks begin at point $(0, 0, 0)$, and the opening angle between the two tracks is fixed to some θ. The events are then rotated so that the lines are distributed isotropically, but retain the fixed opening angle and start point. A thousand events were generated at each angle θ from 2° to 178° in

4° intervals.

The events were first charge weighted with a radius of 6.0 mm, then scaled with a scale size of 3.0 mm. The resulting hits were run through the cell generation process with a default cell generation radius of 6.0 mm, maximum radius 15.0 mm (in unscaled units). The CA algorithm itself was applied to each event in turn. The breaking angle was set to $\theta_{\max} = 10°$ and the resulting tracks were postprocessed with the track road merging algorithm with a cylinder radius of 5.0 mm.[2]

5.4.1 Raw Cellular Automaton Output

θ (°)	R_W (mm)	R_M (mm)	Min. Range (hits)
10	6	5	20

Table 5.1: Cellular Automaton parameters used for the reconstruction of toy tracks.

The number of tracks found by the CA as a function of angle is presented in figure 5.4. With 1000 events at each angle, and an input sample of events containing only two tracks, the number of interest is the efficiency for finding two tracks; that is, the number of events in which two clusters were found divided by the total number of events.

While the raw performance of the CA, characterised by the efficiency for finding two tracks, is poor, the resultant clustering typically provides long track candidates which can be used as key tracks in the track road merging process. Figure 5.5 shows the raw clusters output by the CA for one such event. Although there are many clusters, one track was successfully identified in its entirety, while the other is split into several shorter segments. Each segment is long enough to act as a seed point from which almost all of the segments may be merged together to recover the other long track. As such, the CA provides a significant advantage over applying a combinatoric approach using only track road algorithms, which is impractical given the large numbers of hits typically associated with a neutrino event in a LAr TPC. The segmented track in this event is due to the CA being unable to cope with geometric effects introduced by the voxellised nature of the data. These effects tend to produce runs of linear structures, separated by a jump or step that the CA interprets to be a large angle between cells; one of the criteria for breaking a track in two. This is a natural outcome from the CA and is the desired effect when dealing with vertices and interaction points, so it cannot be “tuned out”. The merging algorithm discussed below aims to compensate for these cases in which the

[2] These parameters were chosen in order to maximise the number of correctly reconstructed events.

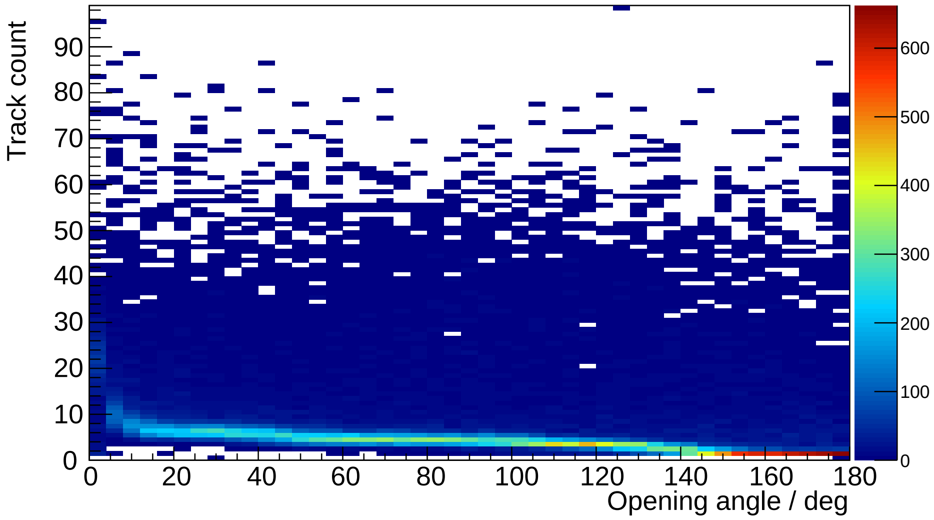


Figure 5.4: Number of tracks found as a function of opening angle for the raw CA operating on two track events with a fixed opening angle. There are 1000 events per opening angle, and the events are rotated so as to be distributed isotropically. Large numbers of tracks are found in some cases due to geometric effects in the way the CA operates. These can be merged back together with an appropriate merging algorithm.

track structure remains sufficiently linear to suggest that the desired outcome is one cluster, not two. Figure 5.6 shows one scenario in which the CA has produced several clusters from a single straight line due to the voxellised nature of the data. Here, the CA has found several straight line structures which must be merged to form a single line.

5.4.2 Performance of the Cellular Automaton with Merging

When the merging process is applied, the track counts (figure 5.7) are considerably improved. The merging process tidies the events up, gathering together any track fragments and recombining them with the track they should belong to. In some cases, small track fragments remain, and while the modal number of tracks after merging is 2, there is a long tail caused by a handful of events with significantly more tracks after merging. A simple range cut can be imposed to remove these remaining fragments.

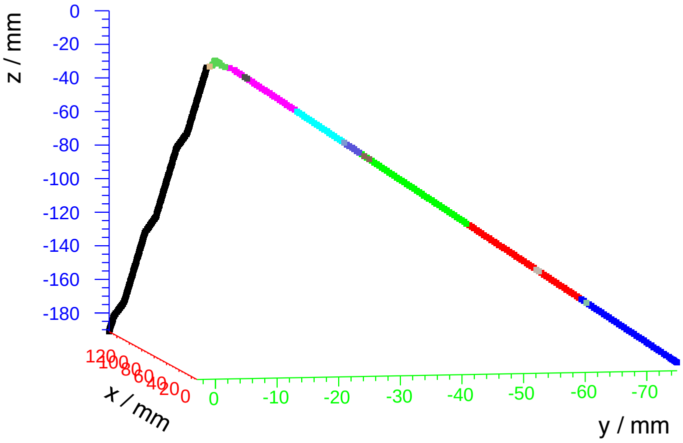


Figure 5.5: Sample event from the toy track generator, with an opening angle of 42° between the two tracks. The cellular automaton clusters one track successfully, but produces several clusters (each coloured segment represents a cluster) along the length of the other due to the geometric effects of the voxellised data. It is important to note that each such cluster forms a segment along the line which is easy to merge together with the other segments, and that the segments include no hits that do not belong (i.e. hits from the other line).

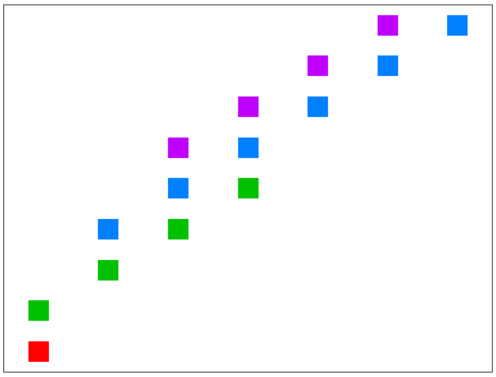


Figure 5.6: Close-up of a track reconstructed by the CA algorithm, showing multiple straight line segments, each represented by a different colour. This is an side effect of the voxellised data format. The view shown is a zoomed region of the xz projection of a line and each hit is positioned on a 3 mm square grid.

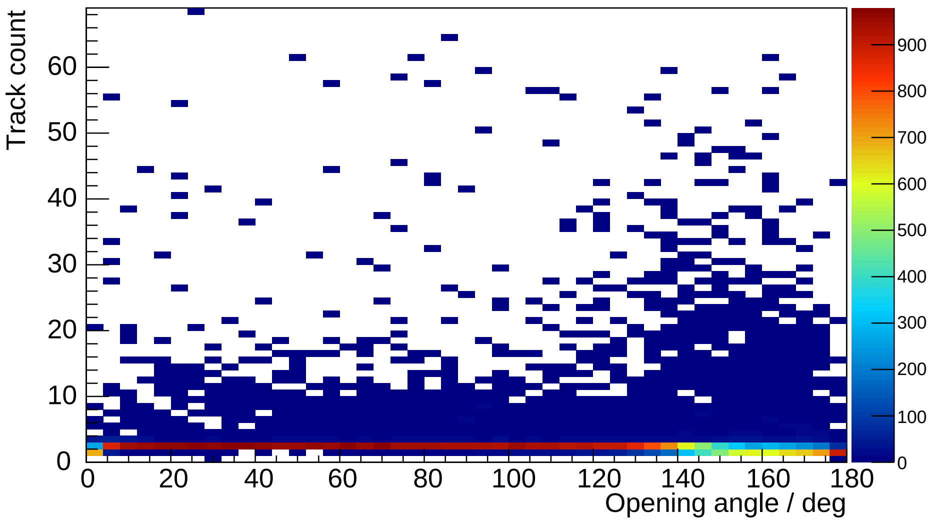


Figure 5.7: Number of tracks found as a function of opening angle for the CA with track road merging operating on two track events with a fixed opening angle. There are 1000 events per opening angle, and the events are rotated so as to be distributed isotropically. A small track fragment is usually left over, which can be removed with a suitable range cut.

5.4.3 Performance After a Range Cut

After imposing a range cut of 20 hits (corresponding roughly to 20 mm or 4.2 MeV deposited by a minimum-ionising particle) the performance improves further. Now, the mean number of tracks (see figure 5.8) is 2, giving a high two track reconstruction efficiency (figure 5.9) for most opening angles. Above an opening angle of about 140° the CA fails to cluster the two lines separately since the change in angle from one line to the next is now small and the CA can smoothly travel across the join, resulting in a single output cluster. This is not symmetric with the case at an opening angle of 40°, where the angle between the tracks is much sharper, and the CA can easily distinguish the two. Efficiency curves are shown in figure 5.9 at four different purity levels, defined on a hit level as the fraction of hits in a cluster which came from the same truth track. At 80% and 85% purity, the CA performs with a high efficiency. This is reduced slightly for a requirement of 90% purity, and substantially for a requirement of 95% purity. The CA can therefore be said to cluster simple topologies with very high efficiency and with reasonable purity, up to geometric opening angles of 140°.

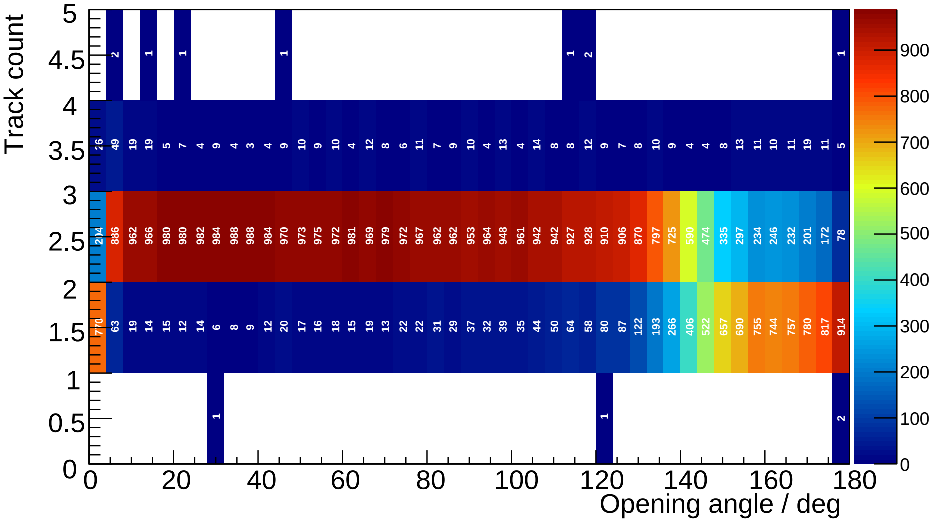


Figure 5.8: Number of tracks found as a function of opening angle for the CA with track road merging operating and a range cut of 20 mm on two track events with a fixed opening angle. There are 1000 events per opening angle, and the events are rotated so as to be distributed isotropically. The mean number of tracks found is now 2, dropping to 1 above 140° due to an inability of the CA to distinguish this topology from that of a single continuous line.

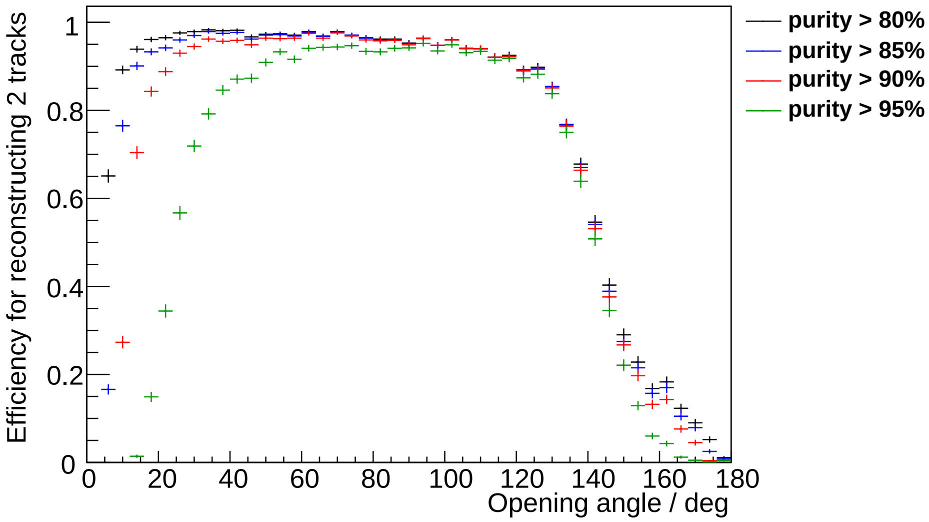


Figure 5.9: Efficiency for finding two tracks in the CA with track road merging and a range cut of 20 mm operating on toy MC events with fixed opening angle, as a function of opening angle. Efficiency curves are plotted for four different purity cuts, requiring a minimum of 80% (black), 85% (blue), 90% (red) and 95% (green) purity in each track. The efficiency is high for almost all angles up to 140°, beyond which the CA is unable to distinguish the topology from that of a single continuous track.

5.5 Performance of the Cellular Automaton on Genie Monte Carlo Events

The data for this section consists of neutrino events generated using Genie and tracked through a detector using Geant4, as described in chapters 3.7 and 3.5 respectively. The reconstruction algorithm is applied first to charged current ν_μ interactions producing $\mu + p$ final states (often referred to as Charged Current Quasi-Elastic events), and then to charged current ν_μ interactions producing $\mu + p + \pi^+$ final states.

5.5.1 Charged Current $\nu_\mu \rightarrow \mu + p$ (CCQE)

The Genie event generator was used to model the interactions of 0.77 GeV muon neutrinos on Argon nuclei. A sample was obtained of 1000 charged current interactions resulting in $\mu + p$ (only) final states. The resulting events were tracked through a liquid Argon TPC using Geant4 and the voxel data stored for analysis. Interactions of this type are of particular interest because they account for approximately 66% of the interaction cross section at low energy and the two-body final state allows an accurate estimate of the muon neutrino energy to be made from the muon kinematics alone.

This data sample raises the possibility of attempting to optimise the CA parameters for reconstructing neutrino events. Since it is a full simulation, the events will contain noise arising from delta electrons, hadronic reinteraction of the proton, muon decay and other physics processes. The truth information recorded for each event allows for the selection of a subset of this information, for example just the hits corresponding to energy deposited by the primary particles. In this way it is possible to optimise the algorithms based on more realistic data than in the toy study, but without including obviously difficult events.

Long Two-Track Events

In the first instance, events were selected in which both the proton and muon contained over 200 hits (corresponding approximately to a 20 cm straight line) and the hits from other particles were stripped from the event prior to running the CA. The resulting events therefore contain two long tracks and nothing else. In a small number of cases, the event topology may make it difficult for the CA to reconstruct these events correctly, e.g. if the muon and proton were produced almost back-to-back, or if either particle undergoes a large scatter along its trajectory, but for the most part these events should be reconstructed with two clusters in the CA output.

A total of 483 events out of the sample of 1000 contain muon and proton tracks with over 200 hits each. Three parameters available in the reconstruction algorithms were varied and these 483 events processed with each permutation. The parameters are the CA track angle θ (10°, 20° and 30°), the charge weighting radius R_w (2.0 mm, 5.0 mm and 10.0 mm) and the track merging radius R_m (10 mm, 20 mm and 30 mm). The results of this optimisation are shown in table 5.2, which lists the number of events reconstructed with one, two or three tracks against the reconstruction parameters used. The results are sorted by the number of events reconstructed with two tracks. The largest number of two-track events are returned from the algorithm when used with parameters $\theta = 10.0^\circ$, $R_w = 5.0\,\mathrm{mm}$ and $R_m = 30.0\,\mathrm{mm}$. These parameters yield 345 two-track events, corresponding to 71.4% of the input sample. In addition, these parameters give a low number of one- and three-track events.

While the number of tracks found in an event is one metric of the quality of the reconstruction, it is also important to consider *hit efficiencies* and *track purities.*

Hit efficiency: A measure of how much of the original event data is retained. Hits that are present in any output cluster, divided by the number of input hits. High hit efficiency means that the algorithm is not throwing away many hits, which makes subsequent energy calculations easier. Due to the nature of the CA, the algorithm will necessarily throw away some hits but the track structures should remain and lost hits can be retrieved with track-road style merging applied to the original data, rather than to a cluster, if required.

Track purity: A measure of how selective the clustering is. An output cluster will contain hits from one or more truth tracks. The track purity is the largest number of contributing hits from a single track, divided by the total number of hits. High purity means that the CA clustered mostly hits from a single trajectory together. This is extremely important, though the purity is often diluted by hits from e.g. delta electrons produced along the length of a muon track, but clustered with it.

Figure 5.10 shows the hit efficiency and figure 5.11 track purity results from the analysis performed with $\theta = 10^\circ$, $R_w = 5.0\,\mathrm{mm}$ and $R_m = 30.0\,\mathrm{mm}$. The efficiency drops slightly with track length, most likely due to increased scattering as the particle that produced the track loses energy and slows down. The CA will struggle with situations where the scattering angles exceed the track breaking angle θ, and will produce many short track segments, each of which will be filtered out

θ (°)	R_w (mm)	R_m (mm)	N_1	N_2	N_3
10.0	5.0	30.0	59	345	72
10.0	5.0	20.0	43	341	84
10.0	10.0	30.0	91	335	53
10.0	5.0	10.0	35	331	70
10.0	10.0	20.0	76	331	66
10.0	10.0	10.0	67	327	62
20.0	5.0	30.0	109	305	62
20.0	5.0	20.0	94	305	70
20.0	2.0	30.0	64	302	109
20.0	5.0	10.0	85	301	55
20.0	2.0	20.0	49	301	114
20.0	2.0	10.0	38	293	98
30.0	2.0	30.0	101	288	87
30.0	2.0	20.0	88	288	89
30.0	2.0	10.0	79	279	77
20.0	10.0	30.0	196	237	46
20.0	10.0	20.0	189	231	53
20.0	10.0	10.0	188	217	56
30.0	5.0	30.0	206	210	59
30.0	5.0	20.0	198	204	67
30.0	5.0	10.0	195	195	55
10.0	2.0	30.0	28	170	246
30.0	10.0	30.0	276	163	40
30.0	10.0	20.0	271	157	45
30.0	10.0	10.0	271	143	50
10.0	2.0	20.0	16	97	285
10.0	2.0	10.0	4	56	189

Table 5.2: Optimisation of reconstruction parameters in the CA for reduced CCQE events (see text for details). Data is shown for the number of events reconstructed as containing one, two or three tracks (N_1, N_2 and N_3) for different combinations of the track opening angle θ, the charge weighting radius R_w and the track merging radius R_m. Entries are sorted by the number of events reconstructed with two tracks (N_2), largest first. Each set of parameters was used to process the same 483 events. Events with more than three reconstructed tracks are not shown.

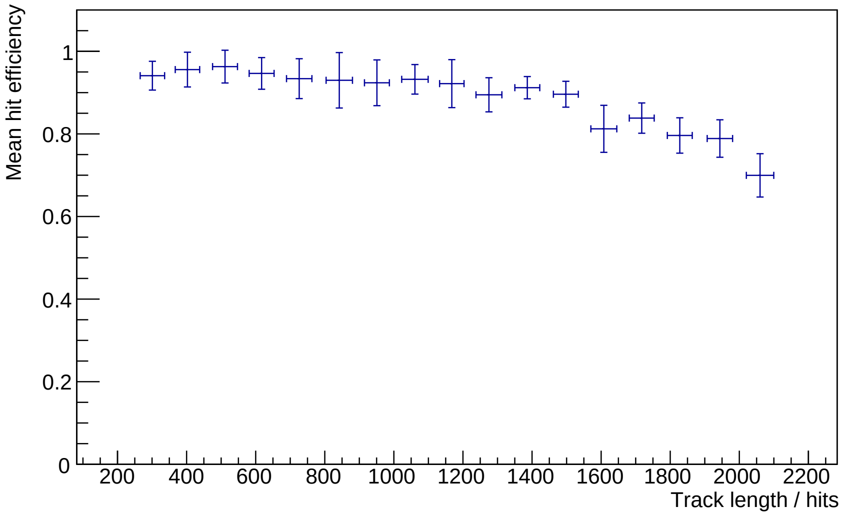


Figure 5.10: Hit efficiency for the CA operating on long muon and proton tracks from CCQE events. The hit efficiency is high, but drops as a function of track length, indicating that the CA is not capable of retaining all hits for long tracks, most likely due to increased scattering.

in subsequent stages unless they can be merged together. The track purity remains over 90% for all track lengths, indicating that the CA with these parameters is capable of differentiating extremely well between unrelated trajectories.

Full Reconstruction

θ (°)	R_W (mm)	R_M (mm)	Min. Range (hits)
10	5	30	20

Table 5.3: Cellular Automaton parameters used for the reconstruction of CCQE ν_μ events.

Having attempted to optimise the parameters on a reduced dataset where two reconstructed tracks are expected, the performance of the CA can be assessed when run against the full simulated events. Here, a much smaller range cut of 20 hits is applied. Events containing proton tracks with fewer than 20 hits are not processed, and in the remaining events all objects with fewer than 20 hits are cut out of the data before reconstruction. A 20 hit track corresponds approximately to a 20 mm trajectory, or an energy deposit of 4.2 MeV, based on a minimum-ionising particle depositing 2.1 MeV cm^{-1}. Such values can be considered as realistic energy

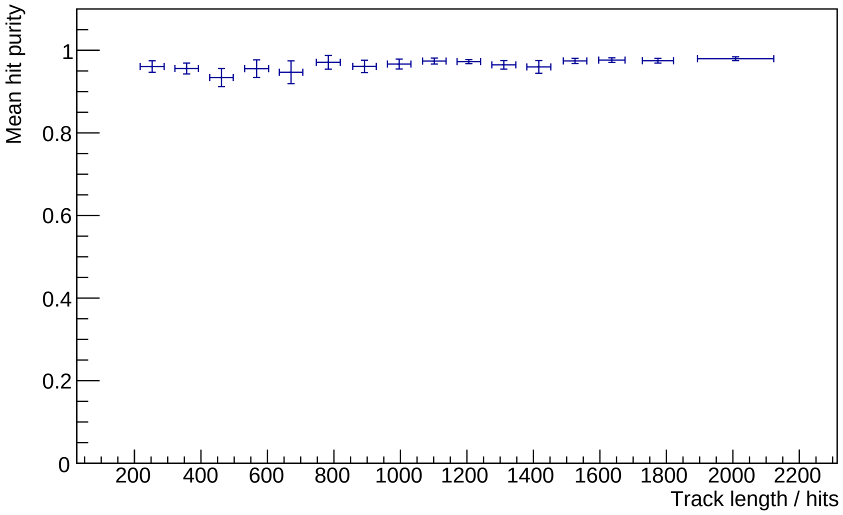


Figure 5.11: Track purity for the CA operating on long muon and proton tracks from CCQE events. The purity is over 90% for all track lengths, indicating that the CA is capable of separating hits resulting from different particles extremely well.

thresholds in any liquid Argon TPC, and the presence of tracks in the simulated data with such short range adds noise that would simply not be seen in a real detector. Following the 20 hit range requirement on protons, 878 events of the 1000 event sample survive for reconstruction.

Because of this relaxation of the initial range cuts, and because the full event data is now included, the results are expected to be poorer than those for the case presented above. In particular, the noise introduced by delta electrons will serve to pull the CA off course as it follows long tracks, and the decay products may themselves contribute additional tracks which can be reconstructed. We should not, therefore, expect the mean number of reconstructed tracks to be two.

Figure 5.13 shows the distribution of reconstructed track counts. Of the 878 events, 420 (47.8%) had two reconstructed tracks, while a further 270 (30%) had three reconstructed tracks and a further 86 events (9.7%) had four or more reconstructed tracks. The maximum number of reconstructed tracks was six, which occurred for only three events. Given the full set of data included, these numbers are within expectation, i.e. almost 80% of the events have two or three tracks, corresponding most likely to the proton, muon and either a delta or Michel electron. Those events with four or more reconstructed tracks bring the total to almost 90%, leaving 102 events (11.6%) reconstructed with just one track.

The relatively large number of events reconstructed with just one track can be put down to a number of geometric effects in the interaction topologies, including short proton tracks running along almost the same trajectory as the muon, or proton tracks at large angles to the muon. It was already demonstrated in chapter 5.4 that the CA cannot cope with extremely large opening angles (greater than about 140°). Figure 5.12 shows the output from the CA for three events from this sample, with clusters indicated by colour.

Figure 5.14 shows the hit efficiency (as defined above) for the full CCQE reconstruction. Here, the drop in efficiency with increasing track length is still present, though the effect is reduced; probably because the additional hits from delta electrons help to smooth out the large scattering angles sometimes present, and allow the CA to run through them. Figure 5.15 demonstrates that the CA produces extremely pure tracks (over 90% pure) even in situations where hits from delta electrons, decay products and other sources of noise are present. Since one of the main goals of clustering is to obtain collections of related hits, a high purity is very important.

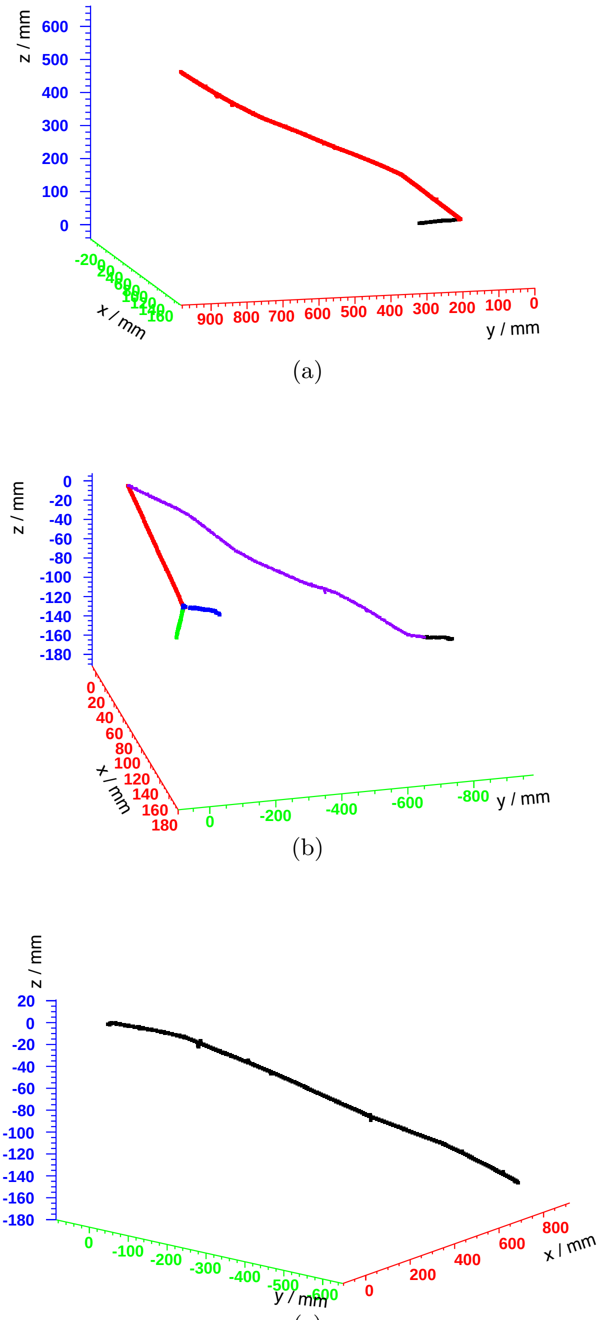
5.5.2 Charged Current $\nu_\mu \rightarrow \mu + p + \pi^+$ (CC1π)

θ (°)	R_W (mm)	R_M (mm)	Min. Range (hits)
10	5	30	20

Table 5.4: Cellular Automaton parameters used for the reconstruction of CC1π ν_μ events.

A small fraction of low energy neutrino charged current interactions will produce a charged pion in the final state, mostly through the production and subsequent decay of nucleon resonances. These events were generated with Genie, taking all charged current interactions at the required energy of 0.77 GeV and selecting those with a $\mu + p + \pi^+$ (only) final state. The reconstruction algorithm parameters were identical to those applied to the CCQE events, above, i.e. $\theta = 10°$, $R_w = 5.0\,\text{mm}$ and $R_m = 30.0\,\text{mm}$. Once again, 1000 events were generated, and a range cut of 20 hits imposed on both the proton and pion tracks, leaving 891 events for reconstruction.

In principle, the task of reconstruction is more challenging for these events due to the additional final state particle when compared to the CCQE events. In practice, the CA handles these events with little degradation of performance. Figure 5.17 shows the distribution of number of tracks reconstructed. More tracks are found, on average, than in the CCQE events, which is to be expected since we must



(a)

(b)

(c)

Figure 5.12: Clusters found by the CA in ν_μ charged current interactions resulting in $\mu + p$ final states. (a) μ (red) and proton (black) tracks correctly clustered. (b) μ track (purple) and Michel electron (black), proton (red) and products of hadronic reinteraction (blue, green). (c) an event in which the μ and proton were produced with nearly identical trajectories and the CA was only able to find one track. Axes are labelled in mm.

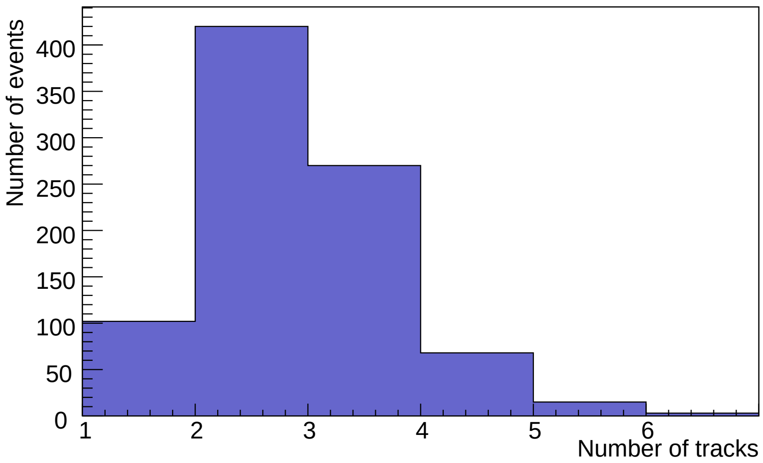


Figure 5.13: Distribution of reconstructed track counts for CCQE ν_μ interactions resulting in $\mu + p$ final states, including hits from secondary particles. Events with a proton track of more than 20 hits were processed, and any trajectory of more than 20 hits was included in the input data. The expectation is to reconstruct two or more tracks (lower numbers are better). 878 events passed the range cut, of which approximately 11% fall into the one-track bin here. The reconstruction can therefore be said to be 89% efficient.

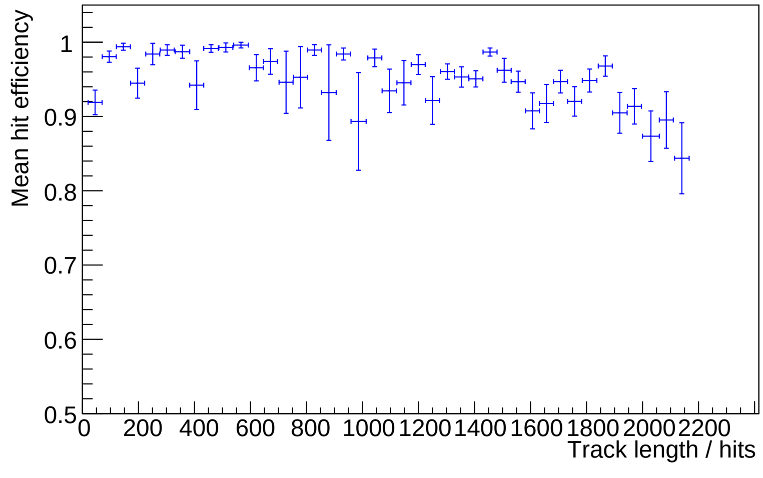


Figure 5.14: The hit efficiency (number of hits from the input that remain in an output cluster) for the CA operating on CCQE ν_μ interactions with a $\mu + p$ final state. The efficiency is high for all truth track lengths, though the reduction in efficiency towards longer tracks remains.

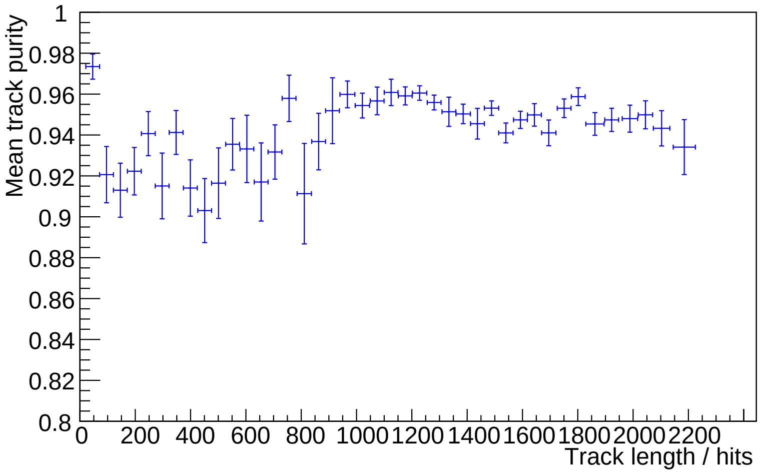


Figure 5.15: The track purity (fraction of hits within a cluster that originate from the same truth track) for the CA operating on CCQE ν_μ interactions with a μ + p final state. The purity is high for all cluster lengths.

now consider the additional primary particle, decay products from the μ or π^+, and delta electrons or hadronic reinteraction of the proton, as before. Only 26 tracks (3%) were reconstructed with fewer than three tracks. The CA therefore successfully clusters 97% of these events. Figure 5.16 shows an example of the CA output for a μ + p + π^+ event in which both the μ and π^+ decay.

The hit efficiency (figure 5.18) for reconstruction of μ + p + π^+ final states is high, but has the same characteristic tail-off with increasing track length. This reinforces the reasoning that the tail-off is due to large scattering events in the muon tracks as the particles deposit energy and slow down. The purity (figure 5.19) is lower than the CCQE case for short tracks, but this is expected because the region around the neutrino interaction vertex is more densely populated, making it harder for the CA to correctly cluster the hits. Further out, away from these sources of noise, the CA performs as well as in the CCQE case, and the purities of long tracks in CC1π events are comparable to those from the CCQE events.

5.6 Conclusions

The CA performs three-dimensional clustering with high efficiency, resulting in clusters of high purity (typically over 90%). There are a number of parameters that affect the operation of the algorithm, and these must, to an extent, be tuned to the particular application; for example the parameters providing optimal results for

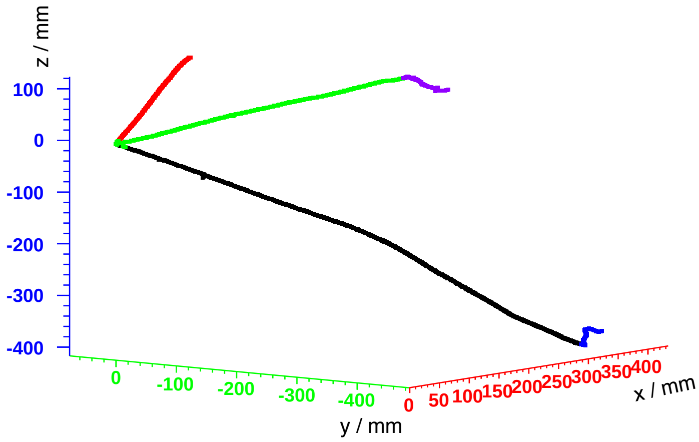


Figure 5.16: Clusters found by the CA for a charged current ν_μ interaction resulting in a $\mu + p + \pi^+$ final state. The μ and π^+ both decay within the detector volume, and the CA produces one cluster for each major particle in the event; proton (red), π^+ (green) and its decay product (purple), the μ (black) and its Michel electron (blue). Axes are labelled in mm.

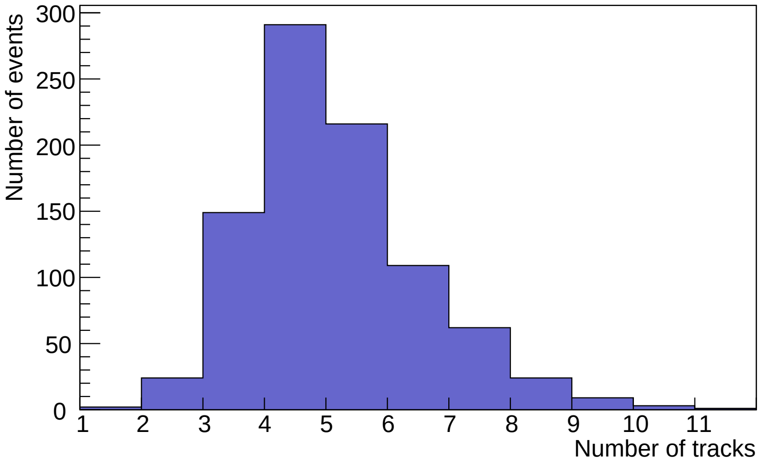


Figure 5.17: Distribution of reconstructed track counts for charged current ν_μ interactions resulting in $\mu + p + \pi^+$ final states, including hits from secondary particles. Events with both proton and pion tracks containing more than 20 hits each were processed, and any trajectory of more than 20 hits was included in the input data. The expectation is to reconstruct at least three tracks, typically more. 891 events passed the range cut, of which approximately 3% were reconstructed with fewer than three tracks.

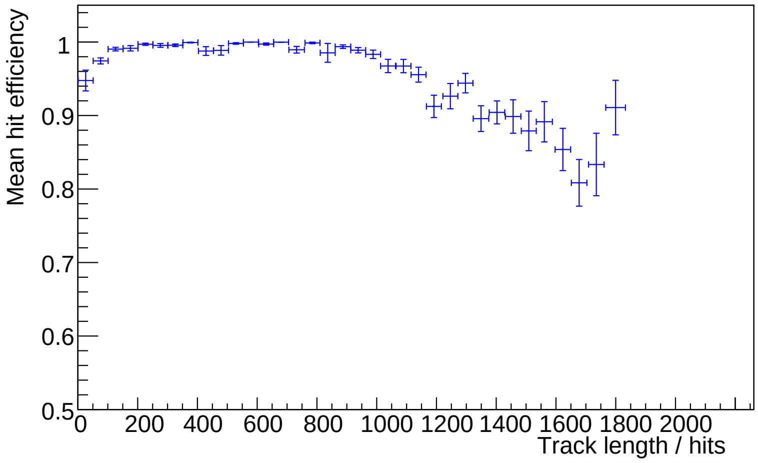


Figure 5.18: The hit efficiency (number of hits from the input that remain in an output cluster) for the CA operating on ν_μ charged current interactions with a $\mu + p + \pi^+$ final state. The efficiency is high for all track lengths, but exhibits the same decline with increasing track length as found in the CCQE reconstruction.

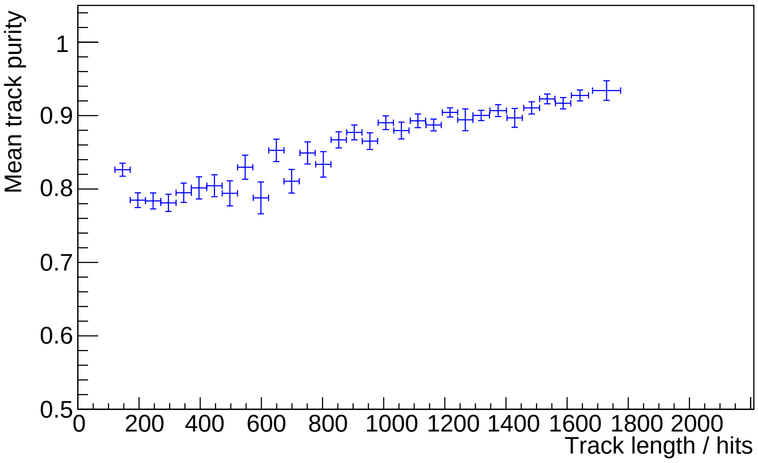


Figure 5.19: The track purity (fraction of hits within a cluster that originate from the same truth track) for the CA operating on ν_μ charged current interactions with a $\mu + p + \pi^+$ final state. Short tracks exhibit low purity here, most likely because the addition of a final state π^+ means that the region around the neutrino interaction vertex is more densely populated than in the CCQE case. High purity is regained for longer tracks, i.e. away from the vertex along the μ track.

clean straight tracks differ from those used to reconstruct noisy physics tracks. The resulting clusters typically require further processing, such as merging and track fitting, but as a tool for hit association, the CA performs well when applied to the typical LAr TPC data with high hit densities.

Most of the inefficiencies have topological causes; typically large angular deviations from straight lines, or a high density of tracks close to a vertex, both of which reduce the ability of the CA to produce a small number of pure clusters. Much of this inefficiency could be recovered by applying other algorithms, such as the feature detection algorithm of chapter 4.5 to locate interaction and decay vertices, then removing hits in a small sphere around those features. This would leave simpler line-like objects for the CA to work with. Another approach might be to explicitly include the charge deposits as an additional 'coordinate', linking together only those cells that form a consistent ionisation energy loss profile.

The CA could be used as part of a larger reconstruction package not only to find and cluster tracks, but also to remove track-like objects prior to shower characterisation and analysis. The identification and analysis of electromagnetic and hadronic showers is an important goal for software reconstruction of events in LAr TPCs, where it is complicated by the homogeneous nature of the detector allowing tracks and showers to develop alongside each other.

In conclusion, the use of a cellular automaton to perform three-dimensional reconstruction of tracks in liquid Argon data is a useful addition to the current knowledge base surrounding LAr TPC reconstruction, and is particularly powerful when used as part of an integrated reconstruction package, where its inefficiencies can be compensated with other algorithms.

6
Track Fitting

6.1 Introduction

Track fitting is the process by which track parameters are determined from a collection of clustered hits. The track parameters include the trajectory, as well as kinematic variables such as the energy or momentum of the particle in question. Track fitting in particle physics is typically performed using a variant of the Kalman filter, which is the optimal estimator for the state of a discrete linear dynamic system.

Originally devised as a noise-reduction and signal filtering technique for communications, the Kalman filter determines optimal estimates of past, present and future states of a linear system based on a series of time-ordered measurements used in conjunction with a statistical model of the system and its measurement errors. It is used extensively for both track and vertex fitting in particle physics, most often in a magnetised detector, where it separates the curvature of a particle due to the magnetic field from the noise introduced by direction changes as a result of multiple scattering.

6.2 The Icarus Kalman Filter

The Icarus [43] experiment presented a Kalman filter [76] which makes statistical use of the distribution of the multiple scattering angle θ to measure the momentum of a particle in a non-magnetised detector [77]. The Kalman filter is used to filter out the noise introduced by limited detector resolution. The details of the Icarus algorithm are presented here as an overview of the mechanism of action of the Kalman filter, but also as the basis for the Latte Kalman filter, which aims to perform the same task.

The Kalman filter operates on a discrete set of states, each represented by a state vector $\mathbf{x}_k$. These states correspond to points on the track, and can in principle be measured anywhere along it. In practice, the track is split into segments of fixed length, and the endpoints of these segments define the set of planes at which the state vector $\mathbf{x}_k$ is evaluated.

The track system is described by the linear equation: [77]

$$\mathbf{x}_k^- = F_{k-1}\mathbf{x}_{k-1} + \mathbf{w}_{k-1} \tag{6.1}$$

Here, F_{k-1} is a matrix defining the propagation of the state vector from plane $k-1$ to plane k, $\mathbf{w}_{k-1}$ is the noise associated with this propagation (which is random, in liquid Argon) and acts to smear the state vector, $\mathbf{x}_{k-1}$ is the filtered state vector in plane $k-1$, and $\mathbf{x}_k^-$ is the predicted state vector in plane k.

The state vector is not usually observed directly; instead, quantities such as the scattering angle are observed, and these are related to quantities in the state vector (such as the particle momentum) through a transformation matrix:

$$\mathbf{m}_k = H_k\mathbf{x}_k + \epsilon_k \tag{6.2}$$

where H_k is a matrix transforming a measurement vector $\mathbf{m}_k$ into a state vector $\mathbf{x}_k$, and ϵ_k represents measurement noise (errors on the measurements).

The process noise $\mathbf{w}_k$ and the measurement noise ϵ_k are taken to be unbiased and with finite variance. The covariance matrices are Q_k (for process covariance) and V_k (for measurement covariance). If the process and measurement noise are both random Gaussian variables, the Kalman filter will be the optimal estimator of the system state.

The Kalman filter proceeds through the following three steps, repeated for each new plane k and its corresponding measurement $\mathbf{m}_k$.

---

1. **Prediction:** Given the state vector $\mathbf{x}_{k-1}$ in plane $k-1$, the prediction step of the Kalman filter estimates the state vector in a future plane k, in the absence of noise. The state $\mathbf{x}_k^-$ is the predicted state at plane k, using the information contained in all of the state vectors up to $k-1$.

$$\mathbf{x}_k^- = F_{k-1}\mathbf{x}_{k-1} \tag{6.3}$$

2. **Filtering:** Given a predicted state vector $\mathbf{x}_k^-$ (or a suitable initial state), the

filtering step determines the present state $\mathbf{x}_k$ by taking into account the measurements of all previous planes (via the predicted state) and the current plane (via the measurement $\mathbf{m}_k$). Since the previous measurements are included via their contribution to the state vector, the Kalman filter automatically includes correlations between measurements.

$$\mathbf{x}_k = \mathbf{x}_k^- + K_k \left(\mathbf{m}_k - H_k \mathbf{x}_k^- \right) \tag{6.4}$$

where K_k is the Kalman gain matrix, and is related to the covariance of the state vector and the measurement noise.

3. **Smoothing:** When the filter is applied to plane $k+1$, a smoothing step can be used to improve the estimate of the state at plane k, taking into account all measurements up to and including $k+1$.

$$\mathbf{x}_k^n = \mathbf{x}_k + A_k \left(\mathbf{x}_{k+1}^n - \mathbf{x}_{k+1}^- \right) \tag{6.5}$$

where A_k is a smoothing gain matrix and $\mathbf{x}_k^n$ is the filtered state vector after smoothing, for plane k.

For particles moving in a non-magnetised detector, the trajectory is split into small track segments, each fitted to a straight line. The state vector $\mathbf{x}_k$ contains the inverse momentum, coordinates, and track slopes:

$$\mathbf{x}_k = \begin{pmatrix} \frac{1}{p} \\ x \\ y \\ \frac{dx}{dz} \\ \frac{dy}{dz} \end{pmatrix} \tag{6.6}$$

The transportation matrix F_k provides a straight line extrapolation, with no changes to the track parameters. The energy deposited in a track segment (which can be determined from the charge collected) is subtracted from the momentum estimate in plane $k-1$ to update the momentum estimate in the state vector for plane k.

$$F_k = \begin{pmatrix} \frac{1}{1-E_{\mathrm{dep}}/p} & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & \Delta z & 0 \\ 0 & 0 & 1 & 0 & \Delta z \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \tag{6.7}$$

The measurement vector includes the coordinates and slopes, but not the inverse momentum. Instead, the angle θ_0 between adjacent track segments at the plane k is used; these angles build up a θ_0^{rms} measurement, which is fed to the Kalman filter. Equation (6.8) relates this scattering angle to the momentum [78].

$$\theta_0^{\mathrm{rms}} = \frac{13.6\,\mathrm{MeV}}{\beta c p} z \sqrt{\frac{l}{X_0}} \left(1 + 0.038 \ln\left[\frac{l}{X_0}\right]\right) \tag{6.8}$$

where β is the velocity of the particle, p is its momentum and z is the charge, X_0 is the radiation length and l the track segment length. The measurement vector $\mathbf{m}_k$ is then:

$$\mathbf{m}_k = \begin{pmatrix} \theta_0^{\mathrm{rms}} \\ x \\ y \\ \frac{dx}{dz} \\ \frac{dy}{dz} \end{pmatrix} \tag{6.9}$$

The matrix H_k, which relates the measurement and state vectors, is given by:

$$H_k = \mathrm{diag}\,(C, 1, 1, 1, 1) \tag{6.10}$$

where C is the constant multiplying $\frac{1}{p}$ in equation (6.8).

Finally, the covariance matrices Q_k (for the system covariance) and V_k (for the measurement covariance) must be taken into account. The Icarus experiment use system covariance matrices from [79], where the covariances for multiple scattering are carefully derived, while the measurements are assumed to be uncorrelated, and the matrix V is diagonal, with each component representing the measurement error.

6.3 The Latte Kalman Filter

The Latte Kalman filter follows closely the formulation used in the Icarus filter. The implementation is based on that present in the SciPy [64] package for scientific programming in the Python language, with a linear system corresponding to the track fitting mechanism used by Icarus.

The state vector x_k is:

$$x_k = \begin{pmatrix} \frac{1}{p} \\ y \\ z \\ \frac{\Delta y}{\Delta x} \\ \frac{\Delta z}{\Delta x} \end{pmatrix} \tag{6.11}$$

The measurement vector is identical to the state vector, except for the scattering angle θ_0^{rms} in place of the inverse momentum:

$$x_k = \begin{pmatrix} \theta_0^{\text{rms}} \\ y \\ z \\ \frac{\Delta y}{\Delta x} \\ \frac{\Delta z}{\Delta x} \end{pmatrix} \tag{6.12}$$

The process covariance matrix Q is a 5×5 matrix defined as: [79]

$$Q = \begin{pmatrix} \left(\frac{0.01}{p}\right)^2 & 0 & 0 & 0 & 0 \\ 0 & \Delta x^2 P_{33} & \Delta x^2 P_{34} & -\Delta x P_{33} & -\Delta x P_{34} \\ 0 & \Delta x^2 P_{34} & \Delta x^2 P_{44} & -\Delta x P_{34} & -\Delta x P_{44} \\ 0 & -\Delta x P_{33} & -\Delta x P_{34} & P_{33} & P_{34} \\ 0 & -\Delta x P_{34} & -\Delta x P_{44} & P_{34} & P_{44} \end{pmatrix} \tag{6.13}$$

where p is the momentum at that step, Δx is the segment length in the x direction, and

$$\begin{aligned} P_{33} &= \theta(1 + S_y^2)(1 + S_y^2 + S_z^2) \\ P_{34} &= \theta S_y S_z(1 + S_y^2 + S_z^2) \\ P_{44} &= \theta(1 + S_z^2)(1 + S_y^2 + S_z^2) \end{aligned}$$

where θ is the scattering angle, and $S_y = \dfrac{\Delta y}{\Delta x}$ and $S_z = \dfrac{\Delta z}{\Delta x}$ are the slopes in y and z.

The measurement covariance matrix is diagonal, with components representing the measurement uncertainties on the values of θ, y, z, $\dfrac{\Delta y}{\Delta x}$ and $\dfrac{\Delta z}{\Delta x}$ respectively:

$$V = \text{diag}\left(1 \times 10^{-4},\ 0.1,\ 0.1,\ 1 \times 10^{-3},\ 1 \times 10^{-3}\right) \tag{6.14}$$

Tracks are split into segments of length L (in mm), with the Kalman filter applied to the points between two segments, and using the slopes of those segments to determine the scattering angle θ and the distribution of those angles to establish θ_0^{rms}. The segment lengths are chosen to be a multiple of the radius R (in mm) used for a charge smoothing process which reduces the total number of hits in the track, replacing groups of hits with a single charge-weighted hit. The track momentum is taken from the first component of the state vector after running along the entire track.

6.4 Tuning the Kalman Filter

The majority of parameters for the Kalman Filter are fixed by either the process itself (the description of multiple scattering that relates scattering angles to momenta) or by the measurement errors. The remaining free parameters are R and L, the charge smoothing radius and segment length, respectively.

A sample of single muon events was generated using the Lamu simulation, producing 100 muons at each energy in the range 100 to 5000 MeV in 100 MeV increments. The muon hits from these events were processed using the Kalman filter, and the resulting momentum measurements recorded. The true momentum p is related to the initial kinetic energy T by:[1]

$$p = \sqrt{T^2 + 2Tm} \tag{6.15}$$

where m is the mass of the muon, 105.6583715 MeV [21].

The segment length was initially chosen to be six times the charge smoothing radius, and the radius varied between 20.1 mm and 50.1 mm. Figure 6.1 shows the resulting measurements, represented by the mean reconstructed momentum as a function of the true momentum. Smaller smoothing radii (20.1 mm) give smaller standard deviation in the results, especially at high momentum, while the mean values are closer to the true momentum for larger smoothing radii of 30.1 mm to 40.1 mm. The distributions of residuals (defined as $p_{\mathrm{true}} - p_{\mathrm{recon}}$) are presented in figure 6.2 for each of the radii considered here. At momenta over approximately 500 MeV, the Kalman filter underestimates the momentum when used with short (20.1 mm and 30.1 mm) segment lengths, but overestimates it for larger segment lengths. As the segment length increases, the range of momentum estimates also

[1] $E^2 = p^2 + m^2$ and $E = T + m$, so $p^2 + m^2 = T^2 + m^2 + 2Tm$ and the result above can be obtained by a cancellation of m^2 followed by the square root operator. This of course assumes natural units, i.e. $c = 1$.

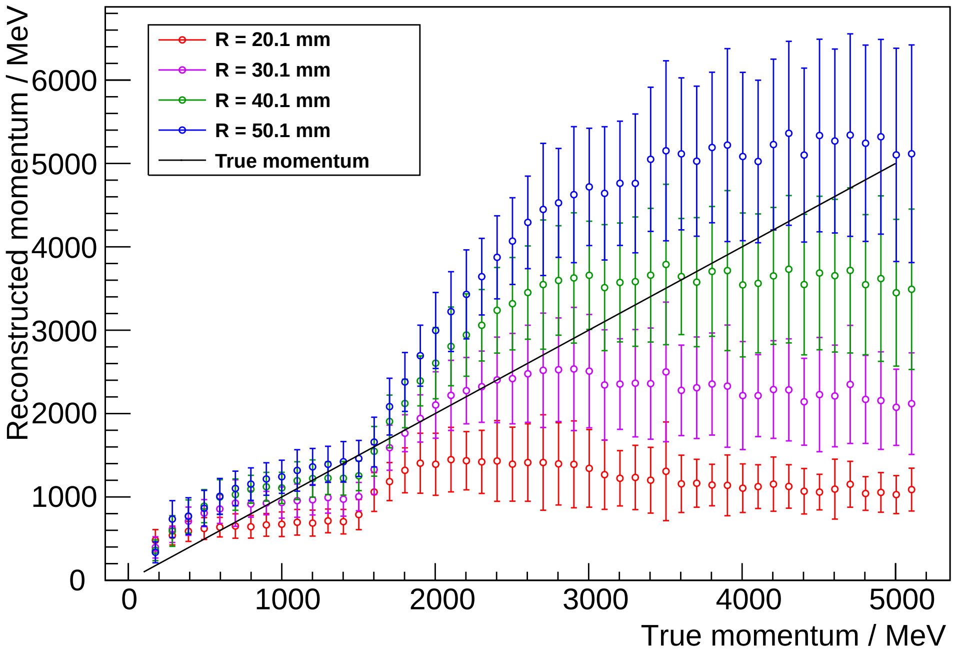


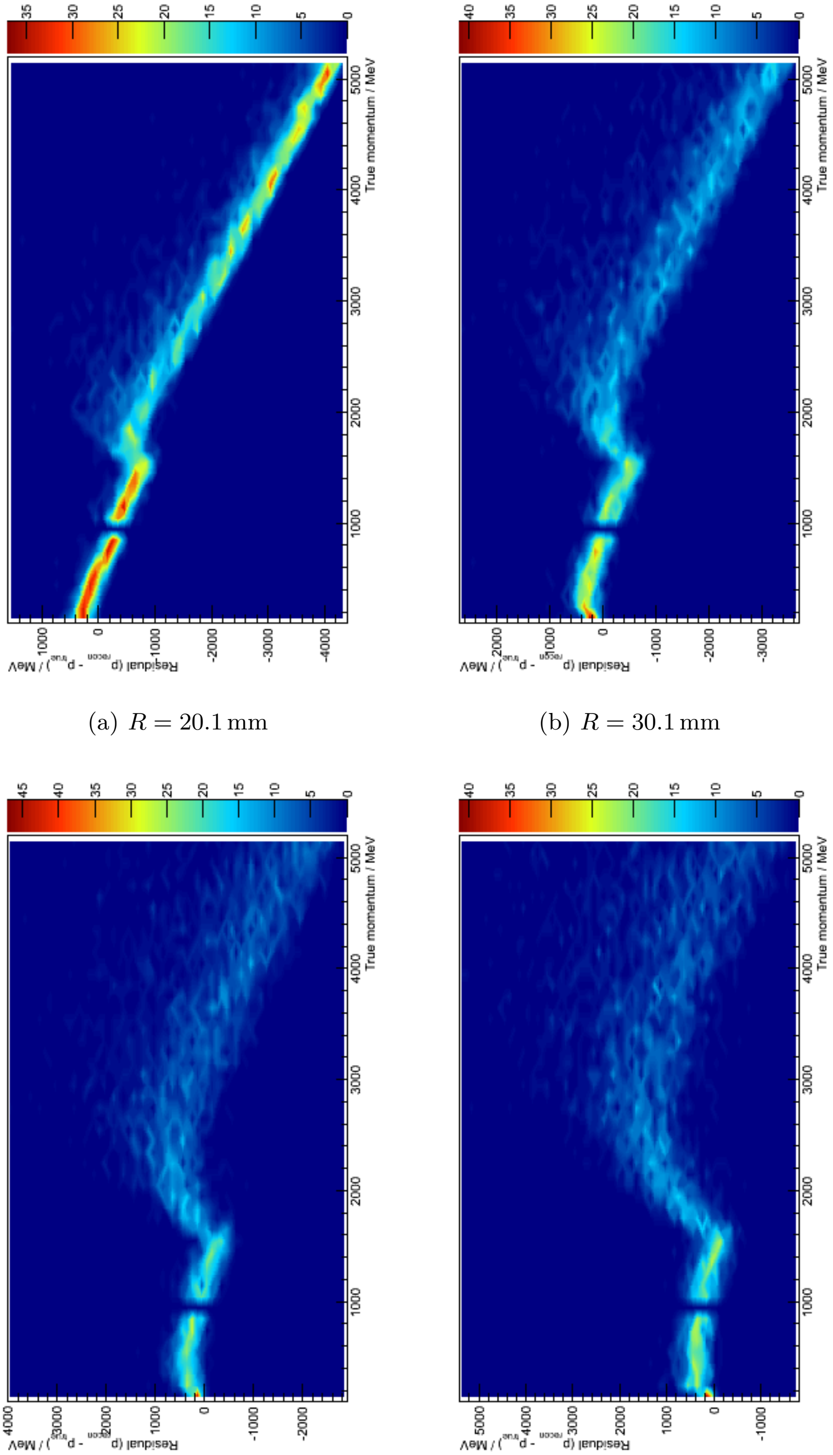
Figure 6.1: The mean and standard deviation of reconstructed momenta as a function of the true momenta of single muons after application of the Kalman filter, for four values of the charge smoothing radius R (and consequently the segment length $L = 6R$).

increases (both in the positive and negative directions). The Kalman filter in this configuration does not, therefore, provide reliable estimates of momentum.

6.5 Momentum Measurements of CCQE μ tracks

In the tuning study of chapter 6.4, muons of a fixed kinetic energy (and, therefore, momentum) were used to attempt to validate the Kalman filter algorithm and the choice of parameters. The results indicate that for a 50.1 mm charge smoothing radius, and choosing segment lengths six times that radius, the mean momentum measurement is close to the true momentum, but the range of measurements is extremely large.

In this section, the Kalman filter is applied to hits produced by the muon generated from charged current ν_μ interactions on Argon nuclei, resulting in $\mu + p$ final states. Two neutrino energies are considered; $E_\nu = 770\,\text{MeV}$ (1000 events) and $E_\nu = 4.5\,\text{GeV}$ (10^4 events). For the high energy data, the detector simulation was expanded to a cylinder of radius 25 m and height 50 m. Muons produced in



(a) $R = 20.1\,\mathrm{mm}$

(b) $R = 30.1\,\mathrm{mm}$

(c) $R = 40.1\,\mathrm{mm}$

(d) $R = 50.1\,\mathrm{mm}$

Figure 6.2: Residuals (reconstructed momentum minus true momentum) for the Kalman filter with different values of the charge smoothing radius R, and therefore of the segment length $L = 6R$. The gaps around 1000 MeV are due to the 100 MeV bin width used. See text for discussion.

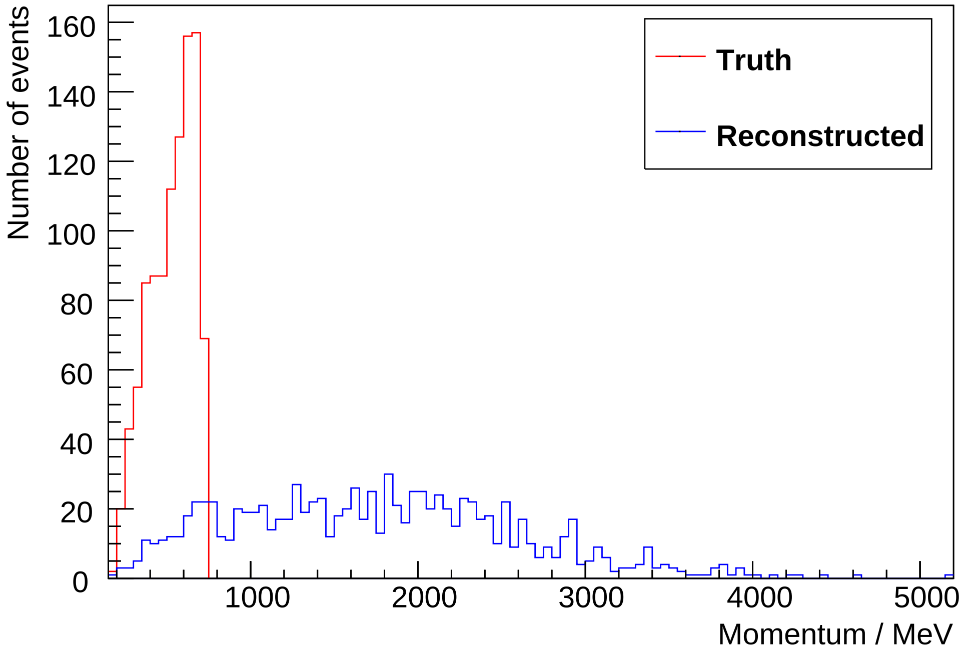


Figure 6.3: Distribution of momenta for muons produced in a charged current low energy ν_μ interaction at 770 MeV resulting in a $\mu + p$ (only) final state (in red). The distribution of reconstructed momenta (in blue) is from the output of the Kalman filter applied to 1000 such events. The distribution produced by the Kalman filter does not match the true distribution, being both flatter and shifted to higher momentum. The filter therefore overestimates (in general) the momentum of these muons (represented by the shift to higher momentum), but not in a consistent manner (represented by the flatness).

these interactions have a range of energies and momenta, as well as a distribution of trajectories determined by Genie from the interaction physics models. The true momentum, stored in the Genie event data, was extracted for comparison against the measurements from the Kalman filter. Figure 6.3 shows the two distributions for the 770 MeV neutrinos, and figure 6.4 shows the distributions for the 4.5 GeV neutrinos.

The Kalman filter overestimates the momentum in both the low energy and high energy neutrinos, though the effect is much more pronounced at high energy. Since the method of estimating momentum through the Kalman filter relies on measurements of the multiple scattering angle, these observations correspond to an underestimate of the angle. The angle is more likely to be underestimated at high

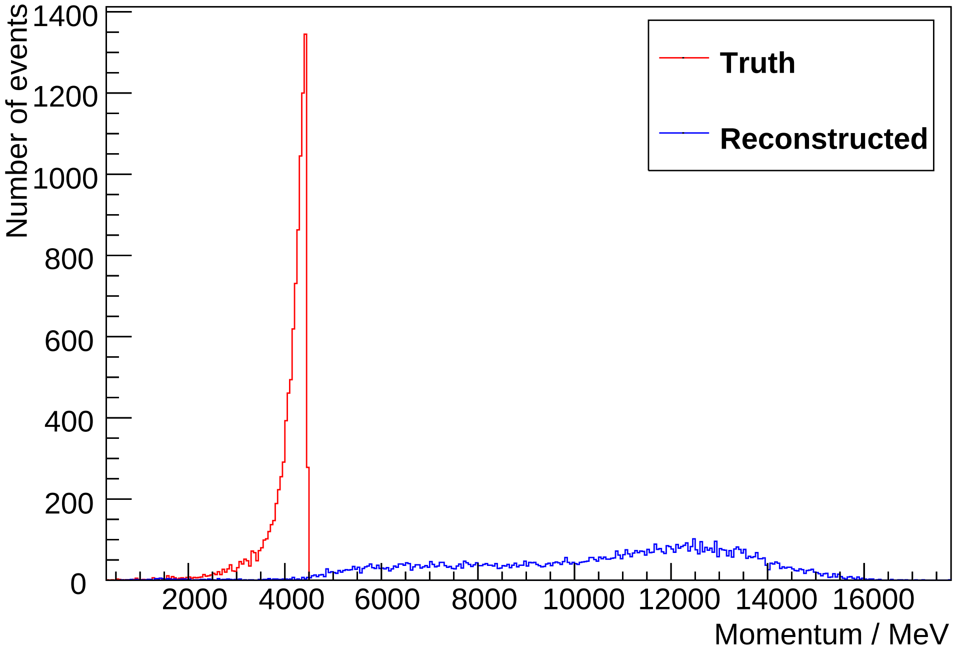


Figure 6.4: Distribution of momenta for muons produced in a charged current high energy ν_μ interaction at 4.5 GeV resulting in a $\mu + p$ (only) final state (in red). The distribution of reconstructed momenta (in blue) is from the output of the Kalman filter applied to 10^4 such events. As for the low energy distribution, the Kalman filter produces overestimates of the muon momentum, though the effect is much more pronounced here.

energy since the effect of scattering is smaller there.

6.5.1 Constrained Momentum Measurements

The data in figure 6.1 implies that an energy (momentum) dependent choice of the smoothing radius may provide better results, but it is not possible to use this in a general purpose reconstruction algorithm, since the momentum is not known in advance. Indeed, the purpose of applying the Kalman filter is to obtain an estimate of this momentum. The distributions of reconstructed momenta for $E_\nu = 770\,\mathrm{MeV}$ and $E_\nu = 4.5\,\mathrm{GeV}$ extend far beyond the energy of the parent neutrino; a situation which is unphysical. Since the maximum possible neutrino energy (i.e. the beam energy) can be known in advance, it is possible to run the Kalman filter in a configuration that enforces constraints on the momentum estimate at each step.

In figure 6.5 the distributions of true and reconstructed momenta are shown for 770 MeV CCQE interactions resulting in $\mu + p$ final states, where the Kalman filter was constrained such that at each step, the estimated momentum $|\mathbf{p}|$ must lie between 0 MeV and 770 MeV. The residuals (i.e. the reconstructed momentum minus the true momentum) for this reconstruction are shown in figure 6.6; the residual is calculated per muon, and a value close to zero indicates compatibility between the true momentum and the reconstructed momentum estimate.

Both plots demonstrate that even with the momentum constrained, the reconstructed values do not agree with the truth. The constraint serves only to truncate all momenta with estimates that would exceed the maximum, resulting in a large peak at the maximum value allowed. From the residuals plot, it is clear that the Kalman filter still over-estimates the momentum of a given track (that is, there are more events on the positive side of the zero point than on the negative side), and that only 5 events out of 1000 have a residual of zero, indicating perfect reconstruction. Many more events have large residuals of over 200 MeV, and these momentum estimates are not suitable for use in later stages of a general purpose reconstruction chain.

The analysis was repeated with 10^4 muon tracks from interactions of 4.5 GeV muon-neutrinos, setting the maximum momentum estimate at each step to 4.5 GeV. The resulting distribution is shown in figure 6.7, where it is compared to the true distribution. The residuals are not shown, since no events were reconstructed with a momentum less than the maximum allowed by the constraints, hence this reconstruction is no better than simply assuming that each muon carries the maximum possible momentum. These results again demonstrate that the Kalman filter is unfit for the general purpose reconstruction of muon momenta in this environment.

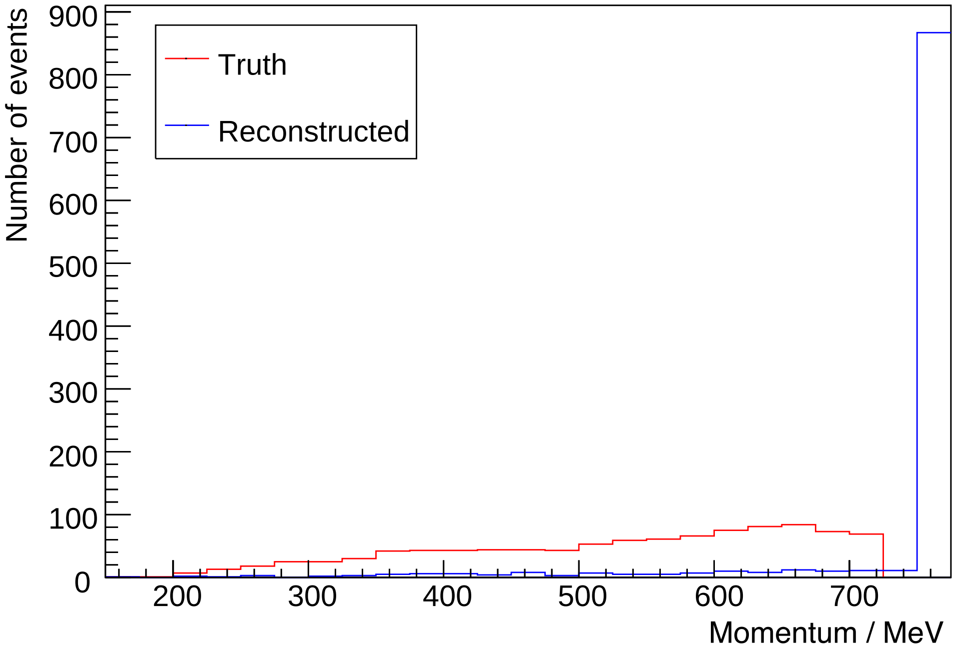


Figure 6.5: Distribution of momenta for muons produced in a charged current ν_μ interaction at 770 MeV resulting in a $\mu + p$ (only) final state (in red). The distribution of reconstructed momenta (in blue) is from the output of the Kalman filter applied to 10^3 such events, with the momentum estimate at each step constrained such that $0 \leq |\mathbf{p}| \leq 770$ MeV. The distributions do not agree, and the highest momentum bin contains almost all of the reconstructed events.

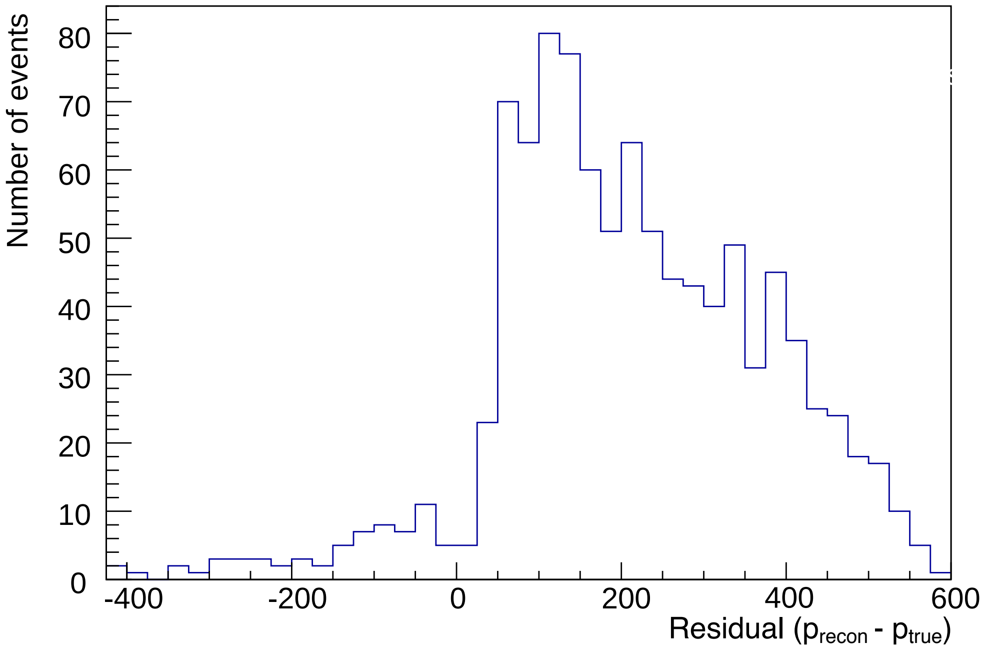


Figure 6.6: Residuals (reconstructed momentum minus true momentum) for a Kalman filter performing a constrained momentum reconstruction of muons resulting from 770 MeV muon neutrinos interacting in liquid Argon and producing $\mu + p$ (only) final states. Data from 10^3 events is shown, of which only 5 have a residual close to zero (which indicates a perfect reconstruction). Most events have a positive residual (which indicates a tendency for the algorithm to over-estimate) and there are many events with large residuals. These results indicate that the algorithm used is not well-suited to this task, and is not capable of delivering accurate momentum estimates on data of this type.

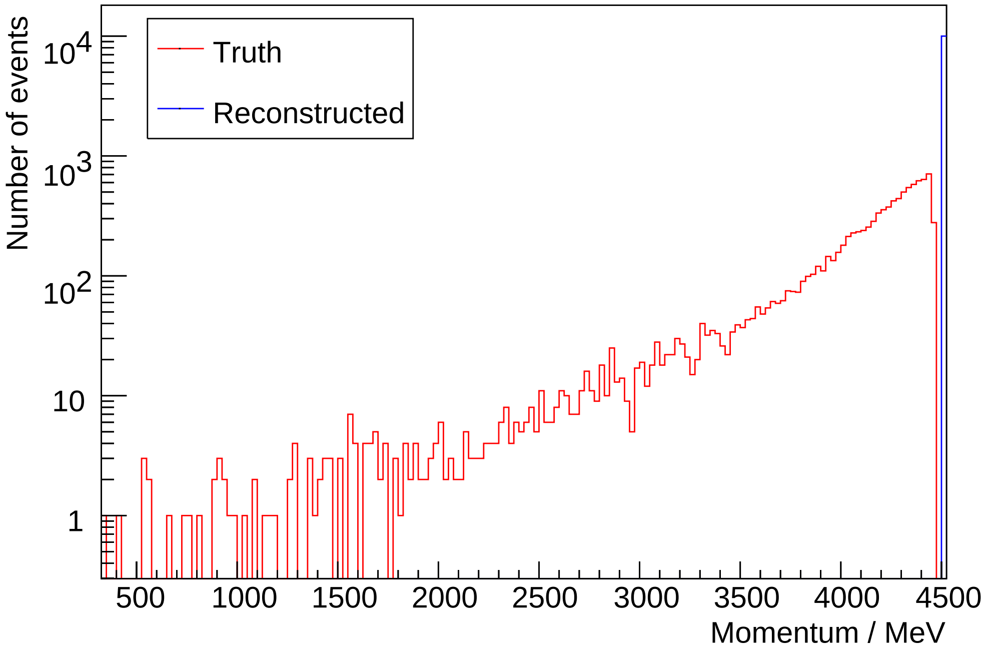


Figure 6.7: Distribution of momenta for muons produced in a charged current ν_μ interaction at 4.5 GeV resulting in a $\mu + p$ (only) final state (in red). The distribution of reconstructed momenta (in blue) is from the output of the Kalman filter applied to 10^4 such events, with the momentum estimate at each step constrained such that $0 \leq |\mathbf{p}| \leq 4500\,\text{MeV}$. The y-axis (number of events) is shown on a log scale. There is no overlap at all between the two distributions; every muon track is reconstructed with the highest allowed momentum.

6.6 Conclusions

The Kalman filter is traditionally a powerful tool for track parameter estimation, but in the environment of a LAr TPC it does not perform well, at least for the estimation of track momentum. The Icarus experiment achieved better results, but with a smaller measurement error on their individual hits ($400\,\mu$m) [77]. For data of the type presented here, the Kalman filter (as implemented) is not a good reconstruction tool for the purposes of estimating particle momentum. Attempts to constrain the estimates at each step of the procedure do not improve the end result and serve only as a simple 'sanity check' to avoid unphysical reconstructed momenta.

The spatial track parameters were not considered here because muons in liquid Argon undergo frequent large scatters, resulting in tracks that can have substantial curvature or kinks. These track features make it extremely difficult to compare the state of the Kalman filter to the spatial parameters of the whole track, and would have to be considered on a smaller scale, with instantaneous state vectors.

7 Particle Identification

7.1 Introduction

Following on from the process of clustering hits together into associated groups, and fitting tracks to obtain parameters such as the slope or particle momentum, it is essential to be able to determine the nature of the particle that produced those hits.

For CCQE neutrino interactions it is of critical importance to be able to identify the flavour of the lepton that was produced, and to be able to associate a track with the passage of a muon (if the interaction involved a ν_μ). Muon identification is often performed with the help of small detector units around the sides of the main detector. A muon typically behaves as a minimum ionising particle, and will often travel all the way through a detector and leave through one of the sides. A signal in such a muon detector is a strong indicator that the track leading into it was produced by a muon.

In the case of a LAr TPC, the aim is to fully contain muons (at least, if they are produced somewhere near the centre of the detector volume), and identification must be performed using information from the track reconstruction stage. For the simplest category of charged current interactions, the muon should be represented by the longest track visible in the event, and it is on this basis that the primary identification of muon tracks will be carried out.

For other types of interaction, or for electron neutrinos, there is a much greater chance of an electromagnetic or hadronic shower developing in the event. One advantage often claimed of LAr TPCs is the ability to achieve electron–pion separation by looking at the rate of energy loss, dE/dx. Attempts at particle identification based on the properties of electromagnetic or hadronic showers are not

	μ	p	e^-	$\pi^\pm$
Included	960	57	0	0
Excluded	40	1143	67228	3

Table 7.1: Composition of tracks after applying a 1000 hit cut on the truth tracks from simulation of 770 MeV ν_μ interactions producing μ + p final states. The cut selects muon tracks with $96.0 \pm 0.6\%$ efficiency and $94.4 \pm 0.8\%$ purity.

considered here, but see [80] for details of one such analysis. Since the homogeneous nature of LAr TPCs allows for the coexistence of tracks and showers, a full analysis framework must be able to cope with showers, in addition to tracks. Furthermore, in order to be applicable to ν_e events, it must be possible to identify electromagnetic showers and obtain calorimetric information from them, in order to be able to estimate the energy of the incoming ν_e. This is an area in which LAr TPCs should excel, but the focus of this thesis is on track reconstruction techniques; an already difficult area for fully automated reconstruction in such a fine-grained environment.

7.2 Muon Track Identification

In order to be able to select muons based on the track length (approximated as the number of hits contained), it is useful to know the distributions of track lengths for various particle types which may be present in the charged current interactions considered.

7.2.1 770 MeV $\nu_\mu \to \mu + p$ (CCQE) Interactions

This study begins by looking at those distributions for the products of charged current interactions of 770 MeV ν_μ which produced μ + p or μ + p + π^+ final states, making use of the truth information saved by the simulation. Figure 7.1 shows the distribution of lengths (in terms of number of hits) for muon, proton and electron tracks. At neutrino energies around 770 MeV, there is a strong case to be made for treating all tracks longer than 1000 hits as muons, resulting in a trade-off between selection efficiency for muons and contamination by protons.

Table 7.1 shows the number of tracks included and excluded by a 1000 hit cut, where the track type is determined from truth information. Since the electron tracks are produced arbitrarily along the muon track, it is impossible to eliminate electron hits from the muon track, even with clustering techniques such as those provided by the cellular automaton (see chapter 5). The purity of any such tracks

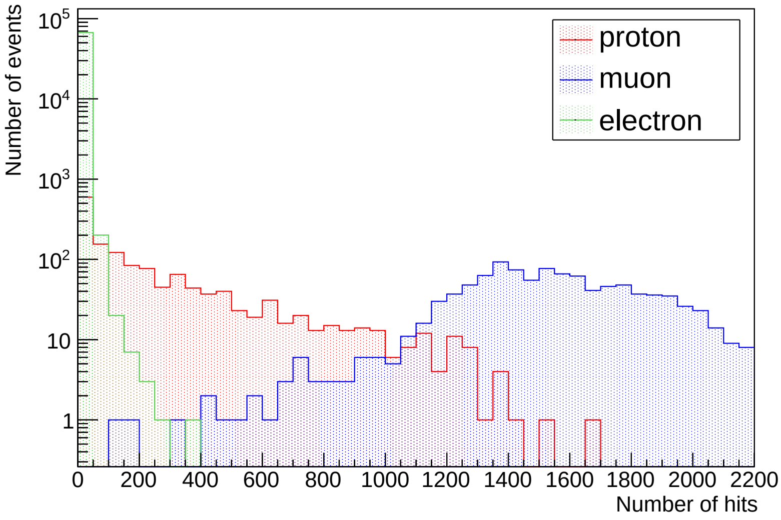


Figure 7.1: Distribution of track lengths, represented as number of hits contained in a track, for muon (blue), proton (red) and electron (green) tracks produced in charged current interactions of ν_μ at 770 MeV. The muon and proton distributions are naturally separated at around 1000 hits. An extremely large number of short electron tracks can be seen, corresponding to the production of delta electrons along the length of the trajectories of the primary muon and proton. Most electron tracks have fewer than 50 hits.

will therefore be reduced by the presence of hits from these short delta electron tracks, i.e. reconstructed muon tracks will not be 100% pure at the hit level.

7.2.2 770 MeV $\nu_\mu \rightarrow \mu + p + \pi^+$ (CC1π) Interactions

This data set contains 1000 events where a 770 MeV ν_μ underwent a charged current interaction with an Argon nucleus, producing a $\mu + p + \pi^+$ final state. The distribution of track lengths (represented in terms of the number of hits) for muons, protons and charged pions is shown in figure 7.2. Here, the separation of muons from other particles by track length alone is not a realistic proposal; although the muons dominate the longer tracks, a number of pion tracks are also long enough to provide significant pollution of the muon selection if track length alone is used to discriminate particle type.

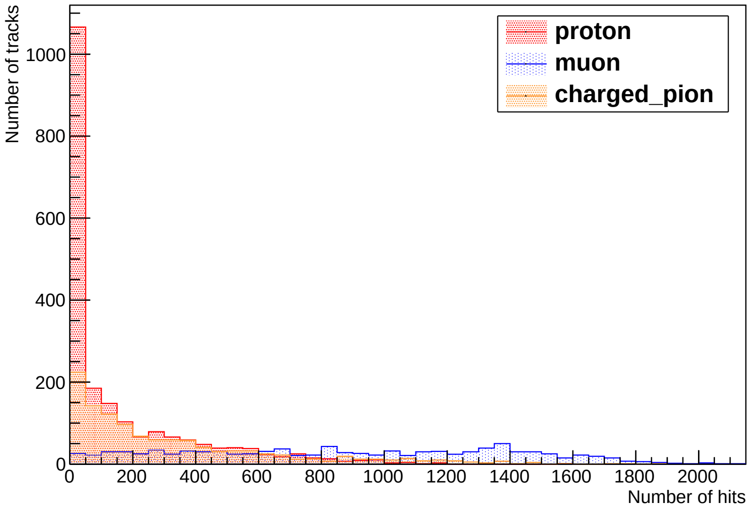


Figure 7.2: Distribution of track lengths, represented as number of hits contained in a track, for muon (blue) and proton (red) tracks produced in charged current interactions of ν_μ at 770 MeV resulting in $\mu + p + \pi^+$ final states. More than 1000 proton tracks are present, indicating that the π^+ sometimes produces one or more protons. The distributions are not clearly separated, and while muon tracks are still typically long, there is much more contamination from the other particle species.

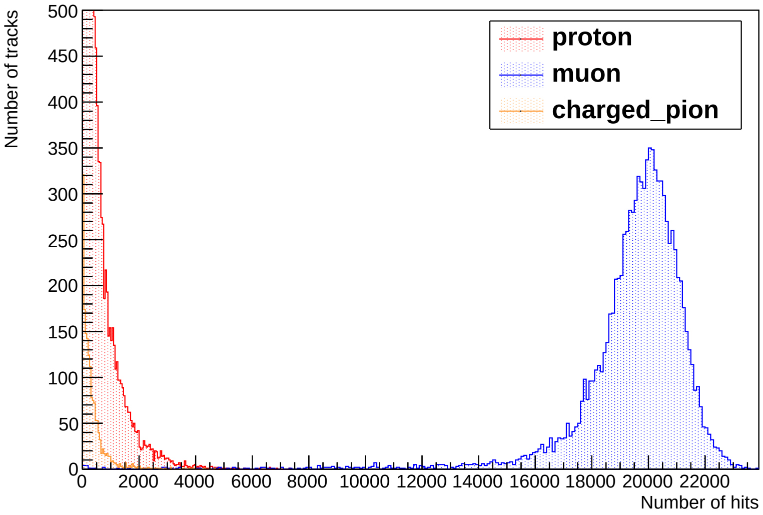


Figure 7.3: Distribution of track lengths, represented as number of hits contained in a track, for muon (blue), proton (red) and charged pion (orange) tracks produced in charged current interactions of ν_μ at 4.5 GeV resulting in $\mu + p$ final states. The proton distribution peaks at over 1.55×10^4 tracks on the left.

7.2.3 4.5 GeV $\nu_\mu \rightarrow \mu + p$ (CCQE) Interactions

This data set contains 10^4 events in which a 4.5 GeV ν_μ underwent a charged current interaction with an Argon nucleus, producing a $\mu + p$ final state. The distribution of track lengths (represented in terms of the number of hits) for muons, protons and charged pions (which are produced as secondary particles, in these events) is shown in figure 7.3. In addition, a large number of short electron tracks are present, the longest containing only 1238 hits (in contrast, the muon peak is at 22000 hits).

For these high energy events, it is clear that any track with more than around 5000 hits must correspond to a muon, and the vast majority of muon tracks have between 14000 and 24000 hits. Using the track length to separate muons from other particles should provide an extremely pure selection of muons, in this case. Table 7.2 shows the number of tracks included and excluded by a 5000 hit cut, where the track type is determined from truth information.

7.2.4 4.5 GeV $\nu_\mu \rightarrow \mu + p + \pi^+$ (CC1π) Interactions

This data set contains 10^4 events in which a 4.5 GeV ν_μ underwent a charged current interaction with an Argon nucleus, producing a $\mu + p + \pi^+$ final state. The

	μ	p	e^-	$\pi^\pm$
Included	9982	14	0	0
Excluded	32	29640	11961433	1646

Table 7.2: Composition of tracks after applying a 5000 hit cut on the truth tracks from simulation of 4.5 GeV ν_μ interactions producing $\mu + p$ final states. The cut selects muon tracks with $99.7 \pm 0.1\%$ efficiency and $99.9 \pm 0.1\%$ purity.

	μ	p	e^-	$\pi^\pm$	Other
Included	9717	24	0	12	12
Excluded	337	67286	12854577	18691	188995

Table 7.3: Composition of tracks after applying a 5000 hit cut on the truth tracks from simulation of 4.5 GeV ν_μ interactions producing $\mu + p + \pi^+$ final states. The cut selects muon tracks with $96.6 \pm 0.2\%$ efficiency and $99.5 \pm 0.1\%$ purity.

distribution of track lengths (represented in terms of the number of hits) for muons, protons and charged pions is shown in figure 7.4. The π^+ and p peaks on the left extend up to 4.3×10^4 tracks. In this case, the muons are once again clearly separated, as for the CCQE events at 4.5 GeV, and a cut at 5000 hits would select almost entirely muon tracks, rejecting those from protons and pions. Table 7.3 shows the number of included and excluded tracks following a 5000 hit cut.

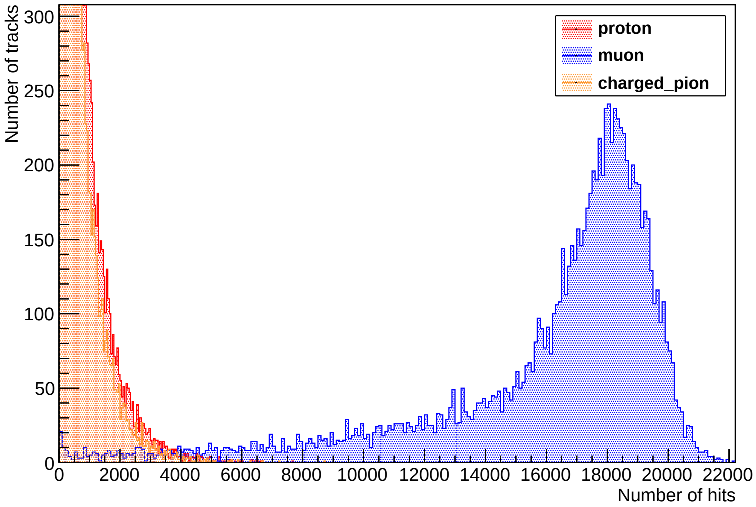


Figure 7.4: Distribution of track lengths, represented as number of hits contained in a track, for muon (blue), proton (red) and charged pion (orange) tracks produced in charged current interactions of ν_μ at 4.5 GeV, resulting in $\mu + p + \pi^+$ final states. The proton and pion peaks on the left extend up to 4.3×10^4, and the plot range has been reduced to better display the separation between hadron (p and π) and lepton (μ) tracks. A cut at 5000 hits would cleanly select muons with low contamination from other tracks.

7.3 Angular Distributions

Relativistic kinematics dictates that the angular distribution of final state particles will depend on their mass [17]. Differences in the angular distribution of muons, protons and charged pions could allow for the construction of PID variables to aid the discrimination between those particle species. In this section, a comparison is made between the angular distributions of protons and muons in CCQE interactions at 0.77 GeV and 4.5 GeV energies, and for CC1π interactions at the same energies, including the distribution of charged pions.

In each case, the data is taken from the output of a set of Genie interactions, using the following quantities based on the final state particle momentum. Note that these correspond to components of a unit vector along the momentum vector, or, equivalently, normalised direction vectors, i.e. direction cosines $\cos \alpha_{x,y,z}$, with a range between -1 and 1.

$$\hat{p}_{x,y,z} = \frac{p_{x,y,z}}{\sqrt{p_x^2 + p_y^2 + p_z^2}} \tag{7.1}$$

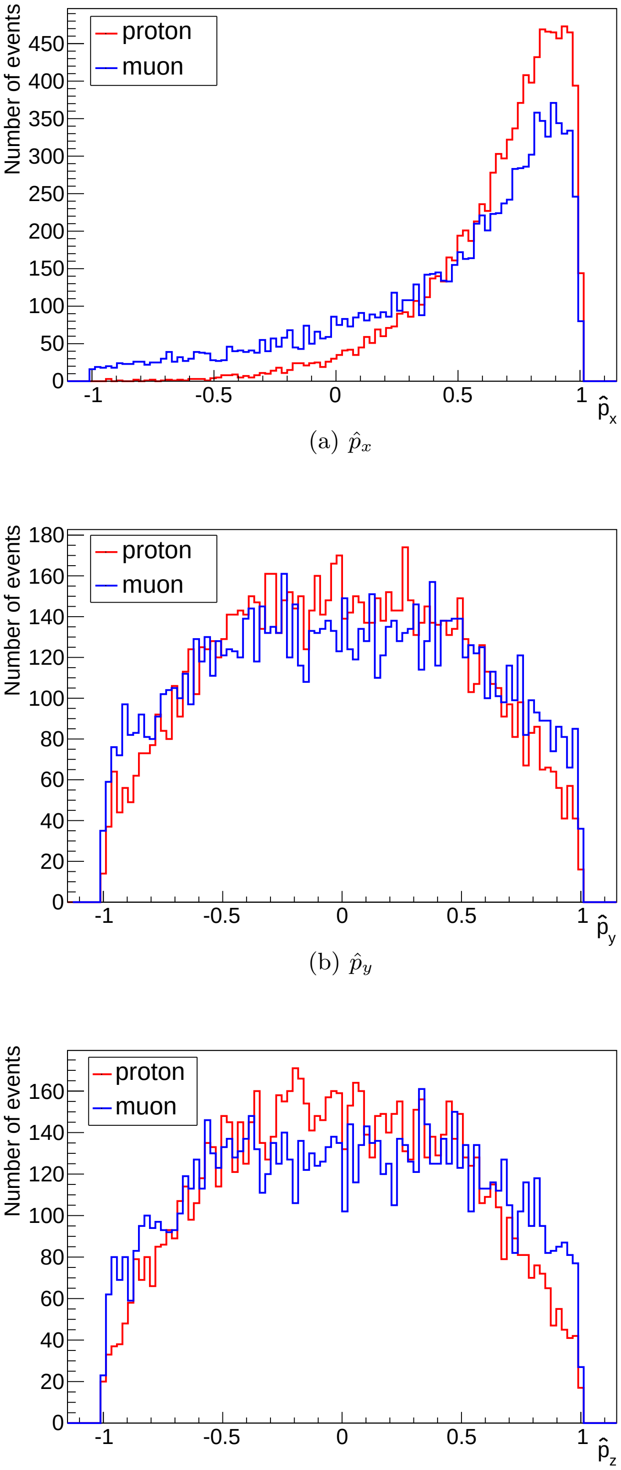
These interactions were generated with a muon neutrino beam in the $+x$ direction, so the distributions should be peaked in the forward x direction, and symmetric in y and z.

7.3.1 770 MeV $\nu_\mu \rightarrow \mu + p$ (CCQE) Interactions

Figure 7.5 shows the angular distribution of muons and protons in CCQE ν_μ interactions at 770 MeV, producing a $\mu + p$ final state. The distributions are almost identical, which does not permit a simple cut value to be selected for PID discrimination and would likely be problematic for a multivariate analysis using e.g. TMVA [81] to combine several discriminators. Separation of muons and protons from CCQE interactions at low energy (770 MeV) using angular information alone is, therefore, not practical.

7.3.2 770 MeV $\nu_\mu \rightarrow \mu + p + \pi^+$ (CC1π) Interactions

Figure 7.6 shows the angular distribution of muons, protons and charged pions in CC1π ν_μ interactions at 770 MeV, producing a $\mu + p + \pi^+$ final state. As with the CCQE case, the distributions are too similar to permit a simple cut value for discrimination, but the $\hat{p}_x$ variable shows the proton peaked much more strongly at $+1$ than the muon or pion, while $\hat{p}_y$ and $\hat{p}_z$ show a stronger proton peak at 0, i.e. the proton is produced with momentum mostly in the $+x$ direction. This could



(a) $\hat{p}_x$

(b) $\hat{p}_y$

(c) $\hat{p}_z$

Figure 7.5: Angular distribution of muons and protons in CCQE interactions at 770 MeV. See text for discussion.

	μ	p
Included	9636	214
Excluded	364	9786

Table 7.4: Composition of tracks after applying a cut on $\hat{p}_x > 0.90$ on 10^4 CCQE events from the Genie generator with a ν_μ at 4.5 GeV incident on liquid Argon. The cut selects muons with an efficiency of 96.4% and a purity of 97.8%.

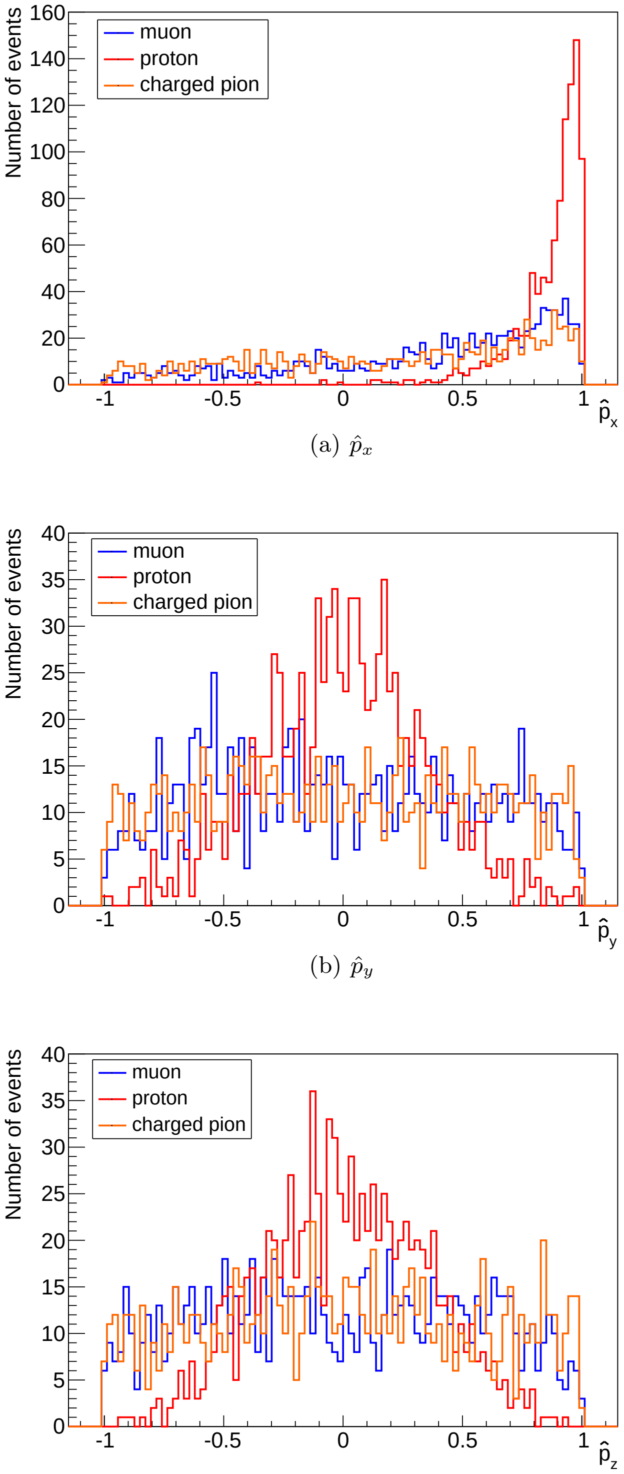
be of use in a multivariate analysis for discrimination, when combined with other PID variables. It should be noted that these variables are not uncorrelated, i.e. $|\hat{\mathbf{p}}| = |(\hat{p}_x, \hat{p}_y, \hat{p}_z)| = 1$, so if one variable is peaked at ±1, it is expected to find the other two close to zero.

7.3.3 4.5 GeV $\nu_\mu \to \mu + p$ (CCQE) Interactions

Figure 7.7 shows the angular distribution of muons and protons in CCQE ν_μ interactions at 4.5 GeV, producing a $\mu + p$ final state. In this case, the muon and proton distributions in the $\hat{p}_x$ variable are sufficiently distinct that a straightforward cut would have some discriminating power. Choosing a cut at $\hat{p}_x = 0.90$, it is possible to select muons with an efficiency of 96.4% and a purity of 97.8%. Table 7.4 shows the detailed results of this cut.

7.3.4 4.5 GeV $\nu_\mu \to \mu + p + \pi^+$ (CC1π) Interactions

Figure 7.8 shows the angular distribution of muons, protons and charged pions in CC1π ν_μ interactions at 4.5 GeV, producing a $\mu + p + \pi^+$ final state. The distributions overlap too much for a cut value to be an effective discriminator, but the muon distribution is much more strongly peaked in $\hat{p}_x$, which may be used as input to a multivariate analysis.

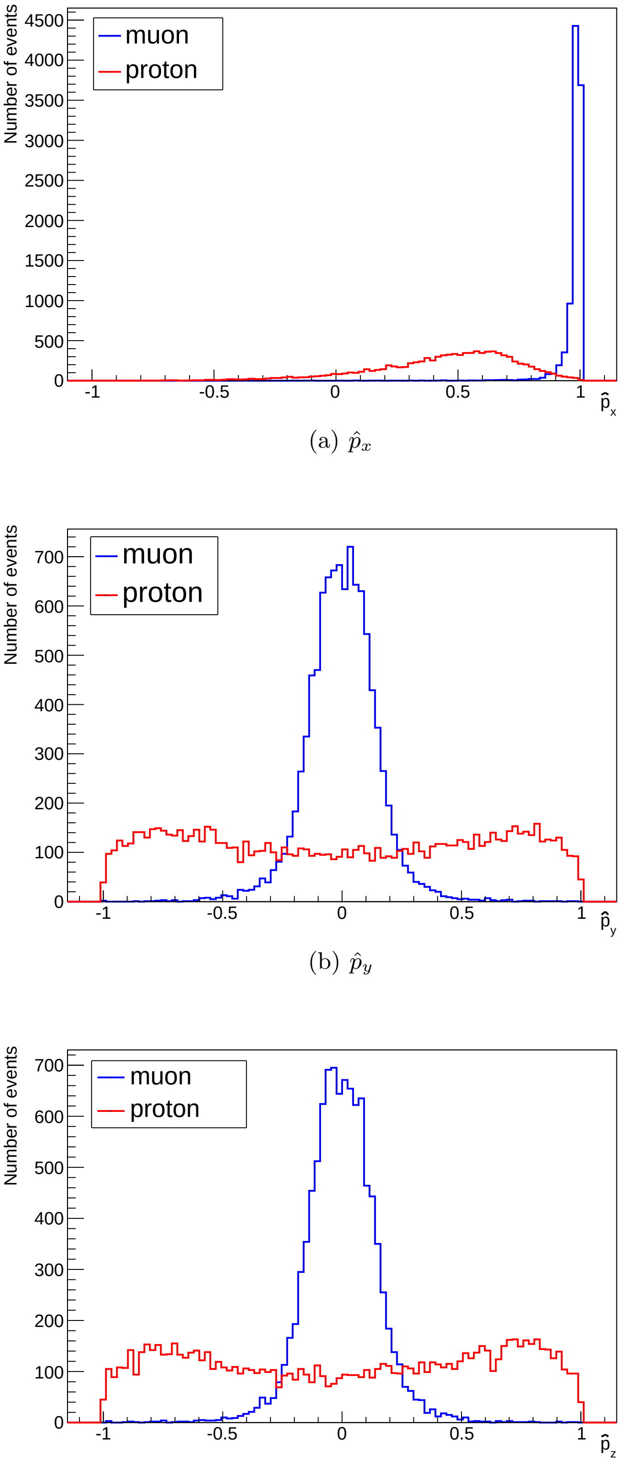


(a) $\hat{p}_x$

(b) $\hat{p}_y$

(c) $\hat{p}_z$

Figure 7.6: Angular distribution of muons, protons and charged pions in CC1π interactions at 770 MeV. See text for discussion.

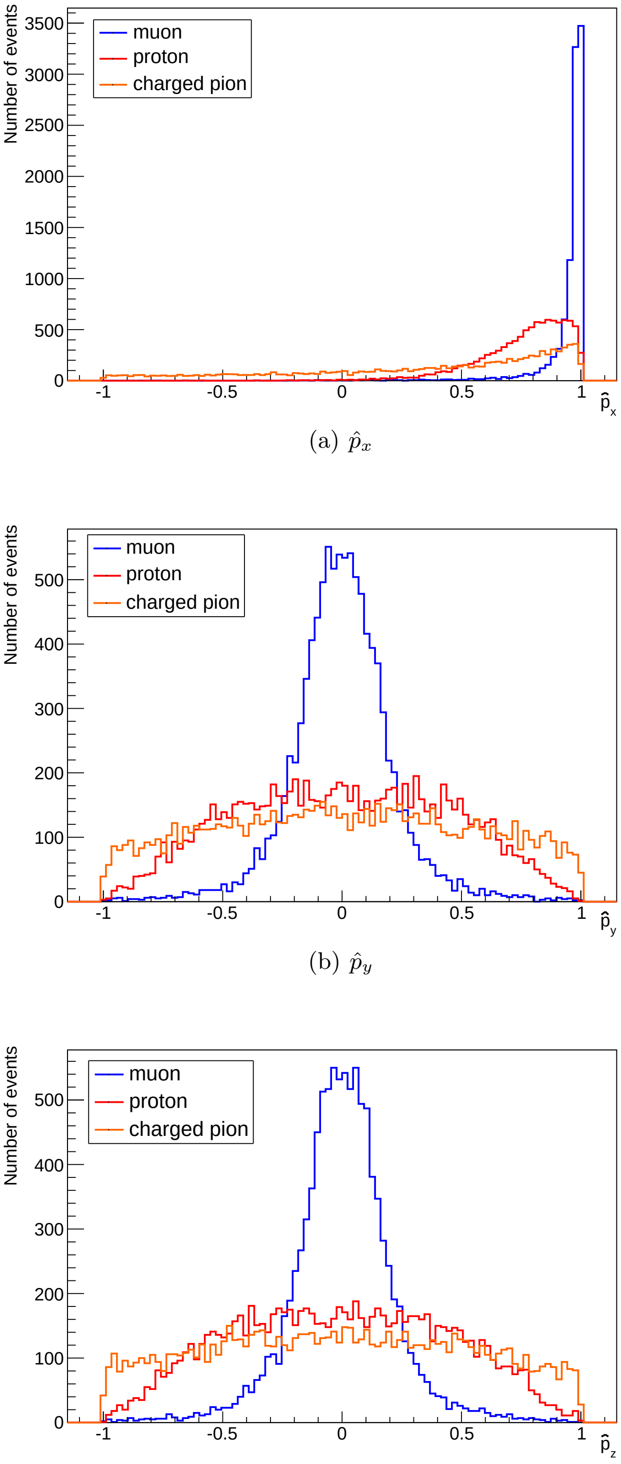


(a) $\hat{p}_x$

(b) $\hat{p}_y$

(c) $\hat{p}_z$

Figure 7.7: Angular distribution of muons and protons in CCQE interactions at 4.5 GeV. See text for discussion.



(a) $\hat{p}_x$

(b) $\hat{p}_y$

(c) $\hat{p}_z$

Figure 7.8: Angular distribution of muons, protons and charged pions in CC1π interactions at 4.5 GeV. See text for discussion.

Symbol	Description	Units
T	Kinetic energy of incident particle	MeV
$m_e c^2$	Electron mass ($\times c^2$)	0.511 MeV
r_e	Classical electron radius	2.818 fm
A	Atomic mass of absorber	$\mathrm{g\,mol^{-1}}$
Z	Atomic number of absorber	—
N_A	Avogadro's number	$6.022 \times 10^{23}\,\mathrm{mol^{-1}}$
K/A	$\left(4\pi N_A r_e^2 m_e c^2\right)/A$	$\mathrm{MeV\,g^{-1}\,cm^{-1}}$
z	Charge of incident particle	Units of electron charge
I	Mean excitation energy	eV
β	v/c	—
γ	$1/\left(\sqrt{1-\beta^2}\right)$	—
$\delta(\beta\gamma)$	Density effect correction to ionisation energy loss	—

Table 7.5: Definition and units for symbols used in the Bethe-Bloch equation. Adapted from [17].

7.4 Energy Loss (dE/dx)

The rate of energy loss for relativistic charged particles travelling through materials of intermediate atomic number is described by the Bethe-Bloch equation, accurate to a few % when $0.1 \leq \beta\gamma \leq 1000$ [17]:

$$-\left\langle \frac{dE}{dx} \right\rangle = Kz^2 \frac{Z}{A} \frac{1}{\beta^2} \left[\frac{1}{2} \ln \frac{2m_e c^2 \beta^2 \gamma^2 T_{max}}{I^2} - \beta^2 - \frac{\delta(\beta\gamma)}{2} \right] \tag{7.2}$$

where table 7.5 describes the symbols and their units, leading to $\frac{dE}{dx}$ values in $\mathrm{MeV\,g^{-1}\,cm^{-1}}$.

The rate of energy loss is often used as a discriminating variable for particle identification. In this section, the energy deposited in the Lamu simulation is shown for primary muons, protons and charged pions resulting from muon neutrino interactions at 0.77 GeV and 4.5 GeV. In each case, the dE/dx is represented by dQ/dx, the charge (energy) deposited in sections of simulated tracks that are 50 hits long, divided by the length of that segment as computed between the first and last hits included. In each case, the x axis represents dQ/dx in $\mathrm{MeV\,mm^{-1}}$ and is truncated at $1\,\mathrm{MeV\,mm^{-1}}$. This is done because samples were taken along the entire track length, giving rise to long tails in each distribution as particles slow down and deposit larger amounts of energy per unit length. These tails are truncated in the results to emphasise the area of interest for discrimination between particle species.

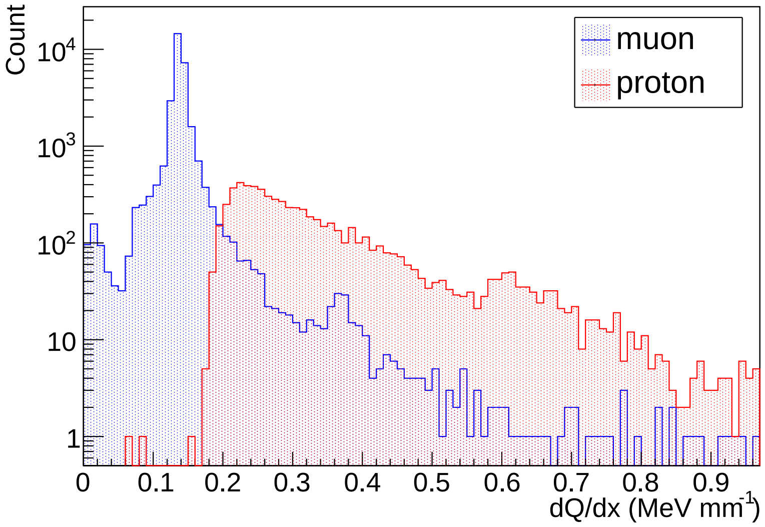


Figure 7.9: Distribution of energy loss rates (dE/dx) for protons and muons in CCQE interactions at 770 MeV. See text for discussion.

7.4.1 770 MeV $\nu_\mu \rightarrow \mu + p$ (CCQE) Interactions

Figure 7.9 shows the dE/dx distribution for muons and protons in CCQE ν_μ interactions at 770 MeV, producing a $\mu + p$ final state. The two distributions are partially separated and a cut value could be selected at $dE/dx \sim 0.18\,\mathrm{MeV\,mm}^{-1}$. The dE/dx information could also be fed into a multivariate analysis along with other PID variables.

Applying a cut at $dE/dx < 0.18\,\mathrm{MeV\,mm}^{-1}$ to the same data, this time restricting the dE/dx calculation to the first 50 hit segment of each track, shows that it is possible to select muons with 100% efficiency and 100% purity. This indicates that the first segment of each track has strong discriminating power in terms of particle identification. Table 7.6 shows the detailed results of this cut. Note that the difficulty will lie in determining whether a reconstructed track segment is at or near the beginning of a track. This requires a reconstruction algorithm with a high hit-level efficiency and most likely requires a vertex identification algorithm to provide assurance that the starting section of a track is selected for dE/dx computation.

7.4.2 770 MeV $\nu_\mu \rightarrow \mu + p + \pi^+$ (CC1π) Interactions

Figure 7.10 shows the dE/dx distribution of muons, protons and charged pions in CC1π ν_μ interactions at 770 MeV, producing a $\mu + p + \pi^+$ final state. In this scenario, the muons and charged pions have similar dE/dx profiles and separation

	μ	p
Included	1000	0
Excluded	0	1000

Table 7.6: Composition of tracks after applying a cut on $dE/dx < 0.18\,\mathrm{MeV\,mm^{-1}}$ to 1000 CCQE events from the Lamu simulation, in which protons and muons were produced from 770 MeV ν_μ interactions. The cut selects muons with an efficiency of 100% and a purity of 100%.

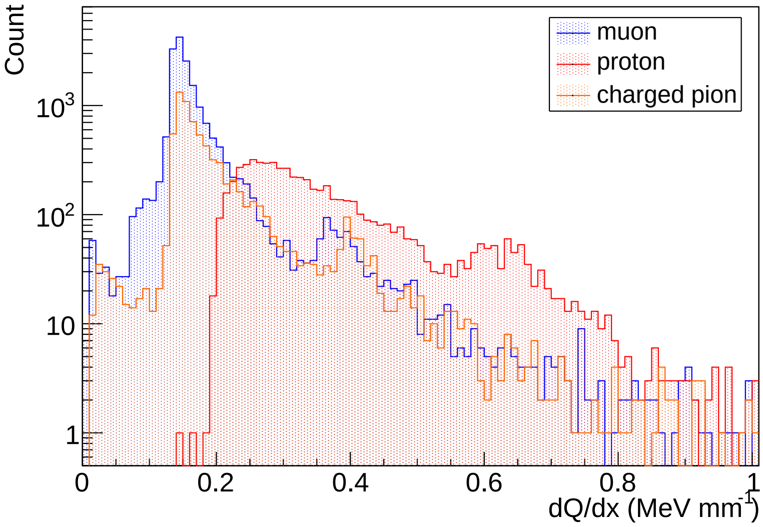


Figure 7.10: Distribution of energy loss rates (dE/dx) for protons, muons and charged pions in CC1π interactions at 770 MeV. See text for discussion.

using dE/dx alone is not possible. As with the other PID variables proposed in this chapter, it may be possible to combine several variables into a multivariate analysis to obtain improved discrimination.

7.4.3 4.5 GeV $\nu_\mu \rightarrow \mu + p$ (CCQE) Interactions

Figure 7.11 shows the dE/dx distribution of muons and protons in CCQE ν_μ interactions at 4.5 GeV, producing a $\mu + p$ final state. A cut at a low dE/dx value (~ 0.15) may provide some discrimination here, but there would be significant contamination. The distributions could be fed into a multivariate analysis for improved PID.

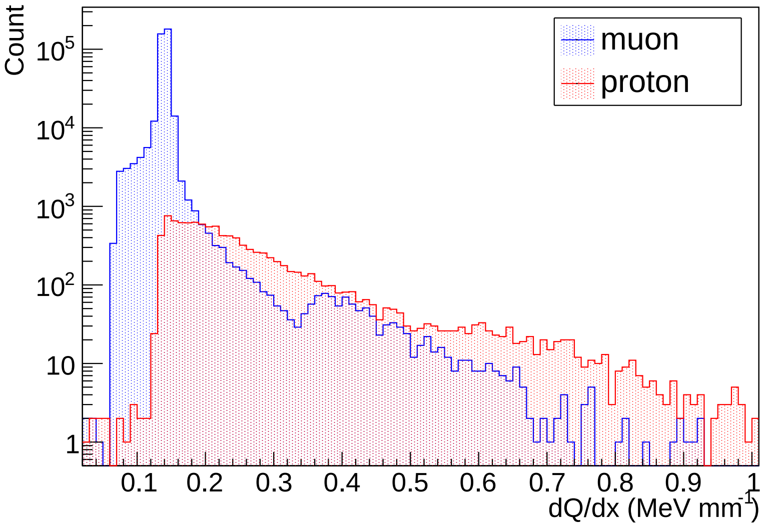


Figure 7.11: Distribution of energy loss rates (dE/dx) for protons and muons in CCQE interactions at 4.5 GeV. See text for discussion.

7.4.4 4.5 GeV $\nu_\mu \rightarrow \mu + p + \pi^+$ (CC1π) Interactions

Figure 7.12 shows the dE/dx distribution of muons, protons and charged pions in CC1π ν_μ interactions at 4.5 GeV, producing a $\mu + p + \pi^+$ final state. As with the low energy CC1π interactions, separation of particle species using dE/dx alone is not a realistic proposal, however more advanced PID techniques may still be able to make use of the dE/dx information.

7.5 Conclusions

It is clear from the track distributions presented above that for the low energy $\mu + p$ events, and for the high energy $\mu + p$ and $\mu + p + \pi^+$ events, a simple cut on the minimum track length is sufficient to select muon tracks from an event. The low energy $\mu + p + \pi^+$ events are an exception to this, and present a more complicated case for analysis, which may require further variables to be considered in order to extract muons with high purity. More traditional particle identification mechanisms, which make use of many variables and attempt to categorise events based on likelihood fits or the results of a neural network are not considered here, but may prove useful in cases such as the low energy $\mu + p + \pi^+$, where the interactions of the particles differ considerably, but are not well represented in the track length alone.

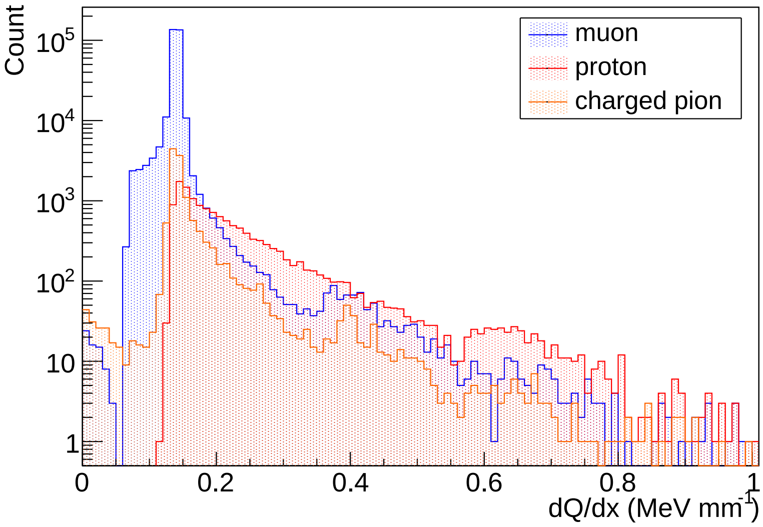


Figure 7.12: Distribution of energy loss rates (dE/dx) for protons, muons and charged pions in CC1π interactions at 4.5 GeV. See text for discussion.

For the remaining three event classes (high and low energy CCQE, high energy CC1π) this range cut should retain high efficiency, though it is worth noting that the studies in this chapter consider truth information, and that the effects of a reconstruction algorithm such as the cellular automaton of chapter 5 will necessarily reduce the efficiency of such a technique.

In addition to a range cut, it is possible to select muons in high energy CCQE events based on their x direction cosine, $\hat{p}_x = \frac{p_x}{\sqrt{p_x^2 + p_y^2 + p_z^2}}$. At low energies, muons can be selected in CCQE events by measuring the energy loss rate, dE/dx, in the first segment of the track. For a 50 hit segment length, muons can be selected with 100% efficiency by requiring $dE/dx < 0.18\,\mathrm{MeV\,mm^{-1}}$. The cut values are summarised in table 7.7. It should be noted that the dE/dx study uses data from the simulation, including a liquid Argon specific quenching factor, and is therefore more realistic in terms of its application to real detector data when compared to PID variables derived from truth information or Genie event generator data.

Class	Cut Value	Efficiency (%)	Purity (%)
CCQE 770 MeV	1000 hits	96.0 ± 0.6	94.4 ± 0.8
CCQE 770 MeV	$dE/dx < 0.18\,\mathrm{MeV\,mm^{-1}}$	100.0	100.0
CCQE 4.5 GeV	5000 hits	99.7 ± 0.1	99.9 ± 0.1
CCQE 4.5 GeV	$\hat{p}_x > 0.90$	96.4 ± 0.3	97.8 ± 0.2
CC1π 4.5 GeV	5000 hits	96.6 ± 0.2	99.5 ± 0.1

Table 7.7: Summary of the cuts for muon selection in different event classes, including the efficiency for selecting muons, and the resulting purity.

8 Analysis of Neutrino Interactions

8.1 Introduction

The goal of this thesis is to demonstrate a fully automated reconstruction chain for event data from LAr TPCs. Since the details of charge readout and position reconstruction will vary depending on the detector technology and geometry, as well as the equipment used, this study begins from the point at which three-dimensional hit data exists in a persistent on-disk format. This chapter uses the algorithms and techniques presented earlier to attempt to reconstruct physics information such as the muon energy from charged current ν_μ interactions at beam energies of 770 MeV and 4.5 GeV.

8.2 Charged Current $\nu_\mu \rightarrow \mu + p$ at 770 MeV

8.2.1 Event Selection Efficiency

A sample of 1000 events produced from charged current interactions of ν_μ at 770 MeV, yielding $\mu + p$ (only) final states were considered. These are the same events used for validation of algorithms earlier in this thesis. Of these 1000 events, 878 survive the initial requirement that the true proton track has more than 20 hits and undergo reconstruction.

Events surviving the initial cut undergo the following reconstruction steps:

1. Apply charge weighting using a weighting radius $R_w = 5\,\text{mm}$.

2. Apply the cellular automaton, using a maximum scattering angle of 30°.

3. Apply the cylinder-based merging algorithm, using a merging radius $R_m = 30\,\mathrm{mm}$ and allowing infinitely long cylinders.

4. Select events with two reconstructed clusters after merging.

5. Select events in which one cluster has > 1000 hits, and the other has < 1000 hits.

6. Associate the longer of the two clusters with the muon track.

7. Associate the shorter of the two clusters with the proton track.

A total of 420 events have only two tracks in the output after merging (events with more tracks in the output can potentially be recovered with improved merging algorithms, or by discarding delta electron tracks). Of these, 407 events have at least one output track containing over 1000 hits (of which, 10 events have both tracks with over 1000 hits). By comparing to truth information, it is found that In 341 events, there is one track with over 1000 hits, and one smaller track. In these events, the long track is identified from the truth information as corresponding to the muon, and the shorter track is identified from the truth information as corresponding to the proton.

Based on the numbers above, an analysis that required precisely two tracks, one of which had over 1000 hits, and the other with fewer than 1000 hits will recover 397 events from the sample of 1000, of which 341 will be correctly identified as $\mu + p$ events, with the tracks tagged with the right particle type. This corresponds to an efficiency of $397/1000 = 39.7\%$ for selecting $\mu + p$ events, with a purity of $341/397 = 85.9\%$. This analysis imposes the following requirements:

1. Events pass an initial proton range cut: length ≥ 20 hits.

2. Events are reconstructed (after merging) with only 2 tracks.

3. One of the tracks has > 1000 hits, one has < 1000 hits.

While the selection efficiency of this analysis is low (under 40%), the purity attained is high (over 85%), and as already mentioned it is possible to recover many more events with improvements to the reconstruction chain. Note that the selection efficiency here refers to the ability to select muon tracks from CCQE events, i.e. the efficiency represents the number of output events matching the above criteria, and the purity is a measure of how many of those events were correctly identified.

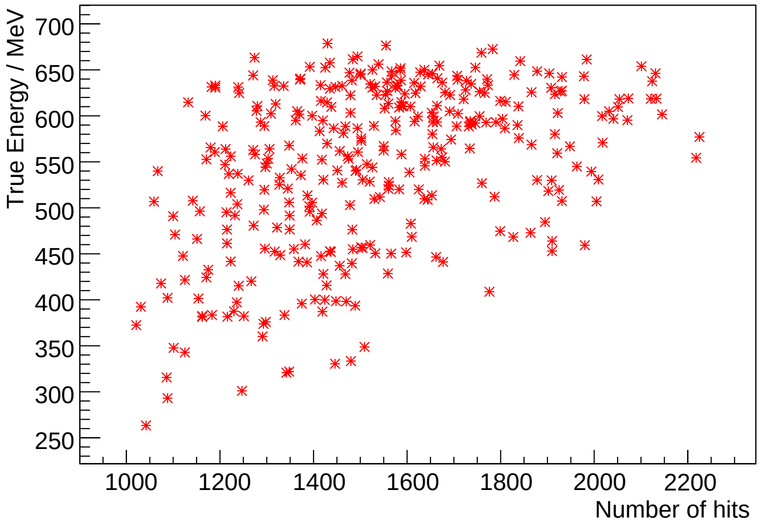


Figure 8.1: The relationship between the number of hits N in a muon cluster and the initial energy E_μ of the particle, in MeV. The broad distribution indicates that the number of hits is not a good estimator for the energy of the particle, and that calibrations based on this quantity cannot provide reliable energy estimates.

8.2.2 Muon Energy Reconstruction

Using Number of Hits

In an ideal scenario, the number of hits in a cluster should correspond approximately to the length of the cluster in mm. Since the range of a muon track varies with the initial energy of the muon, assuming minimum-ionising behaviour throughout, it should be possible to obtain a calibration for conversion between number of hits (or muon track length) and initial muon energy. In practice, the number of hits in an output cluster is a poor estimator for the initial muon energy, as demonstrated in figure 8.1. One possible reason for this is the loss of hits in the cellular automaton state of the reconstruction; the proportion of lost hits rises with track length, resulting in increasingly poor estimates. However, since the hit loss should occur throughout the track, it may be possible to use the length itself to estimate the energy.

Using Track Length

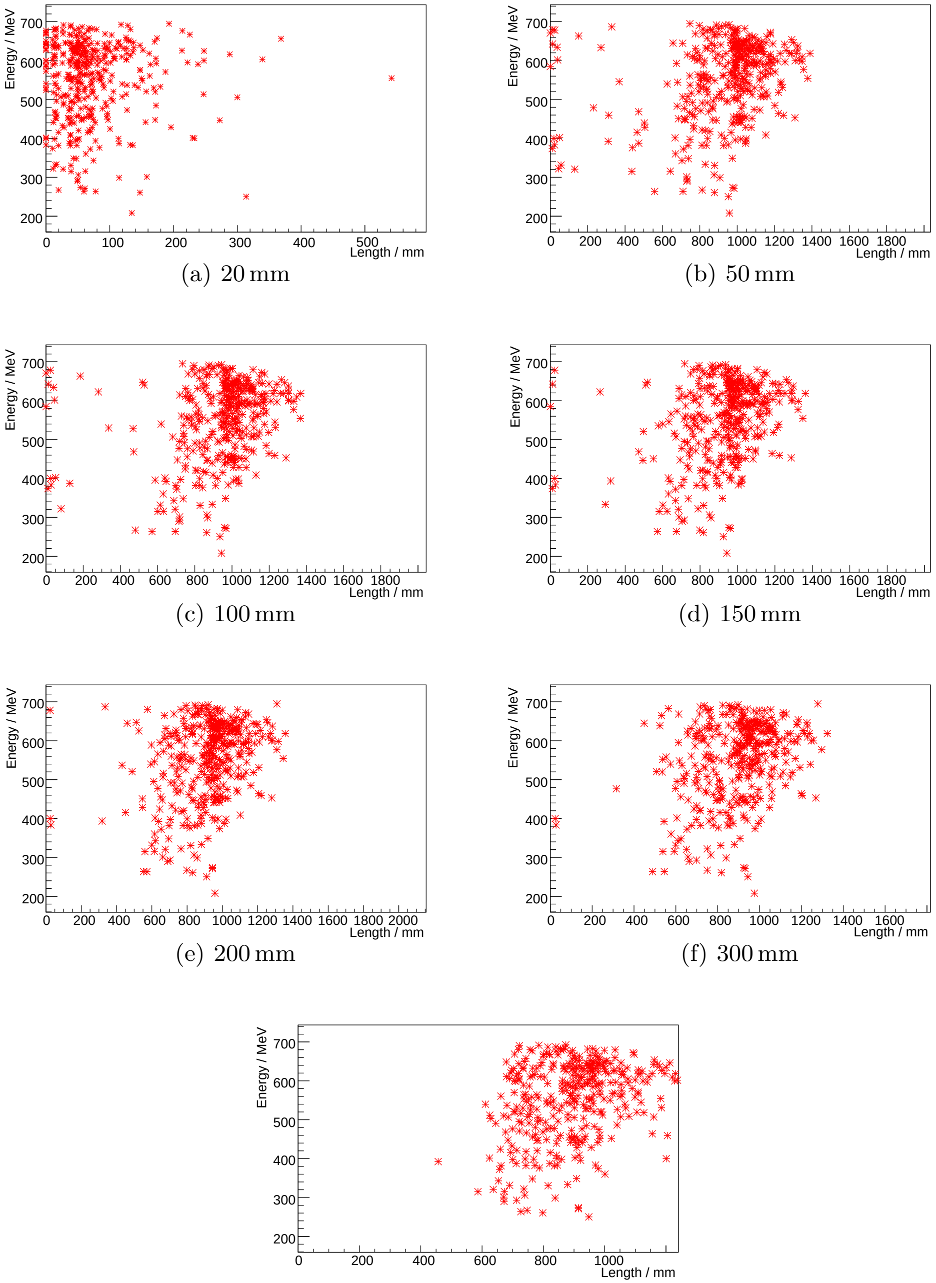
Calculating the length of the track is not as simple as taking the “smallest” and “largest” hits and determining the distance between them, since it is possible that the muon track may undergo significant scattering along its length, and can develop an appreciable curve. In order to determine the track length, an algorithm must run along the track in small steps, adding up the segment lengths as it goes. Such a mechanism can be easily implemented using the KDTree structure described in chapter 4.2, but this introduces another parameter; the radius used for the neighbour search, which becomes the maximum segment length.

The track length is calculated by taking the clustered set of hits, looking for an *extreme* hit, i.e. one end of the track, then searching for neighbours within some radius R. The furthest neighbour becomes the next seed point, and the distance between the two is added to an accumulator variable. This process is repeated, using principal components analysis to ensure that the discovered point is always moving away from the previous steps, until the other end of the track is reached. The sum of step lengths approximates the length of the entire track. The maximum segment length is determined by the value of R, and the results of applying this procedure to the 397 tracks being analysed here are shown in figure 8.2 for seven different maximum segment lengths.

Smaller segment lengths allow for a more accurate estimate of the track length in the presence of curvature, but are more susceptible to gaps in the reconstructed track causing early termination of the algorithm. This can be seen in the results for a 20 mm segment length, where most measurements have very small lengths. Longer segment lengths are more tolerant of gaps, but provide poor estimates when traversing curved tracks, as well as reducing the sensitivity to features on a scale smaller than the segment length. This can be seen across the set of graphs, with the cluster of points shifting rightward (i.e. towards longer estimates) with increasing segment length. The results in figure 8.2 indicate that, once again, the correlation between reconstructed track length and true muon energy is poor, and cannot be used to calibrate for energy reconstruction.

8.3 Conclusions

The analysis in this chapter presents one possible approach to the reconstruction of neutrino events in LAr TPC environments, using many of the tools and algorithms developed, described and characterised in the rest of this thesis. Although the final step of reconstructing the muon energy was not possible using the techniques applied above, there remain other options for reconstructing physics information, and some of the same procedures could be applied to obtain clusters and produce event selections with a very high purity (over 85%). The efficiency of such a selection



(a) 20 mm

(b) 50 mm

(c) 100 mm

(d) 150 mm

(e) 200 mm

(f) 300 mm

(g) 400 mm

Figure 8.2: Relationship between the measured length of a reconstructed track and the true energy of the corresponding muon, shown for segment lengths of 20 mm, 50 mm, 100 mm, 150 mm, 200 mm, 300 mm and 400 mm. See text for discussion.

is low, in part due to the very specific requirements that were imposed, i.e. precisely two reconstructed tracks, one with greater than 1000 hits, and one with fewer than 1000 hits. As has already been mentioned, the selection efficiency (under 40%) can be boosted by considering additional event classes, for instance the three-track events, with extra merging or further processing.

The techniques presented clearly provide building blocks for the clustering and selection of arbitrary track-like objects, and depending on the physics case, an analysis such as the one described in this chapter could be built from these algorithms, tailored to the specific requirements for identifying the kind of particles of interest. While the analysis presented focused on muons, and was ultimately unable to reconstruct the energy from the two techniques applied, the successful demonstration of clustering and selection of these events stands alone; such automated processing has not been fully demonstrated prior to the work of this thesis.

9
Conclusion

A number of algorithms and reconstruction techniques have been introduced in this thesis, and it is worth taking some time to summarize the important results of the characterisation of each.

Chapter 4 introduced a number of algorithms that can be used as building blocks for more complicated reconstruction procedures. In particular, the KDTree has been used to provide near-neighbour search in the cellular automaton of chapter 5 and in the length estimation algorithms of chapter 8. The KDTree is not a reconstruction algorithm in itself, but is demonstrably useful in the implementation of such algorithms.

Of particular note in chapter 4 is the three-dimensional feature detection, which was demonstrated to correctly identify the primary interaction vertex (as well as a number of other features such as the production sites of delta electrons, and track endpoints) in over 70% of charged current ν_μ interactions producing $\mu + p$ final states. The ability to identify these interest points with high efficiency and in three dimensions is a significant result. The technique could be further improved by extending the mathematical oeprations of the feature detection algorithm to three dimensions, rather than using projections as described in this thesis. Such an extension would take full advantage of the three-dimensional data and may result in more accurate determination of features of interest. However, this task relies on finding a suitable function to produce a scalar *response value* from the response matrix.

The cylinder merging algorithm presented merges hits that fall inside an infinitely long cylinder around a seed track. This is a successful strategy for very sparse events, but with higher track multiplicities, it is possible to have intersecting cylinders from key tracks. In such a situation, stealing hits from one track is not desirable, so the algorithm could be trivially extended by considering only the regions near the end of a track. In events with a high track multiplicity, or with high curvature, such a merging strategy would produce better results than the infinite cylinder method used here.

In order to automate the application of these algorithms to perform reconstruction tasks, the *Latte Control* framework was developed and described in chapter 4.10. This allowed for the prototyping of reconstruction sequences, as well as automating reconstruction in a controllable and customisable way. This framework makes automated reconstruction of LAr TPC events possible by simply chaining together correctly configured components, which can perform any task from the initial reconstruction steps through to complex statistical analyses.

Chapter 5 introduced a clustering technique based on a cellular automaton. The performance of this algorithm was characterised by applying it to simple two-track straight line events before running on long two-track events from charged current ν_μ interctions. The final step was to apply the algorithm to the full set of hits from charged current ν_μ interactions producing either $\mu + p$ or $\mu + p + \pi^+$ final states. The resulting clusters were typically of very high purity (over 90%) and although some hit loss reduced the hit-level efficiency, many of these hits could be recovered by returning to the raw data and applying a merging algorithm to fill in any dropped hits.

The cellular automaton itself produces small clusters of high purity based on its ability to discriminate between sets of related hits according to how track-like they are, and according to the directionality of those tracks. When applied to straight line two-track events with a large opening angle (i.e. over 140°) this ability to discriminate is diminished; applying feature detection to isolate the vertex and mask out hits in that region may help to improve the clustering in these cases.

Chapter 6 describes a Kalman filter for muon momentum reconstruction, as presented by the Icarus collaboration. An attempt to re-implement this filter and apply it to muon data from the *Lamu* simulation proved unsuccessful, with no reliable momentum measurements available. It is possible that with further tuning, the Kalman filter may be able to provide realistic momentum estimates from measurements that take into account the multiple scattering of particles in a dense medium, but this will always be an easier task to perform in the presence of a magnetic field, where a helical fit can be applied. Given the large size of planned liquid Argon detectors, a strong magnetic field would be prohibitively expensive, but weaker, non-uniform fields may be sufficient to perform this task. A magnetic field would also provide much needed charge discrimination, e.g. between μ^+ and

μ^-.

The cuts applied in chapter 7 are for the selection of muons, and apply only to data from the *Lamu* simulation, although the arguments used remain valid for any data source. They were presented primarily to allow for the continuation of an analysis based on selecting events by their clustered properties.

Finally, chapter 8 describes an attempt to reconstruct the energy of the muon in charged current ν_μ interactions at 0.77 GeV, resulting in $\mu + p$ final states. Events were selected based on being reconstructed with precisely two tracks (after merging), and further refined by requiring that one track had over 1000 hits, and the other fewer than 1000 hits, as per the cuts described in chapter 7. This resulted in a selection where each event contains one proton track and one muon track, with efficiency 39.7% and purity 85.9%. As stated in the chapter itself, the efficiency is low due to the stringent requirements applied, and could be boosted by applying further merging to events with three or four tracks, based e.g. on eliminating tracks consistent with delta electrons or decay products, and by requiring that tracks considered to be protons and muons originate from the same vertex. In these cases, the information required to perform such an analysis is provided by the algorithms already presented, but there was insufficient time to extend this work to include higher track multiplicities.

A determination of the energy of the muon in each event could not be completed in this thesis, but by improving the merging algorithm to reduce hit loss, and considering the charge deposited (along with the dE/dx profile for a muon in liquid Argon) it should be possible to obtain energy measurements. Indeed, for a fully contained muon track, this should be a matter of summing the charge deposits and correcting for the Argon quenching factors described in chapter 3, as well as any losses in the reconstruction chain.

The algorithms presented here each stand alone as a significant contribution to the field of automated reconstruction in liquid Argon Time-Projection Chambers, and when combined through the *Latte Control* framework, they offer a flexible and customisable package for automated reconstruction and analysis, something that has not yet been successfully demonstrated by any running or planned LAr TPC experiment. It should be noted that the algorithms are written and presented in a manner that is invariant to the voxel size (readout resolution) and the effects of readout or digitization. These effects will require different parameters to be used for each algorithm, optimised for each detector or simulation for which they are used. The Latte framework provides a general purpose reconstruction toolkit, which can be applied to data from a number of experiments or simulations. In addition to the work presented in the main part of this thesis, appendix A describes other algorithms that I investigated in less detail, and appendix B describes the use of the Latte framework, and the cellular automaton algorithm in particular, by others in the research group at Warwick. This continued use of the algorithms and framework presented here demonstrates the value of both as collaborations continue to work towards a fully automated reconstruction chain.

A
Other Algorithms

This appendix describes a number of algorithms and approaches to track reconstruction that did not become a part of the final Latte framework, either because they did not work or because other approaches were more promising and time did not permit sufficiently detailed study of them. They are included here for completeness.

A.1 KDTree Track Finding

The KDTree data structure is described in chapter 4.2 as a data structure for rapid near neighbour search. The early stages of my work on track reconstruction focused on using the KDTree structure to find tracks through three-dimensional point clouds.

Starting at some extreme point, e.g. the outermost hit in the event, it is possible to use the KDTree to find neighbouring points and choose the closest. From this point, the procedure can be repeated, choosing another nearby point, and iterating until all points have been selected. In order to turn this into a track finding algorithm, it was necessary to impose some sense of direction, and to maintain a list of points that had already been visited.

1. Select a seed point, usually the most extreme (i.e. outermost) hit in the event.
2. Find neighbours within a small radius and choose the closest.
3. Mark both the seed point and the newly discovered neighbour as *allocated*, and place them in a list representing the track that has been discovered.
4. When multiple hits are discovered, choose the one that lies closest to the line through the existing track candidate.
5. Repeat the above with the newly discovered hit as the seed point, and search

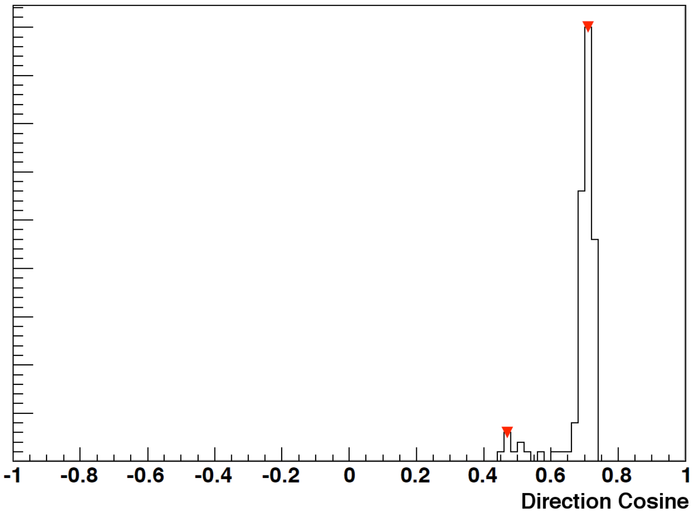


Figure A.1: Histogram of direction cosines (the y axis is an arbitrary value corresponding to the number of samples of the direction cosine) as the KDTree algorithm begins to round a corner between two tracks. The right-most peak corresponds to the first line followed, with a large number of samples. When the second peak begins to form, and can be detected by the TSpectrum peak search algorithm, the KDTree reconstruction algorithm is terminated, and a track candidate is reported.

only for neighbours that are not already allocated. A track candidate is completed when there are no remaining unallocated hits in the search region.

This procedure bears some resemblance to the cellular automaton algorithm of chapter 5, but the cellular automaton is capable of finding multiple track candidates at once, and is easy to implement as a parallel processing algorithm. In addition, the cellular automaton has a number of parameters built in that tailor it to selecting straight line tracks in a local environment, while allowing those tracks to exhibit some curvature on a much larger scale.

When the simple KDTree-based algorithm hits a vertex, it tends to follow the track around the vertex due to the very small scale on which it operates (it sees only immediate neighbours, and so cannot determine that it is following a track around a corner). To overcome this limitation, the direction cosines of the line made by the hits of the track were calculated at each step and added into a Root histogram. If the algorithm follows a single straight line, a peak begins to build for the direction cosines corresponding to that line. As soon as the algorithm begins to follow a second straight line (i.e. it has rounded the corner) a second peak will form. Figure A.1 shows a direction cosine histogram in which a second peak is beginning to form, indicating that the algorithm has started to follow a second straight line.

Using this information, the algorithm was terminated when a second peak was seen in the histogram of direction cosines, determined by running a search using the Root TSpectrum one-dimensional peak search algorithm. At this point, the last few hits of the track candidate were marked as unallocated, until the direction cosine of the remaining hits in the track candidate had direction cosines corresponding to the first peak. The algorithm was then re-run to attempt to pick up the second track.

While this algorithm was capable of finding straight line structures, the method described above for detecting corners was insufficient to prevent contamination of track candidates with hits from another particle, and the cellular automaton proved to be faster and more efficient, while employing some of the same techniques. The KDTree algorithm was able to reconstruct a limited number of toy tracks from the *TrackGen* simulation, but was never tested against the *Lamu* simulated neutrino events.

A.2 The Hough Transform

The Hough transform is a method for finding straight lines from a set of points $p_i = (x_i, y_i)$ in the xy-plane [82]. Each point (x_i, y_i) is transformed into a sinusoidal curve $r(\theta)$ as follows:

$$r(\theta) = x_i \cos\theta + y_i \sin\theta \tag{A.1}$$

A two-dimensional histogram is built up of the values of $r(\theta)$ for each point p in the event, and for θ sampled over the range $(0, \pi)$. Points lying on the same straight line are described by the same parameters r, θ and a peak builds up at each such set of parameters that corresponds to a line in the original event. In this way, the process of finding straight line structures in an image (or event) is reduced to the task of finding peaks in a two-dimensional histogram. Figure A.2 shows one such histogram for a toy event with two tracks. Note that due to the periodic nature of the sine and cosine functions, a peak at a value of $\theta = 0$ will be duplicated at $\theta = \pi$; this means that the Hough transform yields lines that extend infinitely in both directions, rather than vectors that would correspond more closely to the accepted idea of a track in particle physics.

It should be noted, however, that the Hough transform provides only the parameters of a line; it does not perform clustering, i.e. the allocation of hits to track candidates. This must be done after running the Hough transform by finding any hits that lie close to the line and allocating them to a cluster. Furthermore, the

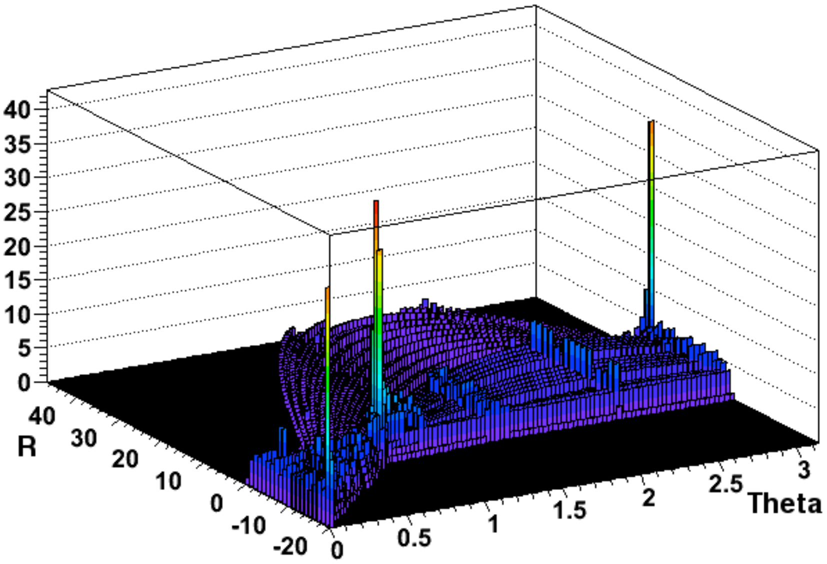


Figure A.2: Example of a histogram produced when the Hough transform is applied to a toy event containing two straight line tracks. Peaks correspond to the values of r and θ that parametrise the lines. See the main text for discussion.

Hough transform yields infinitely long lines; there is no indication of where in an event a track may start or finish. Due to these limitations, the Hough transform was considered to be insufficient for the purposes of track reconstruction in Liquid Argon, where knowledge of the start and end points of tracks is critical to the analysis. In particular, in the absence of a well-defined primary interaction vertex, the Hough transform provides insufficient information to correctly and reliably reconstruct the tracks of a neutrino event.

A.3 Graph Clustering Algorithms

One approach to extracting information from a 3D point cloud that was considered was to build a graph (in the mathematical sense of a connected network of nodes with possibly weighted edges) where each node was a point in the data set, and each edge was weighted by the distance between the nodes it connects.

Given such a graph, it is possible to analyse the data in a number of ways, including determining the shortest path between two nodes using standard shortest path algorithms, for example Dijkstra's algorithm [83]. Using the networkx[1] Python

[1] http://networkx.github.io/

library for graph manipulation, this could be trivially applied to data from the *TrackGen* or *Latte* simulations.

Given two extreme points in the event (e.g. a point close to the origin, and a point near the edge of the event) the shortest path algorithm provides a preliminary clustering of a track, containing those hits that lie on the shortest path between the ends of the track, however it is strongly dependent on a good seed point. Use of this approach in conjunction with the feature detection algorithms outlined in chapter 4.5 may give reasonable seed points for this technique. There still remains the issue of assigning hits surrounding the core to the track that was found, and the cellular automaton once again outperforms in this area.

No substantial analysis of this approach was performed, in part because the cellular automaton provided a promising approach to track reconstruction, and because there are many possible algorithms that could be run against a network or graph of spatial data, and it was not clear which possibilities would give the best results. The use of this technique to cluster or classify showers in a liquid Argon volume was briefly considered, but again no significant characterisation of the properties or mechanics of such an algorithm was performed.

B

Continued Use of the Latte Framework

Following on from the work in this thesis, the Latte framework has seen continued use by others at the University of Warwick. This chapter describes some of these developments and puts them in the context of the work presented in this thesis.

B.1 Cellular Automaton

The characterisation and assessment of the CA algorithm for track finding in liquid Argon has continued, resulting in a draft paper accepted for publication in Eur. Phys. J. C [2].

In the paper, further optimisation of the CA algorithm for muon and proton tracks in liquid Argon yields the set of parameter values shown in table B.1. The CA algorithm itself remains identical to the implementation described in this thesis, but the merging strategy was changed to use information local to the endpoints of each cluster, giving rise to the *stitching* parameters listed.

The CA was applied to two datasets from 0.77 GeV μ_ν interactions, each containing 1000 simulated events. The first set had $\mu + p$ (CCQE) final states, while the second set had $\mu + p + \pi^+$ (CC1π) final states. The reconstruction efficiency and purity for each particle species of interest are shown in table B.2 for the CCQE

Smoothing	CA	CA search radius	Stitching	Stitching	Min. cluster
radius	angle	/ max. radius	angle	num. end-points	hit count
6 mm	20°	3 mm / 12 mm	30°	5	5

Table B.1: Summary of the optimal reconstruction parameters for the CA algorithm applied to CCQE events. Data from [2].

Num. recon. clusters	Muon efficiency (%)	Muon purity (%)	Proton efficiency (%)	Proton Purity (%)
$N = 2$	98.3	92.4	99.0	95.9
$N > 1$	92.7	92.3	98.8	95.5
$N > 0$	93.1	91.7	98.8	95.1

Table B.2: Mean efficiency and purity measured at the hit level for muons and protons from 0.77 GeV ν_μ interactions. A proton range cut at 25 hits was applied based on the Monte Carlo truth data. Data from [2].

Num. recon. clusters	Muon efficiency (%)	Muon purity (%)	Proton efficiency (%)	Proton Purity (%)	Pion efficiency (%)	Pion purity (%)
$N > 2$	96.9	86.9	98.3	90.1	92.6	85.5
$N > 0$	97.3	85.6	98.5	89.0	93.1	84.1

Table B.3: Mean efficiency and purity measured at the hit level for muons, protons and pions from 0.77 GeV ν_μ interactions. A proton range cut at 25 hits was applied based on the Monte Carlo truth data. Data from [2].

events, and table B.3 for the CC1π events. In both cases a proton range cut was applied based on the truth data, requiring protons to have more than 25 hits, equivalent to a 2.5 cm track, or a 10 MeV minimum ionising particle travelling through liquid Argon.

The paper also evaluates the application of the CA algorithm to track-shower discrimination, using data from the clusters found by the CA as input to a Boosted Decision Tree (BDT) based on the TMVA [81] package. The three input variables are derived from the CA output as follows:

1. Ratio of transverse to logitudinal extent of the cluster
2. Ratio of Coulomb energy[1] to the longitudinal extent of the cluster
3. Number of clusters produced by the CA; important for low energy, where the geometric variables do not provide good discrimination between muon tracks and electron showers.

The BDT was trained with a signal consisting of electron showers with energies between 10 MeV and 3 GeV and a background consisting of muons in the same energy range. The optimal cut was defined to be the one that maximises the

[1] The Coulomb energy of a cluster is defined here as the sum over all hits of the charge in each hit divided by the distance from the hit to the main principal component axis of the cluster.

ratio of signal to total event rate. This yields a 97.3% signal efficiency (i.e. a 3.7% contamination) for discriminating electron showers from muon tracks, across all of the energies sampled.

These results demonstrate that the cellular automaton algorithm is fit for its original purpose, but can also be applied to scenarios for which it was never designed. Therefore, as a track finder and as a general purpose clustering tool, the CA is a valuable addition to the toolkit provided by the Latte framework.

B.2 Local Principal Curves

Local Principal Curves (LPC) are an extension of Principal Components Analysis (PCA) in which a curve is found and followed by evaluating the principal components of hits in a local sphere, moving along the main principal component axis, then repeating the procedure [84].

The Warwick group are currently investigating the possibility of using Local Principal Curves for track finding, shower finding and other reconstruction tasks in liquid Argon. The LPC code has been interfaced to the Latte framework through a control module allowing it to be operated in a reconstruction pipeline, and opening up the entire toolbox of algorithms, input and output formats, simulations and facilities provided in Latte. This has allowed for more rapid development of the techniques being explored with the LPC, and provided sample data in the form of simulated events from TrackGen and Lamu, both of which can be read via the Latte framework. A paper describing the results of these investigations is currently in preparation.

B.3 Pandora Particle Flow Analysis

The Pandora PFA (Particle Flow Analysis) algorithm [85] has been successfully applied to neutrino events in liquid Argon detectors by members of the Warwick group. Although the investigation so far has used Pandora as a standalone tool, there are plans to integrate Pandora with the Latte framework. Pandora is written in C++, while the Latte framework is written in Python. Using a library like `Boost.Python`[2], it is possible to integrate the two, demonstrating the utility of Latte as a general purpose control framework for almost any algorithm of interest.

[2] `http://www.boost.org/doc/libs/1_54_0/libs/python/doc/index.html`

Bibliography

[1] G. Rutter, M. Richards, A. J. Bennieston, and Y. A. Ramachers. Point-source reconstruction with a sparse light-sensor array for optical TPC readout. *JINST*, 6:P07006, 2011.

[2] J. J. Back, G. J. Barker, A. J. Bennieston, S. B. Boyd, B. Morgan, and Y. A. Ramachers. Implementation of a cellular automaton algorithm for neutrino interaction reconstruction in a liquid argon volume. 2013. Accepted for publication in Eur. Phys. J. C.

[3] E. Fermi. Versuch einer theorie der β-strahlen. *Z. Physik*, 88:161–177, 1934.

[4] F. L. Wilson. Fermi's theory of beta decay. *Am. J. Phys.*, 36(12):1150–1160, 1968.

[5] F. Reines, C. L. Cowan, F. B. Harrison, H. W. Kruse, and A. D. McGuire. Detection of the Free Neutrino: a Confirmation. *Science*, 124(3212):103–104, 1956.

[6] J. N. Bahcall. Solar Neutrinos. I. Theoretical. *Phys. Rev. Lett.*, 12(11):300–302, 1964.

[7] R. Davis Jr. Solar Neutrinos. II. Experimental. *Phys. Rev. Lett.*, 12(11):303–305, 1964.

[8] G. Danby et al. Observation of high-energy neutrino reactions and the existence of two kinds of neutrinos. *Phys. Rev. Lett.*, 9(1):36–44, 1962.

[9] B. Pontecorvo. Inverse beta processes and non-conservation of lepton charge. *Sov. Phys. JETP*, 7(34):172–173, 1958.

[10] B. Pontecorvo. Neutrino experiments and the problem of conservation of leptonic charge. *Sov. Phys. JETP*, 26(5):984–988, 1968.

[11] K. S. Hirata et al. Observation of ^{8}B solar neutrinos in the Kamiokande-II detector. *Phys. Rev. Lett.*, 63(1):16–19, 1989.

[12] Q. R. Ahmad et al. Measurement of the rate of $\nu_e + d \to p + p + e^-$ interactions produced by ^{8}B solar neutrinos at the Sudbury Neutrino Observatory. *Phys. Rev. Lett.*, 87(7):071301 1–6, 2001.

[13] Z. Maki, M. Nakagawa, and S. Sakata. Remarks on the unified model of elementary particles. *Prog. Theor. Phys.*, 28(5):870–880, 1962.

[14] K. Zuber. *Neutrino Physics.* Taylor & Francis Group, 2004.

[15] D. H. Perkins. *Introduction to High Energy Physics*, page 317. Cambridge University Press, 2000.

[16] A. D. Sakharov. Violation of $\mathcal{CP}$ Invariance, $\mathcal{C}$ Asymmetry, and Baryon Asymmetry of the Universe. *J. Exp. Theor. Phys. Lett.*, 5:24, 1967.

[17] Particle Data Group: J. Beringer et al. Review of particle physics. *Phys. Rev. D*, 86(01001), 2012.

[18] A. Riotto and M. Trodden. Recent progress in baryogenesis. *arXiv:hep-ph/9901362v2*, 1999.

[19] B. Kayser. On the quantum mechanics of neutrino oscillation. *Phys. Rev. D*, 24(1):110–116, 1981.

[20] C. Giunti and C. W. Kim. *Fundamentals of Neutrino Physics and Astrophysics.* Oxford University Press, 2007.

[21] Particle Data Group: K. Nakamura et al. Review of particle physics. *J. Phys. G: Nucl. Part. Phys.*, 37(075021), 2010.

[22] SAGE Collaboration: J. N. Abdurashitov et al. Measurement of the solar neutrino capture rate by the russian-american gallium solar neutrino experiment during one half of the 22-year cycle of solar activity. *J. Exp. Theor. Phys.*, 95: 181–193, 2002.

[23] W. Hampel et al. GALLEX solar neutrino observations: Results for GALLEX IV. *Phys. Lett. B*, 447:127–133, 1999.

[24] GNO Collaboration: M. Altmann et al. Complete results for five years of GNO solar neutrino observations. *Phys. Lett. B*, 616:174, 2005.

[25] M. Mezzetto and T. Schwetz. θ_{13}: Phenomenology, present status and prospect. *J. Phys. G: Nucl. Part. Phys.*, 37(103001):1–46, 2010.

[26] G. Bellini et al. Absence of a day-night asymmetry in the ^{7}Be solar neutrino rate in borexino. *arXiv:1104.2150v2*, 2012.

[27] Super-Kamiokande: J. Hosaka et al. Three flavor neutrino oscillation analysis of atmospheric neutrinos in Super-Kamiokande. *Phys. Rev. D*, 74:032002, 2006.

[28] K2K: M. H. Ahn et al. Measurement of neutrino oscillation by the K2K experiment. *Phys. Rev. D*, 74:072003, 2006.

[29] MINOS Collaboration: P. Adamson et al. Measurement of neutrino oscillations with the minos detectors in the numi beam. *Phys. Rev. Lett.*, 101:131802, 2008.

[30] K. Abe et al. Indication of electron neutrino appearance from an accelerator-produced off-axis muon neutrino beam. *Phys. Rev. Lett.*, 107(4):041801–041809, 2011.

[31] A. An et al. Observation of electron-antineutrino disappearance at Daya Bay. *arXiv:1203.1669*, 2012.

[32] RENO Collaboration: J. K. Ahn et al. Observation of reactor electron antineutrino disappearance in the RENO experiment. *arXiv:1204.0626v1*, 2012.

[33] P. F. Harrison and W. G. Scott. The simplest neutrino mass matrix. *Phys. Lett. B*, 594:324, 2004.

[34] H. Onsumi (on behalf of the NEMO and SuperNEMO Collaboration). SuperNEMO project. *J. Phys.: Conf. Ser.*, 120(052054):1–3, 2008.

[35] C. Kraus and S. J. Peeters. The rich neutrino programme of the SNO+ experiment. *Prog. Part. Nucl. Phys.*, 64(2):273–277, 2010.

[36] D. Green. *The physics of particle detectors*. Cambridge University Press, 2008.

[37] ATLAS Collaboration. ATLAS detector and physics performance technical design report. Technical report, 1999.

[38] E. Aprile, A. E. Bolotnikov, A. I. Bolozdynya, and T. Doke. *Noble gas detectors*. Wiley, 2006.

[39] The LAGUNA Consortium: D Angus et al. The LAGUNA design study — towards giant liquid based underground detectors for neutrino physics and astrophysics and proton decay searches. *arXiv:1001.0077v1*, 2009.

[40] C. Rubbia. The liquid-argon time projection chamber: a new concept for neutrino detectors. oai:cds.cern.ch:117852. Technical Report CERN-EP-INT-77-8, CERN, Geneva, 1977.

[41] J. Birks. *The theory and practice of scintillation counting.* Pergamon Press, 1964.

[42] S. Amoruso et al. Study of electron recombination in liquid Argon with the ICARUS TPC. *Nucl. Instr. Meth. A*, 523:275–286, 2004.

[43] S. Amerio et al. Design, construction and tests of the ICARUS T600 detector. *Nucl. Instr. Meth. A*, 527:329–410, 2004.

[44] B. Baibussinov et al. A new, very massive modular liquid argon imaging chamber to detect low energy off-axis neutrinos from the cngs beam. *Astropart. Phys.*, 29(3):174–187, 2008.

[45] D. B. Cline et al. LANNDDD—A massive liquid argon detector for proton decay, supernova and solar neutrino studies, and a neutrino factory detector. *Nucl. Instr. Meth. A*, 503:136–140, 2003.

[46] D. Y. Stewart et al. Modelling electroluminescence in liquid argon. *JINST*, 5: P10005, 2010.

[47] P. K. Lightfoot et al. Optical readout tracking detector concept using secondary scintillation from liquid argon generated by a thick gas electron multiplier. *JINST*, 4:P04002, 2009.

[48] H. O. Anger. Scintillation camera. *Rev. Sci. Instrum.*, 29:27, 1958.

[49] S. Agostinelli et al. Geant4—a simulation toolkit. *Nucl. Instr. Meth. A*, 506 (3):250–303, 2003.

[50] R. Brun and F. Rademakers. ROOT – an object oriented data analysis framework. *Nucl. Instr. Meth. A*, 389:81–86, 1996.

[51] C. Ferguson. General purpose source particle module for Geant4/SPARSET, Technical Note, 2000.

[52] `http://www.sqlite.org/`.

[53] C. Andreopoulos et al. The GENIE Neutrino Monte Carlo Generator. *Nucl. Instr. Meth. A*, 614:87–104, 2010.

[54] C. Anderson et al. The ArgoNeuT detector in the NuMI low-energy beam line at Fermilab. *JINST*, 7:P10019, 2012.

[55] The MicroBooNE Collaboration. The MicroBooNE Technical Design Report. Fermilab TDR, 2012.

[56] The LBNE Collaboration. The LBNE Conceptual Design Report. 2012.

[57] E. Church. LArSoft. GLA2011: 2nd International Workshop towards the Giant Liquid Argon Charge Imaging Experiment, 2011.

[58] A. A. Aguilar-Arevalo et al. Unexplained excess of electron-like events from a 1 GeV neutrino beam. *Phys. Rev. Lett.*, 102:101802, 2009.

[59] A. Stahl et al. Expression of interest for a very long baseline neutrino oscillation experiment (LBNO). CERN-SPSC-2012-012, SPSC-EOI-007, 2012.

[60] A. Badertscher et al. First operation and drift field performance of a large area double phase LAr Electron Multiplier Time Projection Chamber with an immersed Greinacher high-voltage multiplier. *JINST*, 7:P08026, 2012.

[61] M. Antonello et al. Precise 3D track reconstruction algorithm for the ICARUS T600 liquid argon time projection chamber detector. *arXiv:1210.5089*, 2013.

[62] J. L. Bentley. Multidimensional binary search trees used for associative searching. *Communications of the ACM*, 18(9):509–517, 1975.

[63] A. W. Moore. *Efficient Memory-based Learning for Robot Control.* PhD thesis, Computer Laboratory, University of Cambridge, 1991. Technical Report No. 209.

[64] E. Jones, T. Oliphant, P. Peterson, et al. SciPy: Open source scientific tools for Python, 2001–. URL `http://www.scipy.org/`.

[65] M. Ester, H-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of 2nd International Conference on Knowledge Discovery and Data Mining (KDD-96)*, 1996.

[66] J. Spitz. Argoneut and the neutrino-argon charged current quasi-elastic cross section. *J. Phys.: Conf. Ser.*, 312:072017, 2011.

[67] M. Ankerst, M. Breunig, H-P. Kriegel, and J. Sander. OPTICS: Ordering Points To Identify the Clustering Structure. *SIGMOD 99*, pages 49–60, 1999.

[68] B. Morgan. Interest point detection for reconstruction in high granularity tracking detectors. *JINST*, 5(P07006), 2010.

[69] J. von Neumann and A. W. Burks. *Theory of self-reproducing automata.* University of Illinois Press, 1966.

[70] M. Gardner. Mathematical games, the fantastic combinations of John Conway's new solitaire game ”life”. *Scientific American*, October:120–123, 1970.

[71] A. Glazov, I. Kisel, E. Konotopskaya, and G. Ososkov. Filtering tracks in discrete detectors using a cellular automaton. *Nucl. Instr. Meth. A*, 329:262–268, 1992.

[72] M. Casolino and P. Picozza. A cellular automaton to filter events in a high energy physics discrete calorimeter. *Nucl. Instr. Meth. A*, 364:516–523, 1995.

[73] I. Kisel et al. Cellular automaton and elastic net for event reconstruction in the NEMO-2 experiment. *Nucl. Instr. Meth. A*, 387:433–442, 1997.

[74] I. Abt, D. Emeliyanov, I. Kisel, and S. Masciocchi. CATS: a cellular automaton for tracking in silicon for the HERA-B vertex detector. *Nucl. Instr. Meth. A*, 489:389–405, 2002.

[75] H. Maesaka. *Evidence for muon neutrino oscillation in an accelerator-based experiment.* PhD thesis, Department of Physics, Kyoto University, 2005.

[76] R. Frühwirth. Application of kalman filtering to track and vertex fitting. *Nucl. Instr. Meth. A*, 262:444, 1987.

[77] A. Ankowski et al. Measurement of through-going particle momentum by means of multiple scattering with the ICARUS T600 TPC. *arXiv:hep-ex/0606006v1*, 2006.

[78] Particle Data Group: S. Eidelman et al. *Phys. Lett. B*, 592(1), 2004.

[79] E. J. Wolin and L. L. Ho. *Nucl. Instr. Meth. A*, 329(493), 1993.

[80] J. J. Back, G. J. Barker, A. J. Bennieston, S. B. Boyd, B. Morgan, and Y. A. Ramachers. Electron-hadron shower discrimination in a liquid argon time projection chamber. *Eur. Phys. J. C*, 73:2369, 2013.

[81] P. Speckmayer, A. Hoecker, J. Stelzer, and H. Voss. *J. Phys.: Conf. Ser.*, 219 (032057), 2010.

[82] P. V. C. Hough. Machine analysis of bubble chamber pictures. *Proc. Int. Conf. High Energy Accelerators and Instrumentation*, 1959.

[83] E.W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1(1):269–271, 1959.

[84] J. Einbeck, J. Tutz, and L. Evers. Local principal curves. *Statistics and Computing*, 15(4):301–313, 2005.

[85] M. A. Thomson. *Nucl. Instr. Meth. A*, 611:25–40, 2009.