Fakultät für Physik und Astronomie

Ruprecht-Karls-Universität Heidelberg

Diplomarbeit

im Studiengang Physik

vorgelegt von

Roland Matthias Richter

aus München

November 2000

Implementierung eines Algorithmus für den LHCb Level-1 Vertex-Trigger

Die Diplomarbeit wurde von Roland Matthias Richter ausgeführt am

Kirchhoff-Institut für Physik

unter der Betreuung von

Herrn Prof. Dr. Volker Lindenstruth

Inhalt

Aktuelle Experimente der Hochenergiephysik zur Präzisionsmessung der CP-Verletzung anhand der Zerfälle von B-Mesonen aus hadronischen Reaktionen stellen besonders hohe Anforderungen an das Trigger-System, da nur ein sehr kleiner Bruchteil aller erzeugten Ereignisse für eine solche Untersuchung geeignet ist. In dieser Arbeit wird ein Algorithmus für den Level-1 Vertex-Trigger des am CERN geplanten LHCb Detektors entwickelt und innerhalb des Rahmenprogramms GAUDI implementiert, der Sekundärvertices mit signifikantem Abstand zum primären Wechselwirkungspunkt als Indiz für die Existenz interessanter B-Zerfälle erkennt und rekonstruiert. Die für die Berechnung benötigten Zeiten werden detailliert gemessen und die Rekonstruktions-Ergebnisse qualitativ untersucht.

Abstract

Current high-energy experiments for precision measurement of CP violation on the basis of B mesons from hadronic reactions make great demands on the trigger system. Only a very small fraction of all produced events qualifies for such analysis. In this thesis an algorithm for the Level-1 vertex trigger of the future LHCb detector at CERN is developed and implemented within the GAUDI framework. It recognizes and reconstructs secondary vertices with a significant distance from the primary interaction point. The existence of these is interpreted as an indication of the presence of interesting B decays in the event. The processing time is measured in great detail and the reconstruction results are qualitatively analyzed.

Erklärung:

Ich versichere, daß ich diese Arbeit selbständig verfaßt und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Heidelberg, den 1. November 2000

....

Inhaltsverzeichnis

Kapitel 1

Einleitung

Die Phänomene der Teilchenphysik werden derzeit durch das sogenannte Standardmodell beschrieben, das unter Physikern allgemein anerkannt ist und sich in seiner Anwendung großer Beliebtheit erfreut, da es die Messungen der Experimentalphysik gut beschreibt und durch solche bislang auch nicht widerlegt werden konnte. Nach dem Modell existieren drei unterschiedliche Sorten von Teilchen, welche im Rahmen dieser Theorie als elementar angesehen werden, also keine innere Struktur besitzen und nicht in angeregten Zuständen auftreten. Das sind zum einen Leptonen und Quarks mit Spin $^1/_2$ (Fermionen), die als Teilchen und Antiteilchen auftreten und die Bausteine der Materie darstellen, und zum anderen die Vektorbosonen mit Spin 1, die „Träger“ der verschiedenen Elementarkräfte. Von diesen gibt es die masselosen Photonen als Austauschteilchen der elektromagnetischen Wechselwirkung zwischen geladenen Teilchen, die sehr massiven W- und Z-Bosonen, die die schwache Wechselwirkung mit sehr geringer Reichweite zwischen allen Fermionen übertragen, und schließlich die ebenfalls masselosen Gluonen, durch deren Austausch die starke Wechselwirkung zwischen Quarks entsteht. Diese Kraft nimmt mit dem Abstand zu, weswegen Quarks nicht isoliert, sondern nur in gebundenen Zuständen nachgewiesen werden können. Diese werden als Hadronen bezeichnet und treten als Mesonen (Paare aus Quark und Antiquark) und Baryonen (drei Quarks oder drei Antiquarks) auf.

So gut das Standardmodell die meisten Geschehnisse im Quantenbereich auch bestätigen oder voraussagen kann, so gibt es doch einige Unzulänglichkeiten und offene Fragen. Beispielsweise wird die vierte bekannte Kraft, die Gravitation zwischen massebehafteten Teilchen, durch dieses Modell nicht beschrieben. Gegenstand der Bemühungen vieler aktueller Experimente der Hochenergiephysik sind jedoch zwei andere Fragen. Zum einen sagt die Theorie die Existenz eines schweren und neutralen Bosons mit Spin 0 voraus, des sogenannten Higgs-Bosons, das neben dem Photon und den W- und

Z-Bosonen als viertes Austauschteilchen der vereinigten elektroschwachen Kraft gefordert wird. Bislang konnten noch keine Higgs-Bosonen mit Sicherheit nachgewiesen werden, erste Anzeichen wurden jedoch kürzlich am LEP[1] bei einer Masse von etwa 114 GeV gefunden.

Die zweite Frage stellt sich bei der CP-Verletzung. Die CP-Invarianz, die für ruhende Teilchen und deren Antiteilchen identische Eigenschaften verlangt, wird in sehr seltenen Fällen gebrochen. Dies wurde bislang nur im System neutraler Kaonen beobachtet. Während die vom Standardmodell vorausgesagte CP-Verletzung das Verhalten von K^0-Zerfällen noch korrekt beschreibt, ist sie jedoch zu schwach um die im Kosmos beobachtete Asymmetrie von Materie und Antimaterie vollständig zu erklären. In der Theorie der elektroschwachen Wechselwirkung resultiert die CP-Verletzung aus den Eigenschaften der Cabibbo-Kobayashi-Maskawa (CKM) Matrix, die die Mischung zwischen den Eigenzuständen der Quarkmassen und denen der schwachen Wechselwirkung beschreibt und durch vier Parameter eindeutig definiert ist. In der Wolfenstein-Parametrisierung hat die Matrix dann folgende Gestalt:

$$\begin{pmatrix} V_{ud} & V_{us} & V_{ub} \\ V_{cd} & V_{cs} & V_{cb} \\ V_{td} & V_{ts} & V_{tb} \end{pmatrix} \simeq \begin{pmatrix} 1-\frac{\lambda^2}{2} & \lambda & A\lambda^3(\rho - i\eta) \\ -\lambda & 1-\frac{\lambda^2}{2} & A\lambda^2 \\ A\lambda^3(1-\rho-i\eta) & -A\lambda^2 & 1 \end{pmatrix} \tag{1.1}$$

Die Parameter A und λ sind auf wenige Prozent genau bekannt. Die Messung der Werte von ρ und η wird für gewöhnlich mit Dreiecken in der ρ/η-Ebene illustriert, die die Unitaritätsbedingungen für die CKM-Matrix beschreiben. Eines der beiden für die Untersuchung von B-Mesonen relevanten Dreiecke zeigt Abbildung 1.1, wo der durch bisherige Messungen erlaubte Bereich markiert ist. Anhand von bekannten CP-Asymmetrien in Teilchenzerfällen können die Winkel dieser Dreiecke direkt bestimmt und mit den indirekt ermittelten Werten verglichen werden. Sollte hier keine Übereinstimmung festgestellt werden, wäre dies ein Hinweis auf „neue“ Physik.

Präzisionsmessungen der CP-Verletzung sind demnach ein wichtiger Test des Standardmodells und bergen das Potential, Physik jenseits dieser Theorie zu finden. Die besten Möglichkeiten für eine solche Messung bieten die Systeme der neutralen Mesonen B_d^0 und B_s^0, da hier im Gegensatz zu den Kaonen eine Vielzahl von Endzuständen auftritt, die für die Bestimmung von CP-Asymmetrien verwendet werden können. Das Problem ist allerdings, daß diese Zerfallskanäle nur in sehr kleinen Verzweigungsverhältnissen auftreten (Tabelle 1.1). Geeignete Experimente sind daher auf die Produktion riesiger Mengen von B-Mesonen angewiesen. Die Experimente BABAR [6] am

[1] Speicherring für Elektronen und Positronen am CERN in Genf

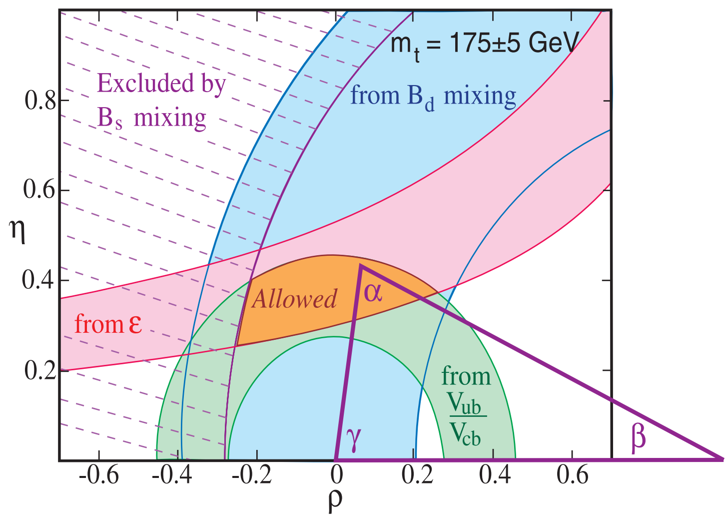


Abbildung 1.1: Dreieck in der ρ/η-Ebene, das die Unitaritätsbedingung $V_{tb}^{*}V_{td}+V_{cb}^{*}V_{cd}+V_{ub}^{*}V_{ud}=0$ in der Wolfenstein-Parametrisierung beschreibt. Die markierten Bereiche ergeben sich durch Messung der CP-Verletzung im System neutraler Kaonen (ϵ), der Oszillation von $\mathrm{B_d^0}$ ($\mathrm{B_d}$ mixing) und $\mathrm{B_s^0}$ Mesonen ($\mathrm{B_s}$ mixing) sowie der semileptonischen b-Zerfälle ohne Charme ($\frac{\mathrm{V_{ub}}}{\mathrm{V_{cb}}}$). [4]

SLAC in Stanford und BELLE [7] am KEK in Tsukuba werten die Zerfälle von B-Mesonen aus, die bei e^+e^--Kollisionen entstehen, müssen sich dabei aber mit Produktionsraten von etwa 10^8 B-Paaren pro Jahr begnügen, was die Experimente statistisch stark limitiert. Eine andere Möglichkeit bieten hadronische Reaktionen. Hier ist der Wirkungsquerschnitt für die Erzeugung von $\mathrm{b\bar{b}}$-Paaren wesentlich größer, allerdings kann die Kollisionsenergie nicht mehr durch den Speicherring auf die Resonanz für deren Erzeugungen eingestellt werden, wodurch der Anteil an unerwünschten Reaktionen viel größer ist und damit große Anforderungen an die Trigger gestellt werden. Dies wird derzeit nur bei HERA-B [8] am DESY in Hamburg durchgeführt, wo Protonen auf ein festes Target geschossen werden und die Asymmetrie im Zerfall $\mathrm{B^0} \rightarrow \mathrm{J}/\psi\,\mathrm{K_S}$ gemessen wird. Die B-Produktionsrate liegt hier bei etwa 10^9 pro Jahr. Auch der Protonenbeschleuniger am Fermilab in Chicago soll in naher Zukunft vergleichbare Raten erzielen, die dann von den Detektoren CDF [9], D0 [10] und BTeV [11] ausgewertet werden.

In diesem Bereich wird der für den Einsatz ab 2005 geplante Speicherring LHC (Large Hadron Collider) am Europäischen Kernforschungszentrum (CERN) in Genf neue Maßstäbe setzen. Hier werden Protonen mit Schwerpunktsenergien von bis zu 14 TeV kollidieren und man erwartet 10^{12} produzierte $\mathrm{b\bar{b}}$-Paare pro Jahr. Vier Experimente wurden für diesen Beschleu-

$B_d^0 \to \pi^+\pi^-$	$0,7 \cdot 10^{-5}$	$B_s^0 \to D_s^+K^-$	$3,1 \cdot 10^{-5}$
$B_d^0 \to J/\psi\, K_S$	$4,4 \cdot 10^{-4}$	$B_s^0 \to D_s^-K^+$	$2,0 \cdot 10^{-4}$
$B_d^0 \to D^0K^{*0}$	$1,3 \cdot 10^{-5}$	$B_s^0 \to J/\psi\, \phi$	$9,3 \cdot 10^{-4}$

Tabelle 1.1: Einige wichtige Zerfälle für die Messung der CP-Verletzung und deren Verzweigungsverhältnis. [17]

niger bewilligt und befinden sich derzeit in Entwicklung. Mit dem ALICE Detektor werden Kollisionen von Schwerionen untersucht, um die Physik stark wechselwirkender Materie bei extremen Energiedichten und das erwartete Quark-Gluon-Plasma zu studieren. Die Experimente ATLAS und CMS werden so ausgelegt, daß sie möglichst universell einsetzbar sind. Hier sollen Kollisionen zwischen Protonen untersucht werden, um beispielsweise das Grundverständnis von Materie und Kräften durch die Messung von Higgs-Bosonen zu verbessern.

Als viertes Experiment wird schließlich der LHCb Detektor gebaut, der auf die Präzisionsmessung der CP-Verletzung im System neutraler B-Mesonen und die Suche nach sehr seltenen oder gar vom Standardmodell ausgeschlossenen Zerfällen spezialisiert wird. Dieser Detektor muß in der Lage sein, im Betrieb aus einer Unzahl von Ereignissen die wenigen interessanten auszuwählen und die hohe Ereignisrate von 40 MHz zu bewältigen. Dies erfordert unter anderem ein besonders leistungstarkes Trigger-System. Der Entwicklung eines Teils davon widmet sich die vorliegende Arbeit, in deren Rahmen eine Software für den Vertex[2]-Trigger implementiert und mit Hilfe von simulierten Ereignissen getestet wurde.

Im weiteren Verlauf wird das LHCb Experiment kurz vorgestellt und die Motivation der Arbeit beschrieben. Sodann folgt eine Erklärung der Funktionsweise des Vertex-Trigger-Algorithmus, gefolgt von der Beschreibung der Implementierung im Rahmen der verwendeten Programmierumgebung. Anschließend werden die Ergebnisse im einzelnen vorgestellt und schließlich zusammengefaßt und diskutiert.

[2] Ort der Wechselwirkung von Teilchen in Hochenergie-Experimenten

Kapitel 2

Das Experiment LHCb

Das Experiment LHCb (A Large Hadron Collider Beauty Experiment) entsteht derzeit am Europäischen Kernforschungszentrum in Genf (CERN) und soll im Jahr 2005 in Betrieb gehen. Dort wird der e^+e^--Speicherring LEP und seine Detektoren entfernt und durch den neuen Large Hadron Collider (LHC) ersetzt. Das Ziel von LHCb ist es, Präzisionsmessungen der CP-Verletzung und seltener Zerfälle von B-Mesonen durchzuführen. Dazu müssen eine große Anzahl von $\mathrm{B_d^0}$ und $\mathrm{B_s^0}$ Mesonen erzeugt, erkannt und deren Enstehungs- und Zerfallsvertices sowie die verschiedenen Zerfallskanäle rekonstruiert werden können.

2.1 Der Speicherring

Der LHC ist ein Ringbeschleuniger, in dem Protonen oder Ionen bei bislang weltweit nicht erreichten Energien in entgegengesetzt laufenden Strahlen gespeichert und zur Kollision gebracht werden. Aus Kostengründen wird der LHC in dem bereits bestehenden Tunnel des LEP gebaut, der seit 1989 in Betrieb ist, und bekommt somit eine Länge von 27 km. Außerdem kann dadurch die bestehende Infrastruktur an Vorbeschleunigern genutzt werden. Abbildung 2.1 zeigt schematisch das Beschleuniger-System am CERN.

Der wesentliche Vorteil von Beschleunigern hadronischer Strahlen liegt in der großen Masse der gespeicherten Teilchen. Die erreichbaren Energien werden dann nämlich nicht mehr durch starke Synchrotronstrahlung beschränkt. Die dadurch verlorene Energie pro Umlauf verhält sich proportional zu $E^4m^{-4}R^{-1}$ (wobei R den Radius des Speicherrings angibt) und liegt damit beispielsweise für leichte Teilchen wie Elektronen oder Positronen etwa um den Faktor 10^{13} höher, als für Protonen oder Antiprotonen. Die Energie für die Teilchenstrahlen wird dann hauptsächlich durch die tech-

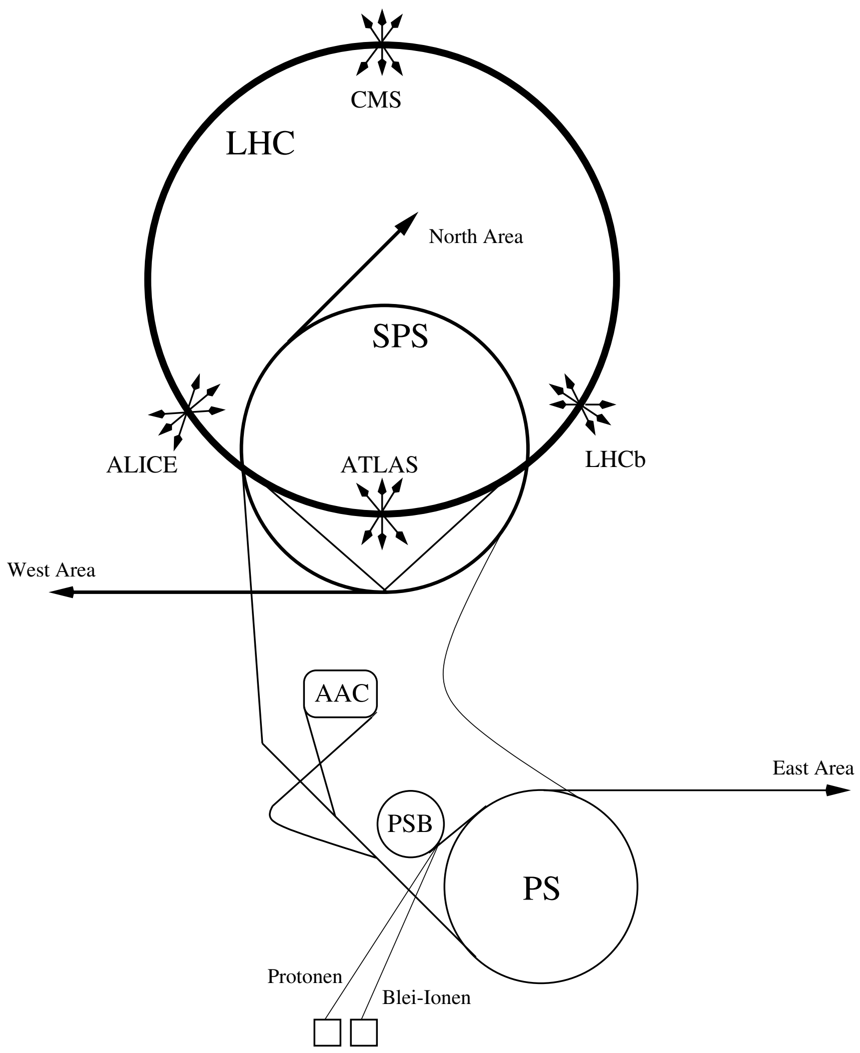


Abbildung 2.1: Skizze des LHC mit seinen Experimenten ALICE, ATLAS, CMS und LHCb sowie den Vorbeschleunigern SPS und PS, dem Antiproton Accumulator Complex (AAC) und dem Proton Synchrotron Booster (PSB). Die Darstellung ist nicht maßstabsgetreu. [18]

Protonen-Speicherung im LHC	
Energie bei der Kollision	7 TeV
Energie bei der Injektion	450 GeV
Dipol-Feldstärke bei 7 TeV	8,33 T
Luminosität	$10^{34}\mathrm{cm}^{-2}\mathrm{s}^{-1}$
Strom eines Strahls	0,56 A
Anzahl der Teilchenbündel pro Strahl	2808
Räumlicher Abstand zwischen den Bündeln	7,48 m
Zeitlicher Abstand zwischen Bündeln	24,95 ns
Zahl der Teilchen pro Bündel	$1,1 \cdot 10^{11}$
Lebensdauer der Luminosität	10 h
Energieverlust pro Umlauf	7 keV
Abgestrahlte Leistung pro Strahl	3,8 kW
Gespeicherte Energie pro Strahl	350 MJ
Füllzeit pro Ring	4,3 min

Tabelle 2.1: Allgemeine Parameter des LHC bei der Speicherung von Protonenstrahlen. [12]

nisch realisierbaren Magnetfelder limitiert, die die geladenen Teilchen auf einer Kreisbahn halten. Am LHC werden durch supraleitende Magnetspulen Feldstärken über 8 Tesla erreicht, wodurch Protonenstrahlen mit Energien von 7 TeV pro Teilchen gespeichert werden können

Die höchsten Schwerpunktsenergien bei Teilchenkollisionen erreicht zur Zeit das Tevatron am Fermilab, wo bei Zusammenstößen zwischen Protonen und Antiprotonen bis zu 2 TeV frei werden. Im LHC werden Protonen bei 14 TeV und Blei-Ionen bei 1150 TeV kollidieren. Zwar steht bei der Kollision komplexer Teilchen nicht mehr die komplette Strahlenergie für Elementarreaktionen zur Verfügung, da sie sich auf mehrere Quarks und Gluonen verteilt, dennoch liegen die erreichbaren Energien in $q\bar{q}$-Reaktionen im TeV-Bereich, während beispielsweise am LEP zuletzt „nur“ 206 GeV in e^+e^--Kollisionen erreicht wurden.

Der LHC zeichnet sich neben seiner Vielseitigkeit und den hohen Energien auch durch seine bislang unerreichten Reaktionsraten aus. Und dies, obwohl bei höheren Teilchenenergien der Wirkungsquerschnitt aufgrund der kleineren De Broglie Wellenlängen mit $1/E^2$ abnimmt. Die hohe Produktivität wird durch die enorme Luminosität von $10^{34}\mathrm{cm}^{-2}\mathrm{s}^{-1}$ gewährleistet. Dazu werden beide Ringe mit 2835 Bündeln von jeweils 10^{11} Teilchen gefüllt, die dann an den Wechselwirkungszonen mit einer Frequenz von 40 MHz kollidieren. Weitere Parameter für die Speicherung von Protonen sind in Tabelle 2.1 aufgeführt [12].

2.2 Der Detektor

Der LHCb Detektor ist ein einarmiges, magnetisches Vorwärts-Spektrometer, das um die Strahlachse gebaut wird. In der Horizontalen wird ein Winkelbereich von 10 bis 300 mrad abgedeckt. Wegen des vertikalen Magnetfeldes ist dies die Ebene, in denen die Bahnen geladener Teilchen gekrümmt sind, weswegen auch von der Krümmungsebene gesprochen wird. In der Vertikalen reicht der erfaßte Bereich nur bis 250 mrad, die Bahnen bleiben hier aber auch gerade. Diese Wahl der Geometrie begründet sich in der Tatsache, daß bei hohen Energien b- und $\bar{\mathrm{b}}$-Hadronen bevorzugt in dieselbe Richtung unter kleinen Winkeln zur Strahlachse entstehen, was mit Hilfe von simulierten Ereignissen in Abbildung 2.3 illustriert wird. Abbildung 2.4 zeigt den Anteil von $\mathrm{B_d^0} \to \pi^+\pi^-$ Ereignissen, der durch das Spektrometer rekonstruiert werden kann. Einen Schnitt durch das komplette Spektrometer in beiden Ebenen zeigt Abbildung 2.2. Das in diesem und allen weiteren Kapiteln verwendete Koordinatensystem wird in Abbildung 2.5 beschrieben.

LHCb besteht aus einem Vertex-Detektor zur Positionsbestimmung von Vertices nahe der Wechselwirkungszone, zwei RICH-Detektoren zur Teilchenidentifizierung, einem Magneten, um Impulsmessungen zu ermöglichen, einem Spurdetektor zur Rekonstruktion von Teilchenbahnen, zwei Kalorimetern zur Messung von Teilchenenergien und einem Myon-Detektor. Die Selektion von Ereignissen erfolgt über ein komplexes Trigger-System, das auf die Anreicherung von Ereignissen mit B-Zerfällen durch die Suche nach Sekundärvertices und Teilchen mit hohem Transversalimpuls spezialisiert ist.

Der Stand der hier vorgestellten Daten entspricht dem Technical Proposal [17] und wird sich in einigen Details bis zur Inbetriebnahme noch ändern. Das Experiment ähnelt in seinem Aufbau stark dem Detektor HERA-B am DESY, der seit Februar 2000 in Betrieb ist, und kann daher von den dort gewonnenen Erkenntnissen profitieren.

2.2.1 Der Magnet

Um geladene Teilchen auf Kreisbahnen zu lenken und so Impulsmessungen mit Hilfe der Spurdetektoren zu ermöglichen, wird ein Dipolmagnet installiert, der ein vertikales Magnetfeld mit guter Homogenität über einen horizontalen Bereich von 4,3 m und maximale Feldstärken von 1,1 T erreicht, was in einem Feldintegral entlang der Strahlachse von 4,0 Tm resultiert. Im Magnetfeld wird eine Energie von 50 MJ gespeichert, wobei ein Strom von 20 kA durch die supraleitenden Spulen fließt. Um die erforderliche Größe möglichst gering zu halten, wird der Magnet bereits nach dem Vertex-Detektor und dem ersten RICH montiert, die durch einen Eisenschild vom

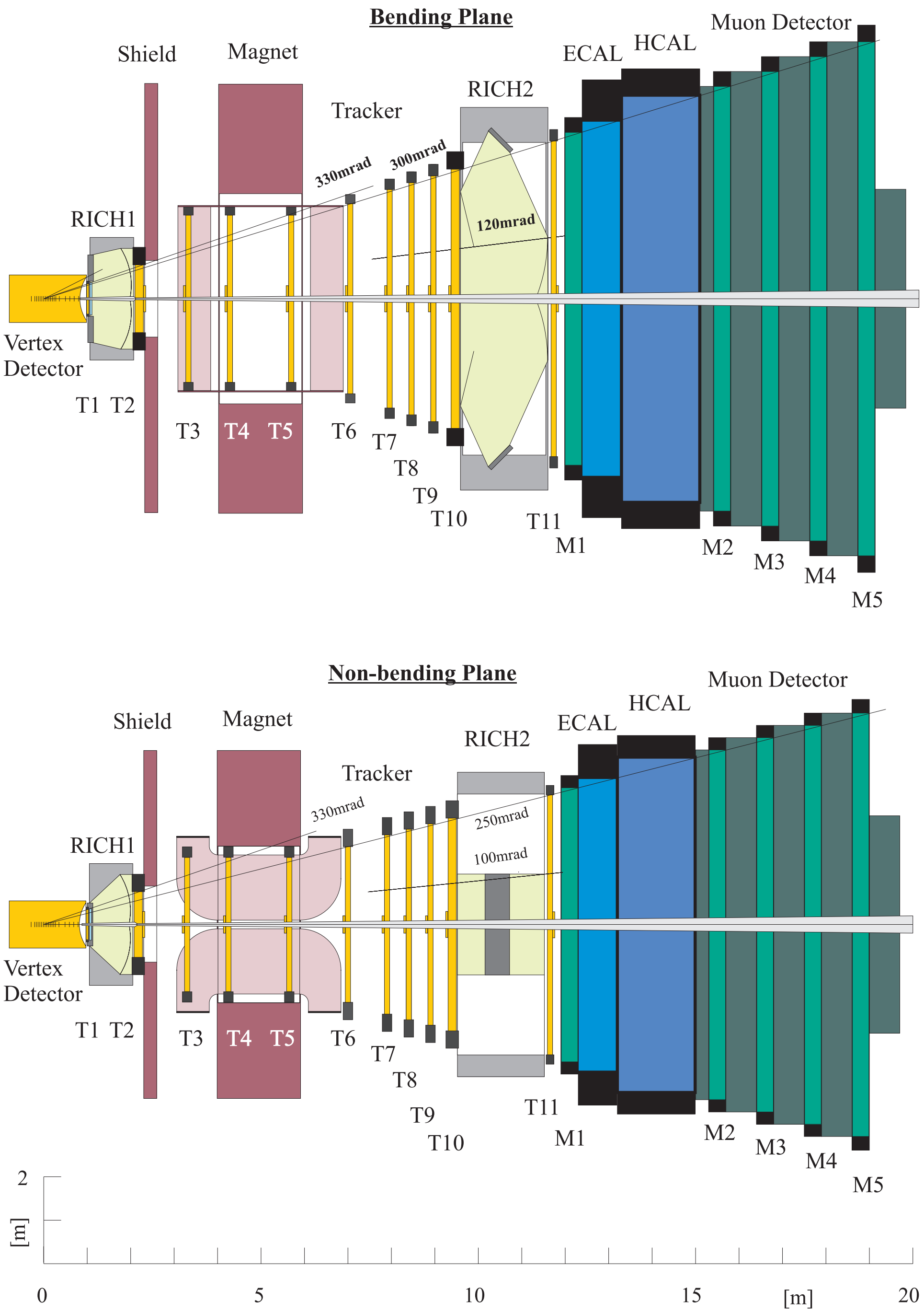


Abbildung 2.2: Der LHCb-Detektor in der Ebene gekrümmter und gerader Spuren. [17]

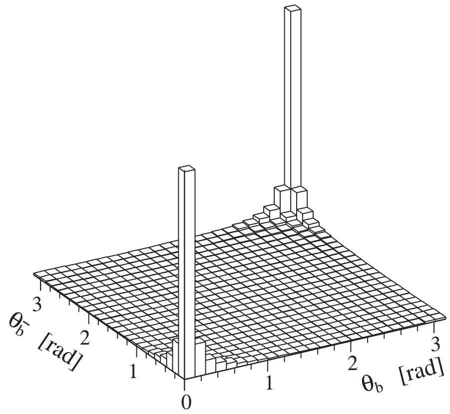


Abbildung 2.3: Polarwinkelverteilung von Paaren aus b- und $\bar{b}$-Hadronen, die vom Ereignis-Generator PYTHIA erzeugt wurden. [17]

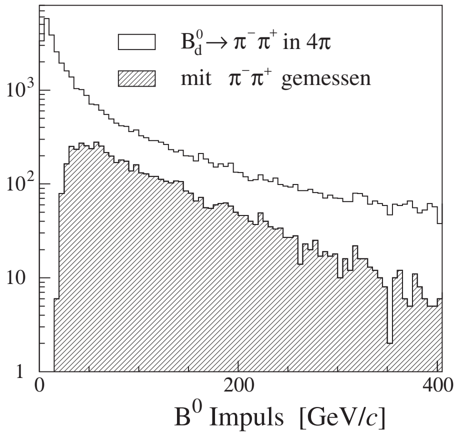


Abbildung 2.4: Impulsverteilung von $B_d^0 \to \pi^+\pi^-$ Zerfällen in den gesamten Raumwinkel und solchen, deren Pionen im Spektrometer rekonstruiert werden können. [17]

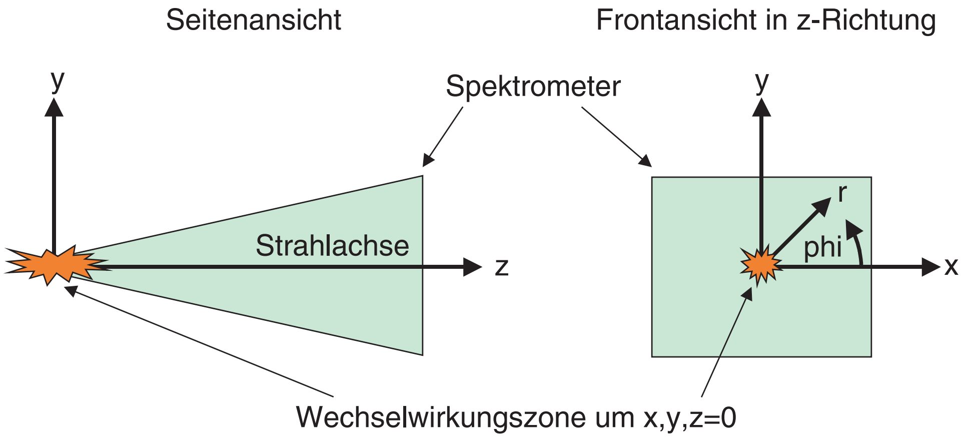


Abbildung 2.5: Das verwendete Koordinatensystem.

störenden Einfluß des Magnetfelds abgeschirmt werden.

2.2.2 Der Vertex-Detektor

Der Silizium-Vertex-Detektor soll Präzisionsmessungen von Spurkoordinaten nahe am primären Wechselwirkungspunkt ermöglichen. Diese werden vom Level-1 Trigger zur Suche nach Sekundärvertices und in der Offline-Rekonstruktion zur exakten Positionsbestimmung der Entstehungs- und Zerfallsvertices von B-Mesonen benötigt.

Der Vertex-Detektor besteht aus 17 Stationen, die entlang der Strahlachse verteilt und senkrecht zur ihr ausgerichtet sind. Sie enthalten je zwei Scheibendetektoren aus Silizium, in denen kreisförmig bzw. radial angeordnete Streifen die von Teilchen erzeugten Ladungen aufnehmen und so Signale erzeugen, aus denen r- und ϕ-Koordinaten konstruiert werden können. Der Vorteil solche Halbleiterdetektoren gegenüber Driftkammern liegt in der hohen Auflösung und der kurzen Ansprechzeit. Letzteres ist besonders für die Auswertung durch das Trigger-System wichtig. Die erreichbaren Ortsauflösungen liegen für die Koordinaten je nach Streifenabstand in den unterschiedlichen Sektoren und der Zahl benachbarter Signale zwischen 6 und 40 μm. Die Stationen sind in zwei Hälften unterteilt, die während der Injektion von Teilchenstrahlen in den LHC um 3 cm vom der Strahlachse entfernt werden können, um Strahlenschäden zu vermeiden. Da genaue Lage und Geometrie der Stationen und Streifen von entscheidender Bedeutung für den Trigger-Algorithmus sind, wird darauf gesondert in Abschnitt 4.1 eingegangen.

Zwei weitere Siliziumdetektoren zur Messung von r-Koordinaten sind hinter den Stationen des Vertex-Detektors in entgegengesetzter Richtung zum Spektrometerarm installiert. Sie liefern Daten für den Level-0 Trigger, der damit die z-Koordinaten von Primärvertices abschätzen und so unerwünschte Ereignisse mit mehr als einer pp-Reaktion früh erkennen und zurückweisen kann. Diese Einrichtung wird als Pile-up Veto Counter bezeichnet.

2.2.3 Die Spurdetektoren

Das System von Spurdetektoren soll die genaue Rekonstruktion der Bahnen geladener Teilchen und die Präzisionsmessung ihres Impulses ermöglichen. Außerdem ist die Kenntnis der Richtung von Spuren für die Rekonstruktion der Čerenkov-Ringe in den RICH-Detektoren erforderlich. Das System besteht aus 11 Stationen (T1–T11), die zwischen dem Vertex-Detektor und den Kalorimetern verteilt sind.

Bei den Stationen handelt es sich im äußeren Bereich um konventionelle

Driftkammern, die für den hohen Teilchenfluß im inneren Bereich um die Strahlachse allerdings ungeeignet sind. Hier kommen wahrscheinlich 3-GEM oder Silizium-Detektoren zum Einsatz. Die Spurkoordinaten in der Krümmungsebene werden von allen Stationen durch Streifen gemessen, die vertikal — zum Teil mit Stereowinkeln von $\pm 5°$ zur 3D-Rekonstruktion — ausgerichtet sind. Die Stationen unmittelbar vor und hinter den RICH-Detektoren (T2, T10 und T11) erlauben durch zusätzlich Streifen in x-Richtung die Messung der vertikalen Spurkoordinaten.

2.2.4 Die RICH-Detektoren

Mit Hilfe der RICH-Detektoren (Ring Image Čerenkov Counter) werden geladene Teilchen identifiziert, insbesondere Pionen von Kaonen unterschieden. Es handelt sich dabei um Čerenkov-Zähler, die durch die Messung der Intensität und des Emissionswinkels von Čerenkov-Licht Rückschlüsse auf die Energie und Geschwindigkeit von Teilchen zulassen, die sich schneller als Licht im Detektormedium bewegen. Da Teilchen über den weiten Impulsbereich von 1–150 GeV identifiziert werden sollen, sind zwei Detektoren mit unterschiedlichen Medien erforderlich.

RICH1 ist für Teilchen mit niedrigen Impulsen ausgelegt und befindet sich — abgeschirmt vom Magnetfeld — unmittelbar hinter dem Vertex-Detektor. Hier können auch Teilchen detektiert werden, die auf ihrem weiteren Weg durch den Detektor wegen ihres niedrigen Impulses und der daraus resultierenden stärkeren Ablenkung durch das Magnetfeld den Bereich der Detektorakzeptanz verlassen. Als Radiator wird eine Schicht aus Aerogel und C_4F_{10}-Gas eingesetzt. Das Čerenkov-Licht wird durch sphärische Spiegel zu beiden Seiten der Strahlachse auf die Photodetektoren fokussiert, die dann ringförmige Signalverteilungen registrieren. RICH2 befindet sich zwischen den Spurdetektoren und den Kalorimetern und ist für die Detektion von Teilchen mit höheren Impulsen gedacht. Als Radiator wird hier CF_4-Gas eingesetzt (Abb. 2.6).

2.2.5 Die Kalorimeter

Die Kalorimeter dienen in erster Linie dazu, Elektronen und Hadronen zu identifizieren und deren Position und Energie zu messen. Diese Messungen werden sowohl von mehreren Trigger-Algorithmen ausgewertet, als auch in der Offline-Analyse zur Rekonstruktion benötigt. In Kalorimetern werden Teilchen absorbiert, indem sie sich durch elekromagnetische oder hadronische Wechselwirkung in Materieschichten in Teilchenschauer auflösen, deren Ausmaße und Energien mit Hilfe von Szintillatoren gemessen werden.

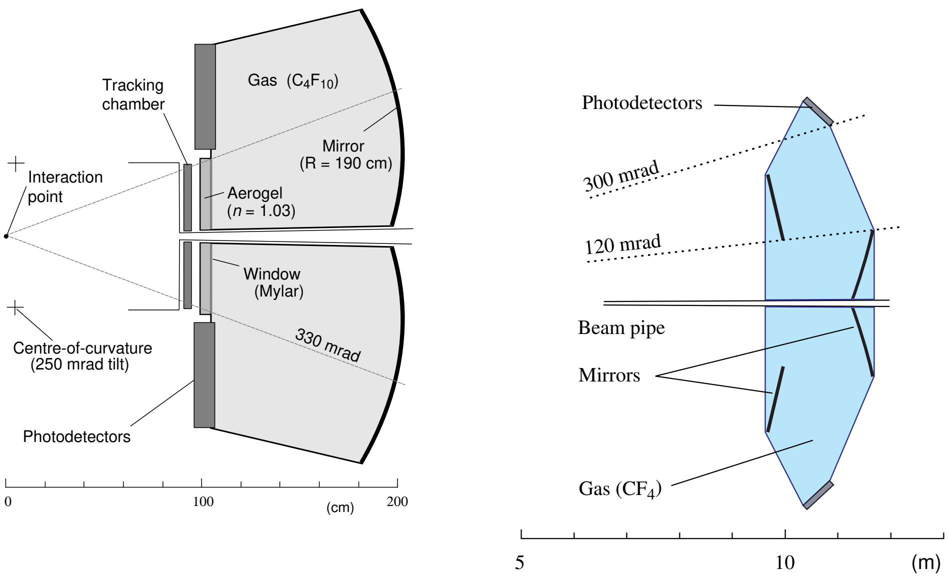


Abbildung 2.6: Skizzen des RICH1- und RICH2-Detektors (von oben betrachtet). [17]

Wie für Experimente dieser Art typisch, besitzt auch LHCb ein elektromagnetisches (ECAL) und ein hadronisches (HCAL) Kalorimeter. Das ECAL enthält Bleischichten, in denen hauptsächlich Elektronen, Positronen und Photonen schauern und damit deren Positions- und Energiebestimmung möglich macht. Außerdem hilft es bei der Rekonstruktion neutraler Pionen (Zerfall in Photonen). Die Gesamtstrahlungslänge beträgt $25\,\mathrm{X}_0$ und es können Energien bis zu 200 GeV gemessen werden. Im HCAL wird Eisen als Absorber für Hadronen verwendet, damit auch deren Position und Energie ermittelt werden kann. Darüberhinaus wird unter anderem die Trennung hochenergetischer Elektronen von Hadronen ermöglicht und die Myon-Identifizierung verbessert. Die durchschnittliche Strahlungslänge im HCAL beträgt $68\,\mathrm{X}_0$, die hadronische Absorptionslänge $7,3\,\lambda_\mathrm{I}$ und es kann Energien bis 300 GeV messen.

2.2.6 Der Myon-Detektor

Der Myon-Detektor besteht aus einer Station vor den Kalorimetern (M1) und vier Stationen dahinter (M2–M5), die in einem longitudinal segmentierten Stahlschild untergebracht sind, der Elektronenen, Photonen und Hadronen absorbiert. Mit Hilfe dieses Detektors können Myonen identifiziert und deren Bahn bestimmt werden, da diese wegen ihrer großen Masse nur wenig Energie durch Ionisation verlieren und daher als einzige geladene Teilchen

Ereignisrate		40 MHz
Level-0	Latenz	3,2 μs
	Akzeptanzrate	1 MHz
Level-1	Latenz [35]	1024 μs
	Akzeptanzrate	40 kHz
Level-2	Latenz	$\approx$10 ms
	Akzeptanzrate	1 kHz
Level-3	Latenz	$\approx$200 ms
	Akzeptanzrate	200 Hz

Tabelle 2.2: Latenz-Zeiten und durchschnittliche Akzeptanzraten der einzelnen Trigger-Stufen. [17]

die Kalorimeter und den gesamten Myonschild durchdringen und so Signale in den Myon-Stationen erzeugen. Diese Informationen werden in der Offline-Rekonstruktion benötigt, aber auch vom Level-0 Trigger zur Erkennung von Myonen mit hohem Transversalimpuls ausgewertet. In Regionen mit hohen Signalraten, also insbesondere in der nicht abgeschirmten Station M1, werden voraussichtlich Cathode Pad Chambers (CPC) und ansonsten Multigap Resistive Plate Chambers (MRPC) zum Einsatz kommen.

2.3 Das Trigger-System

Der LHC produziert Ereignisse mit einer Rate von 40 MHz, die im Durchschnitt jeweils 100 kB Meßdaten erzeugen. Das Datenaufkommen liegt somit bei 4 TB/s und muß vor der Aufzeichnung erheblich reduziert werden. Dies geschieht durch ein vierstufiges Trigger-System, das die Ereignisrate auf 200 Hz senken soll, so daß die Aufzeichnung der Daten zur späteren Offline-Analyse mit etwa 20 MB/s erfolgt. Die Ereignisselektion geschieht anhand der Wahrscheinlichkeit für die Existenz rekonstruierbarer Zerfälle der Mesonen $\mathrm{B_d^0}$ und $\mathrm{B_s^0}$. Ereignisse mit B-Mesonen können von anderen inelastischen pp-Reaktionen durch die Präsenz von Sekundärvertices und Teilchen mit hohem Transversalimpuls unterschieden werden, was für den Zerfall $\mathrm{B_d^0} \to \pi^+\pi^-$ in den Abbildungen 2.7 und 2.8 illustriert ist. Die vorgeschlagenen Werte für die Latenzen und Akzeptanzraten der einzelnen Trigger-Level sind in Tabelle 2.2 zusammengefaßt. Die Latenz-Zeiten bestimmen die erforderlichen Puffergrößen für das Auslesesystem und sollten daher so klein wie möglich gehalten werden.

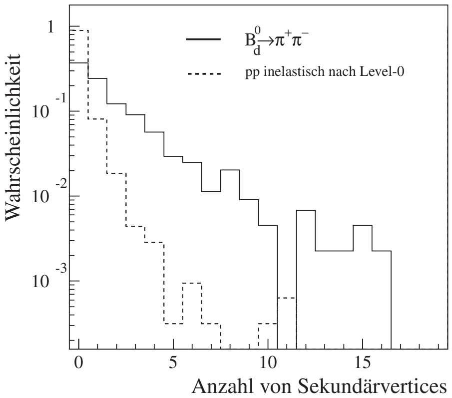


Abbildung 2.7: Anzahl potentieller Sekundärvertices für den Level-1 Trigger für alle inelastischen pp-Reaktionen und für $B^0_d \to \pi^+\pi^-$ Ereignisse. [17]

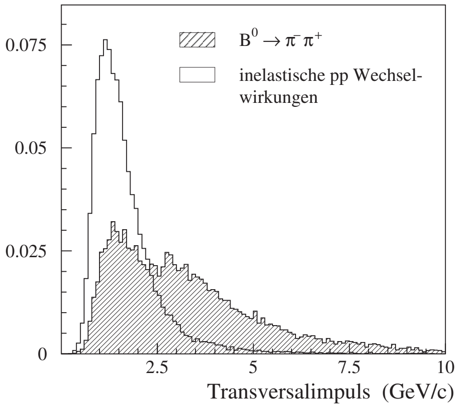


Abbildung 2.8: Verteilung des Transversalimpulses der geladenen Hadronen mit dem größten Wert im Ereignis für inelastische pp-Reaktionen und $B^0_d \to \pi^+\pi^-$ Ereignisse. [17]

2.3.1 Level-0 Trigger

Bei den Level-0 Triggern handelt es sich um besonders schnelle Hardware-Trigger, die mit der Ereignisrate von 40 MHz am Wechselwirkungspunkt arbeiten müssen. Sie erkennen mit Hilfe von Meßdaten aus den Kalorimetern und dem Myon-Detektor Elektronen, Photonen, Hadronen und Myonen mit hohem Transversalimpuls als Indiz für die Existenz eines B-Zerfalls. Außerdem werden Ereignisse mit mehr als einer pp-Reaktion zurückgewiesen, da deren Auswertung zu kompliziert ist. Solche Mehrfachreaktionen werden vom Pile-up Veto Counter erkannt, der durch primitive Spurrekonstruktion die z-Koordinaten von Primärvertices abschätzen kann.

2.3.2 Level-1 Trigger

Level-1 besteht aus zwei Komponenten, die mit der Akzeptanzrate des Level-0 Triggers von 1 MHz operieren. Der Vertex-Trigger soll Ereignisse mit einem oder mehreren sekundären Vertices erkennen, indem er die Meßdaten des Vertex-Detektors zur Rekonstruktion von Spuren verwendet. Dies wird durch einen Software-Algorithmus geschehen, der auf den Rechnern einer Prozessor-Farm zum Einsatz kommt. Da die Implementierung eines solchen Algorithmus Gegenstand dieser Arbeit ist, wird die genaue Rekonstruktions-Methode detailliert in Kapitel 4 behandelt. Die zweite Komponente wertet Daten der Spurdetektoren aus, um Fehler bei der Identifizierung von Teilchen mit hohem Transversalimpuls durch Level-0 auszumachen, die durch Sekundärreaktionen im Detektor oder überlappende Schauer in den Kalorimetern entstehen können.

2.3.3 Level-2 und Level-3 Trigger

Die Selektions-Algorithmen der Highlevel-Trigger L2 und L3 sind vollständig in Software auf Prozessor-Farmen implementiert. Level-2 sondert vom Vertex-Trigger akzeptierte Ereignisse aus, in denen falsche Sekundärvertices konstruiert wurden. Dazu werden zusätzlich zu den Meßwerten des Vertex-Detektors Positions- und Impulsinformationen der Spurdetektoren ausgewertet. Auf Level-3 Ebene kommen schließlich Rekonstruktions- und Analysealgorithmen zum Einsatz, die Informationen sämtlicher Detektorkomponenten verarbeiten und so die Auswahl bestimmter b-Hadron Zerfälle ermöglichen. Dadurch wird schließlich die finale Akzeptanzrate von 200 Hz erreicht, mit der die Ereignisdaten aufgezeichnet werden.

Kapitel 3

Motivation der Arbeit

Voraussetzung für den effektiven Einsatz des Level-1 Triggers ist eine leistungsstarke Software, die innerhalb einer vorgeschriebenen Zeit eine unter physikalischen Gesichtspunkten sinnvolle Bewertung von Ereignissen liefern kann.

Die zugrunde liegende Hardware wird aus einer Farm kommerzieller Personal Computer (PC) bestehen, die durch ein Hochgeschwindigkeits-Netzwerk miteinander verbunden sind, über das sie mit Ereignisdaten versorgt werden. Der Prototyp eines solchen Triggers wird derzeit am Kirchhoff-Institut für Physik in Heidelberg entwickelt [20]. Ein Algorithmus, der auf diesen Rechnern zum Einsatz kommen könnte, existierte bislang nur im Rahmen des für LHCb entwickelten Standard Monte-Carlo-Simulationspakets SICb als Implementierung in Fortran [22].

Inhalt dieser Arbeit ist die erste Implementierung eines ähnlichen Algorithmus in C++, der in den Grundzügen eine vergleichbare Strategie bei der Suche nach Teilchenspuren und Zerfallsvertices anwendet, wie der angesprochene Fortran-Code. Damit werden im wesentlichen zwei Ziele verfolgt. Zunächst soll die von der LHCb Collaboration beschlossene Umstellung der Rekonstruktions- und Analyse-Software von Fortran auf C++ unterstützt werden. Es soll gezeigt werden, daß keine grundsätzlichen Probleme bei der Portierung entstehen, und daß die speziell für diesen Zweck neu entwickelte Programmierumgebung GAUDI einen geeigneten Rahmen bieten kann.

Die Hauptmotivation liegt jedoch darin, eine Möglichkeit zu schaffen, mit der das Zeitverhalten des Algorithmus detailliert untersucht werden kann. Dies ist von entscheidender Bedeutung für die Planung der Hardware, was die Dimension der Prozessor-Farm, die benötigte Leistung der einzelnen Rechner und die verwendete Technik zur Verteilung der Ereignisdaten angeht. Da der Trigger eine Ereignisrate von 1 MHz verarbeitet, erfordert jede Mikrosekunde, die der Algorithmus im Durchschnitt zusätzlich benötigt, einen weiteren Farm-Computer. Auch die maximal mögliche Auswertungsdauer ist eine wichtige Größe, weil sie die benötigte Latenz bestimmt.

Über die Messung absoluter Zeiten hinaus gilt es herauszufinden, welchen Einfluß dabei Werte wie die Ereignisgröße, die Rechengeschwindigkeit der CPU, die Speicherbandbreite, die Cache-Größe bzw. allgemein die Rechnerarchitektur hat. Nur so kann die Hardware bestmöglich an die Bedürfnisse der Software angepaßt und so ein kostengünstiges Konzept realisiert werden. Außerdem soll versucht werden, durch Geschwindigkeitsoptimierungen am Code und effektivere Suchstrategien die Anforderungen an die Rechnergeschwindigkeit möglichst gering zu halten.

Voraussetzung für solche Studien ist eine Software, die sich schnell und unkompliziert installieren läßt und somit ohne großen Aufwand auf unterschiedlichen Systemen getestet werden kann. Um den Code für die verwendete Architektur optimieren zu können, ist außerdem eine Programmiersprache wie C++ hilfreich. Besonders wichtig aber ist, daß der Algorithmus mit simulierten Daten dasselbe Verhalten zeigt, wie man es im späteren Einsatz mit echten Meßwerten erwartet. Dies erfordert ein Konzept, das im Bereich von Hochenergie-Experimenten bislang unüblich war, nämlich den Code der Rechenroutinen des Algorithmus so zu entwickeln, daß er sowohl in Simulations- als auch in Experimentalumgebung einsatzfähig ist. Der Algorithmus greift dann also nicht unmittelbar auf die von der Simulation erzeugten Datenstrukturen zu, sondern erst nachdem diese in ein Format konvertiert wurden, wie es später von Ausleseeinheiten erzeugt und von der Netzwerk-Hardware in den Speicher der Farm-Computer geschrieben wird. Dies ermöglicht dann auch, mit diesem Format in Hinblick auf schnellere Speicherzugriffe zu experimentieren. Ein entscheidender Vorteil der Auswertung von Simluationsdaten, nämlich die Kenntnis der physikalischen Ursachen für Meßwerte, darf dabei jedoch nicht verloren gehen. Dies erhöht den Aufwand der Implementierung, da parallel zu den rekonstruierten Spuren und Vertices aus den reinen Meßdaten noch Datenstrukturen aufgebaut werden müssen, die es erlauben, auf die zugehörigen Daten der Monte-Carlo-Simulation zurückzugreifen. Damit können die Ergebnisse qualitativ bewertet werden.

All dies ist mit dem L1-Code in SICb nicht zu erreichen. Bereits die Option zur Messung der Ausführungszeiten ist nicht verfügbar. Außerdem müssen Rechner, auf denen SICb zur Anwendung kommen soll, in die Software-Infrastruktur vom CERN eingebunden sein, wodurch der Einsatz auf Prototypen außerhalb des CERN extrem aufwendig würde. Somit ist die unbedingte Notwendigkeit gegeben, eine Arbeit wie die vorliegende anzufertigen.

Kapitel 4

Der Vertex-Trigger-Algorithmus

Ein entscheidendes Merkmal für Ereignisse des LHCb Experiments, die vom Level-1 Trigger selektiert werden sollten, ist die Anzahl von Sekundärvertices mit signifikantem Abstand zum primären Wechselwirkungspunkt. Bei diesen handelt es sich mit hoher Wahrscheinlichkeit um Zerfälle von B-Mesonen, die bei mittlerer Lebensdauer etwa 7 mm zurücklegen. Um diese Entscheidung in möglichst kurzer Zeit treffen zu können, wurde ein spezieller Vertex-Detektor entworfen, der eine schnelle Spur- und Vertexrekonstruktion in Echtzeit mit hoher Genauigkeit und Effektivität ermöglichen soll. Ein Algorithmus zur Berechnung einer solchen Entscheidung wurde im Rahmen dieser Arbeit entwickelt und seine Funktionsweise wird in diesem Kapitel beschrieben. Zunächst wird jedoch die Geometrie des Vertex-Detektors erläutert, wie sie für die Monte-Carlo-Simulation von Ereignissen und die Implementierung verwendet wurde.

4.1 Die Geometrie des Vertex-Detektors

Der Vertex-Detektor besteht aus 17 kreisrunden Stationen, die senkrecht zum Strahl in Richtung der z-Achse verteilt sind (Abb. 4.1). Die Stationen sind in eine obere und untere Hälfte unterteilt, die um 2 cm versetzt positioniert sind und während der Strahlinjektion um 3 cm auseinandergebracht werden. Jede der beiden Hälften gliedert sich in 3 Sektoren mit einer Spanne von jeweils 61°. Dadurch überlappen beide an den Seiten in der x/y-Ebene um 3°, wodurch eine schnelle Ausrichtung des Detektors im Betrieb ermöglicht wird. Die Stationen enthalten im Abstand von 1 mm zwei Silizium-Scheiben zur Messung der r- und ϕ-Koordinate von geladenen Teilchen. Die Sektoren

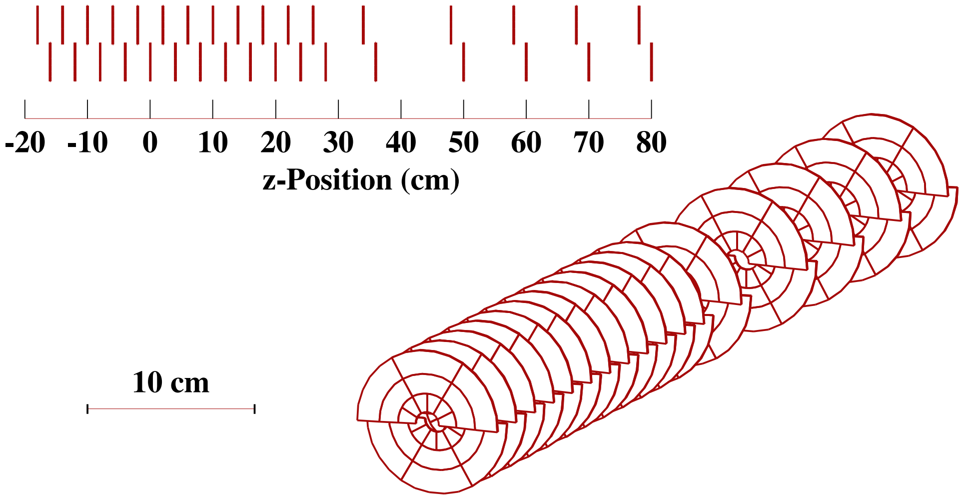


Abbildung 4.1: Anordnung der 17 Stationen des Vertex-Detektors entlang der Strahlachse. [17]

zur Messung der r-Koordinate unterteilen sich weiter in 4 Untersektoren, in denen die Streifen zur Signalerzeugung mit konstantem Radius und Abstand zueinander verlaufen, während die ϕ-Detektoren lediglich aus 2 Untersektoren bestehen und der Streifenabstand mit zunehmendem Radius wächst (Abb. 4.2 und 4.3). Die Ausmaße und Streifendichten sind in Tabelle 4.1 zusammengefaßt.

Die Suche nach Vertices wird entscheidend vereinfacht und beschleunigt, wenn der Detektor einer solchen r/ϕ-Geometrie folgt. Allein mit den r-Koordinaten können Spuren in der r/z-Ebene bestimmt werden, die bereits zur Lokalisierung des primären Vertex ausreichen. Dessen Position erlaubt es, den Nutzen der gefundenen Spuren in der r-Projektion für die weitere Berechnung der Trigger-Entscheidung abzuschätzen. Die ϕ-Koordinaten brauchen dann nur für einen kleinen Bruchteil der Spuren ausgewertet zu werden, um deren genaue Lage im Raum zu erhalten und so die Suche nach Zerfällen

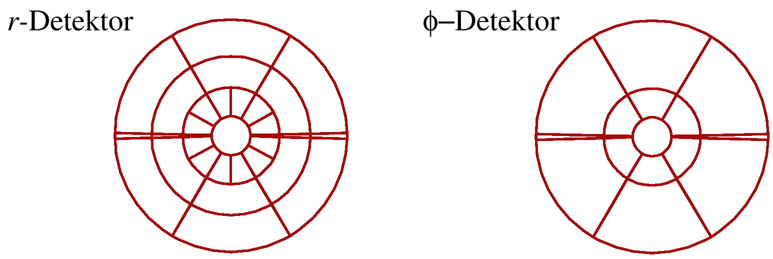


Abbildung 4.2: Anordnung der Untersektoren im ϕ- und r-Detektor.

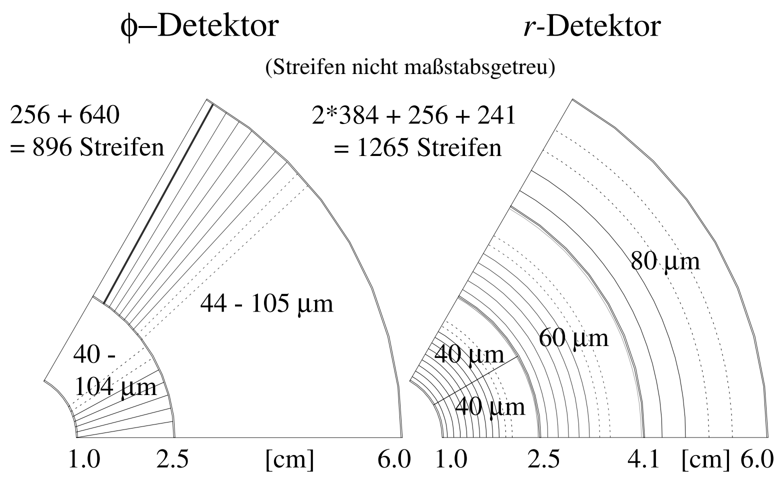


Abbildung 4.3: Anordnung der Streifen im r- und ϕ-Detektor. [17]

Untersektor	Radius	Streifenanzahl	Streifenabstand
r Innen	1,0–2,5 cm	384	40 μm
r Mitte	2,5– 4,1 cm	256	60 μm
r Außen	4,1– 6,0 cm	241	80 μm
r Gesamt	**1,0–6,0 cm**	**1265**	
ϕ Innen	1,0–2,5 cm	256	40–104 μm
ϕ Außen	2,5–6,0 cm	640	40–104 μm
ϕ Gesamt	**1,0– 6,0 cm**	**896**	

Tabelle 4.1: Ausmaße und Streifendichte in den Untersektoren der r- und ϕ-Detektoren. [31]

von B-Mesonen zu ermöglichen. Ein weiterer Vorteil liegt in der radialen Unterteilung in Untersektoren. Die Geometrie wurde so gewählt, daß der erste Meßpunkt von Spuren, die für die physikalische Auswertung von Interesse sind, in einem der inneren Untersektoren liegt. Wie Simulationen zeigen, ist dort die Spurmultiplizität in etwa eins, wodurch der kombinatorische Aufwand bei der Suche nach Spuren erheblich reduziert wird. Die r/ϕ-Bauart stellt darüberhinaus gegenüber einer x/y-Geometrie eine flexiblere Konstruktion in Hinsicht auf eine Optimierung des Algorithmus dar. Beispielsweise können die Streifenabstände leichter an die erforderlichen Auflösungen und Belegungsgrade angepaßt werden, welche hauptsächlich vom Radius abhängen. Außerdem wird eine Parallelisierung der Berechnung stark vereinfacht, weil die Spurensuche in den einzelnen ϕ-Sektoren voneinander unabhängig abläuft.

Geladene Teilchen, die durch einen der Silizium-Detektoren fliegen, erzeugen häufig Signale auf mehreren benachbarten Streifen. Solche Häufungen von Signalen werden bereits von der Off-Detector Electronic (ODE) erkannt und als *Cluster* bezeichnet. Bei den Daten, die an den Level-1 Trigger geleitet werden, handelt es sich dann bereits um gewichtete Koordinaten solcher Cluster. Dadurch wird eine bessere Ortsauflösung erreicht, als sie sich allein aus dem Streifenabstand ergeben würde, und die Datenmenge reduziert. Auf die Präzision, mit der die Simulation rechnet, geht Abschnitt 5.3 ein.

Die Ausrichtung der Streifen in den ϕ-Detektoren weist eine Besonderheit auf. Sie sind am Punkt des innersten Radius des jeweiligen Untersektors um $5°$ gedreht. Die Richtung wechselt dabei in aufeinanderfolgenden Stationen. Die gemessenen ϕ-Koordinaten beschreiben daher nicht die tatsächliche Teilchenposition, sondern die des getroffenen Streifens am inneren Rand des jeweiligen Untersektors. Um die wirkliche Koordinate zu erhalten, muß deshalb ein Korrekturterm berechnet werden, der vom dazugehörigen Radius abhängt. Der Vorteil dieser Streifenanordnung liegt darin, daß durch die r-Abhängigkeit der ϕ-Koordinaten Mehrdeutigkeiten besser aufgelöst werden können. Dies wird anhand eines einfachen Beispiels von zwei Spuren in Abbildung 4.4 demonstriert. Die Spuren schneiden beide die z-Achse, jedoch unter verschiedenen Winkeln ϕ. Auf jeder Station sprechen jeweils zwei r- und ϕ-Streifen an. Aus dieser Information ergeben sich vier mögliche Positionen für Meßpunkte an den Schnittstellen der getroffenen Streifen, von denen die richtigen als Punkte und die falschen als Kreuze markiert sind. Mit radialen Streifen im ϕ-Detektor würden sich daraus vier mögliche r/ϕ-Tripletts und somit Spuren ergeben. Mit entgegengesetzt rotierten Streifen auf aufeinanderfolgenden Stationen liegen die Kreuze nicht mehr auf einer Gerade und werden deshalb vom Algorithmus auch nicht mehr fälschlicherweise als Triplett angenommen.

Ob die beschriebene Geometrie beim Bau des Detektors tatsächlich rea-

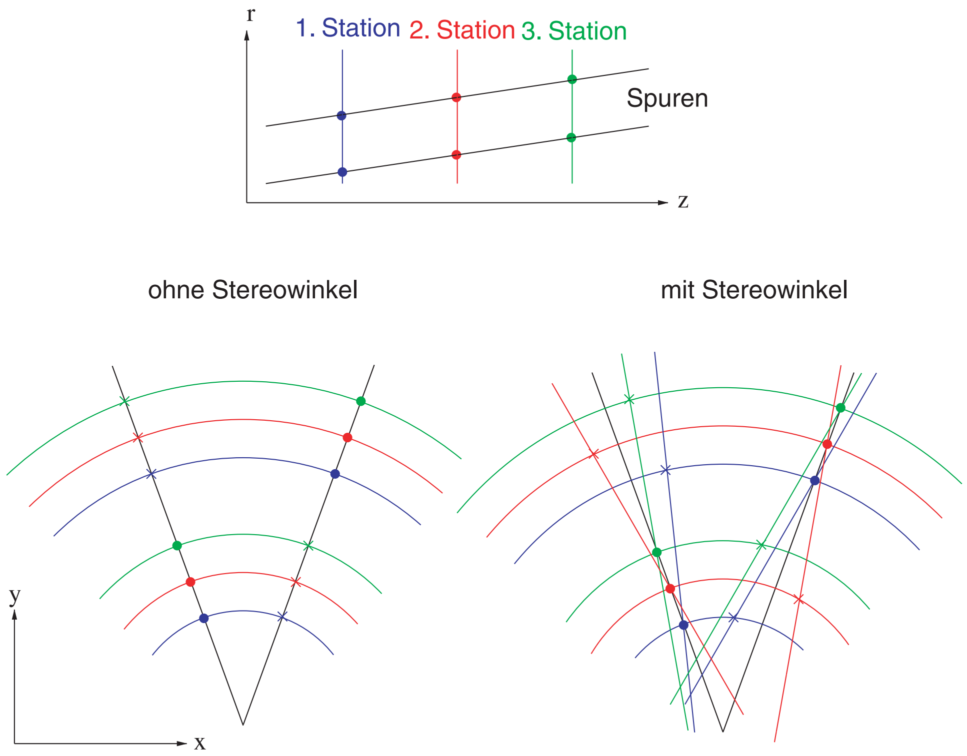


Abbildung 4.4: Effekt des Stereowinkels der Streifen in den ϕ-Detektoren. Gezeigt sind zwei Spuren (schwarz), die unter verschiedenen Winkeln ϕ die z-Achse schneiden und auf drei aufeinanderfolgenden Stationen (blau, rot, grün) Meßpunkte erzeugen (Punkte). Das obere Bild stellt eine Seitenansicht der Stationen dar. Die unteren beiden zeigen die Streifen des r- und ϕ-Detektors in der Frontansicht mit und ohne Stereowinkel, wobei die Bilder aller drei Stationen übereinandergelegt sind. Die Kreuze markieren Schnittpunkte von r- und ϕ-Streifen, die keinem Meßpunkt entsprechen.

lisiert wird, oder sich zuvor noch weiter optimieren läßt, wird derzeit in verschiedenen Studien untersucht. Beispielsweise wird überlegt, die Zahl der Stationen auf 25 zu vergrößern, wodurch die Zahl von Treffern pro Spur um 50% erhöht würde [21]. Änderungen dieser Art stellen aber kein Problem für den Algorithmus dar, weil die Positionen und Dimensionen der Stationen sowie die Zahl und Größe ihrer Sektoren als variable Parameter implementiert sind und deshalb an neue Designs angepaßt werden können.

4.2 Funktionsweise des Algorithmus

Der Algorithmus zielt darauf ab, ein Ereignis unter Verwendung der Koordinaten von r- und ϕ-Clustern in möglichst kurzer Zeit auf die Existenz von Sekundärvertices hin zu untersuchen. Als Vertices werden Wechselwirkungspunkte von Teilchen bezeichnet, wobei zwischen dem Primärvertex, der die auslösende Reaktion zweier Strahlteilchen beschreibt, und sekundären Vertices, die durch den Zerfall eines der Produkte dieser primären Wechselwirkung entstehen, unterschieden wird. Treten diese im Vertex-Detektor auf, handelt es sich mit hoher Wahrscheinlichkeit um die Zerfälle von B-Mesonen.

Die gesamte Prozedur gliedert sich in vier Abschnitte, in denen jeweils Spuren oder Vertices als Basis für die weiteren Berechnungen in den darauffolgenden Abschnitten gesucht werden:

- Suche nach zweidimensionalen Spuren (2D-Spuren) in der r/z-Projektion unter Verwendung der r-Cluster.
- Positionsbestimmung des Primärvertex mit Hilfe der 2D-Spuren.
- Berechnung von dreidimensionalen Spuren (3D-Spuren) unter Hinzunahme der ϕ-Cluster.
- Suche nach Sekundärvertices mit Hilfe der 3D-Spuren.

Diese Unterteilung ermöglicht einen frühzeitigen Abbruch, wenn anhand der Ergebnismenge sinnvolle Resultate ausgeschlossen werden können. Dadurch wird die im Schnitt benötigte Rechenzeit verringert, was wiederum die erforderlich Zahl von Rechnern reduziert. Außerdem erleichtert sie die Analyse von Zwischenergebnissen. Im folgenden werden die einzelnen Unteralgorithmen, sowie die Kriterien für eine positive Trigger-Entscheidung im Detail besprochen. Die Qualität der Ergebnisse kann entscheidend durch einige Parameter beeinflußt werden, auf die an dieser Stelle jedoch nur kurz eingegangen wird. Deren Einfluß auf die Resultate wird in Abschnitt 5.4.3 näher erläutert.

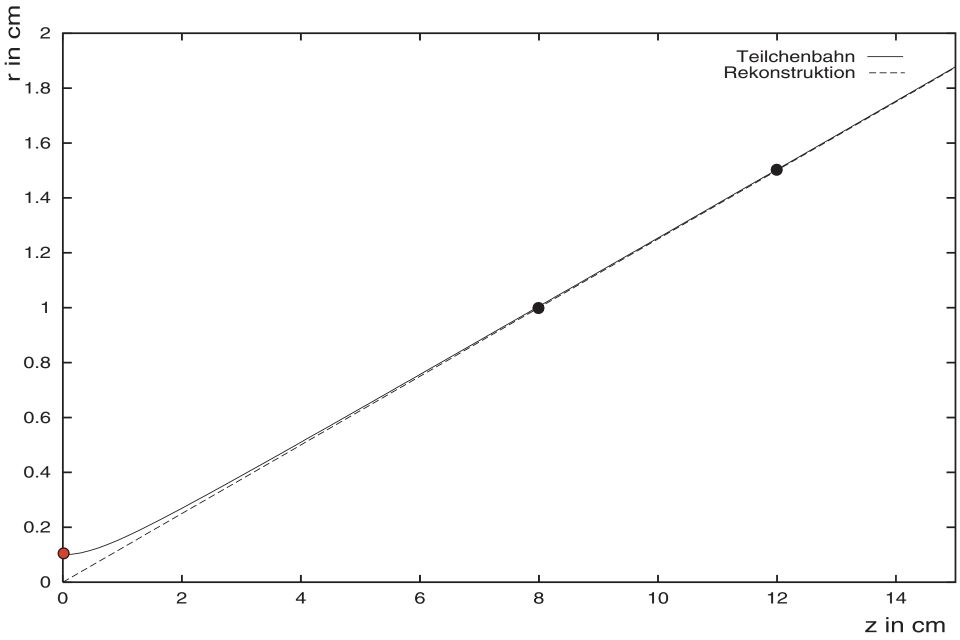


Abbildung 4.5: Bahn eines Teilchens in der r-Projektion, dessen Entstehungsvertex (rot) 1 mm von der Strahlachse entfernt liegt. Die ersten beiden Meßpunkte sind markiert.

4.2.1 Spurensuche in zwei Dimensionen

In diesem Unteralgorithmus werden ausschließlich die Koordinaten der r-Cluster verwendet, was bedeutet, daß nach Spuren in der r/z-Projektion gesucht wird. Da im Vertex-Detektor kein Magnetfeld besteht und die Teilchenbahnen deswegen — abgesehen von leichten Ablenkungen durch Wechselwirkung mit dem Detektormaterial — gerade sind, können sie durch die Position des Entstehungsvertex $\vec{v}$ und die Flugrichtung $\vec{r}$ beschrieben werden, womit sich folgende Projektion ergibt:

$$r(z) = \sqrt{\left(v_x + (z - v_z)\,\frac{r_x}{r_z}\right)^2 + \left(v_y + (z - v_z)\,\frac{r_y}{r_z}\right)^2}$$

Da der Abstand $\sqrt{v_x^2 + v_y^2}$ der Vertices von der Strahlachse klein gegen den Abstand $z - v_z$ zu den Meßpunkten ist, kann der Zusammenhang in sehr guter Näherung als linear angenommen werden (Abb. 4.5):

$$r(z) \approx (z - v_z)\sqrt{(r_x^2 + r_y^2)/r_z^2}$$

Der Algorithmus sucht deshalb in aufeinanderfolgenden ϕ-Sektoren nach

Treffern, die in der r/z-Ebene auf einer Gerade liegen. Um den Rechenaufwand möglichst gering zu halten, beschränkt sich diese Suche auf die hierfür minimal erforderliche Anzahl von drei Koordinaten. Dabei werden die Ortsauflösungen der Cluster berücksichtigt, die in der Simulation vom Untersektor bzw. Streifenabstand im Detektor und der Zahl getroffener Streifen abhängt (siehe Abschnitt 5.3). Zunächst werden die Meßpunkte der ersten beiden Stationen (r_1, z_{r_1}) und (r_2, z_{r_2}) auf die dritte linear extrapoliert

$$r_3^{extr} = r_1 + \frac{(z_{r_3} - z_{r_1})(r_2 - r_1)}{z_{r_2} - z_{r_1}}$$

um dann nach Meßpunkten (r_3, z_{r_3}) zu suchen, die im Bereich des berechneten Fehlers liegen, also folgende Bedingung erfüllen:

$$\begin{aligned} \left| r_3 - r_3^{extr} \right| &< n \cdot \sigma \\ \sigma &= \sqrt{\frac{\sigma_{r_1}^2 \left(z_{r_3} - z_{r_2}\right)^2 + \sigma_{r_2}^2 \left(z_{r_3} - z_{r_1}\right)^2}{\left(z_{r_2} - z_{r_1}\right)^2} + \sigma_{r_3}^2} \end{aligned} \tag{4.1}$$

Hierbei sind σ_{r_1,r_2,r_3} die vom Untersektor abhängigen Auflösungen der r-Koordinaten auf den drei Stationen. Die Akzeptanz kann durch den einstellbaren Parameter n gesteuert werden. Eine Gruppe aus drei solchen Clustern wird im folgenden als *Triplett* bezeichnet (Abb. 4.6). Dabei wird zwischen Vorwärts- und Rückwärts-Ttripletts unterschieden, je nachdem ob die auslösenden Teilchen in Richtung des einarmigen LHCb Detektors fliegen, oder entgegengesetzt dazu. Rückwärtige Spuren gelangen zwar nicht in das Spektrometer und werden deshalb auch nicht direkt zur Suche nach B-Zerfällen herangezogen, verbessern aber die Auflösung bei der später beschriebenen Bestimmung der Primärvertex-Position.

Neben der näherungsweisen Kollinearität werden folgende Bedingungen an drei Cluster gestellt, ehe die Suchroutine sie als Triplett akzeptiert:

- Für die Koordinaten gilt $r_1 < r_2 < r_3$ und $z_{r_1} < z_{r_2} < z_{r_3}$ (bzw. $z_{r_3} < z_{r_2} < z_{r_1}$ bei der Suche nach rückwärtigen Spuren).
- Der r-Cluster auf der ersten Station befindet sich in einem der beiden inneren Untersektoren. Dadurch wird sichergestellt, daß die höhere ϕ-Auflösung zur späteren Berechnung der x- und y-Koordinate des primären Vertex genutzt werden kann (12 innere ϕ-Untersektoren gegenüber nur 6 mittleren und äußeren). Außerdem wird der kombinatorische Aufwand bei der Suche nach Tripletts stark reduziert.

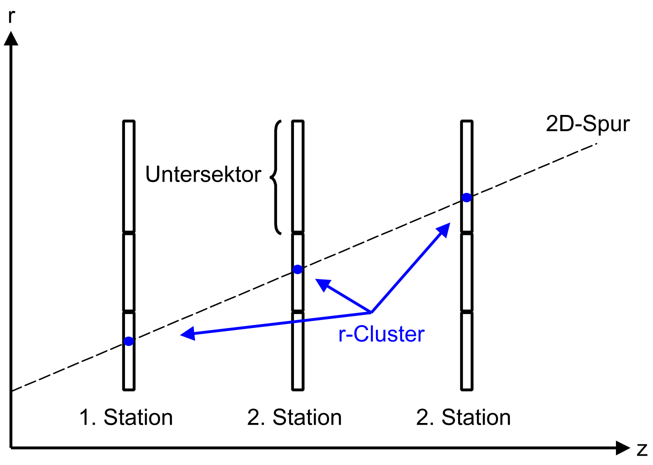


Abbildung 4.6: Triplett aus drei r-Clustern, die eine 2D-Spur bilden.

- Der r-Cluster auf der zweiten Station befindet sich ebenfalls in einem inneren oder im mittleren Untersektor. Auch hier liegt der Grund in einer weiteren Einschränkung der Zahl möglicher Kombinationen.

Die Tripletts werden durch die Steigung m und den Achsenabschnitt b charakterisiert. Die Berechnung dieser Spurparameter basiert auf den Koordinaten des ersten und dritten Clusters:

$$m = \frac{r_3 - r_1}{z_{r_3} - z_{r_1}} \tag{4.2}$$

$$b = r_1 - z_{r_1} \cdot m \tag{4.3}$$

Da viele Spuren mehr als drei Stationen treffen und somit auch mehrere Tripletts erzeugen können, sorgt eine Auswahlroutine dafür, daß jeweils nur eines pro Spur übrig bleibt. Dazu werden die Fehler der Spurparameter in Abhängigkeit von den Auflösungen der r-Koordinaten berechnet:

$$\sigma_m = \frac{\sqrt{\sigma_{r_1}^2 + \sigma_{r_3}^2}}{|z_{r_1} - z_{r_3}|}$$

$$\sigma_b = \frac{\sqrt{z_{r_3}^2 \sigma_{r_1}^2 + z_{r_1}^2 \sigma_{r_3}^2}}{|z_{r_1} - z_{r_3}|}$$

Wenn für zwei Tripletts (m_1, b_1) und (m_2, b_2) die Bedingungen

$$|m_1 - m_2| < \sigma_{m_1} + \sigma_{m_2}$$

$$|b_1 - b_2| < \sigma_{b_1} + \sigma_{b_2}$$

erfüllt sind, wird dasjenige mit größerer Entfernung zum Primärvertex entfernt, weil hier eine schlechtere Auflösung bei der Vertex-Rekonstruktion zu erwarten ist. Abbildung 4.7 zeigt alle verbleibenden Tripletts und die daraus resultierenden 2D-Spuren sowie alle übrigen r-Cluster für zwei ausgewählte Ereignisse.

4.2.2 Suche nach dem Primärvertex

Allein mit Hilfe der 2D-Spuren, die ausschließlich aus radialen Meßwerten gebildet wurden, kann bereits die Position des primären Vertex in allen drei Raumkoordinaten abgeschätzt werden. Dies wird durch die Unterteilung des inneren Bereichs der r-Detektoren in 12 Untersektoren ermöglicht (Abb. 4.2). Da der erste Meßpunkt einer Spur auf jeden Fall in diesem Bereich liegt, kann die mittlere ϕ-Koordinate des Untersektors mit einem maximalen Fehler von $\pm\pi/12$ als diejenige des r-Clusters und somit der Spur angesehen werden.

Zunächst wird jedoch die Position entlang der Strahlachse bestimmt, indem die Schnittpunkte zwischen 2D-Spuren berechnet und deren z-Koordinate in ein Histogramm eingetragen werden (Abb. 4.8). Um die Auflösung zu verbessern, sollten zwei Spuren bei dieser Kalkulation einen möglichst großen Öffnungswinkel aufweisen. Deshalb werden nur Spurpaare berücksichtigt, die sich in gegenüberliegenden ϕ-Sektoren bzw. bei Kombinationen aus Vorwärts- und Rückwärts-Spuren im selben Sektor befinden. Im Histogramm wird dann nach einem Häufungspunkt gesucht, dessen Position bestimmt und als z-Koordinate des primären Vertex festgehalten wird.

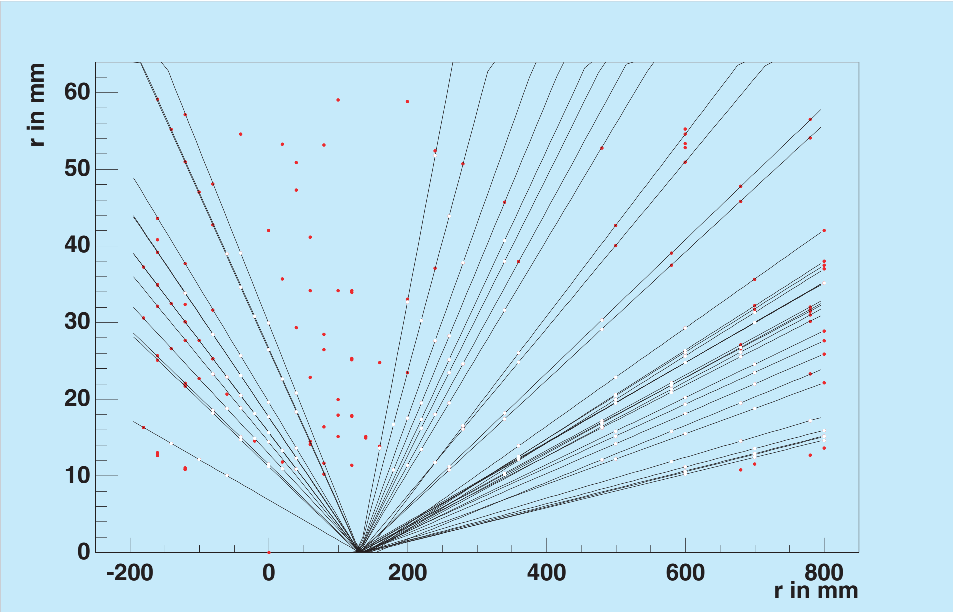
Mit Hilfe dieses Wertes kann dann auch die Lage in der x/y-Ebene ermittelt werden. Dazu wird für alle 2D-Spuren die r-Koordinate an der Stelle des primären Vertex (z_{pv}) mit Hilfe der Spurparameter m und b berechnet

$$r = m \cdot z_{pv} + b$$

und zusammen mit der abgeschätzten ϕ-Koordinate der Spur die x- und y-Koordinate ausgerechnet

$$\begin{aligned} x &= r \cdot cos\,\phi \\ y &= r \cdot sin\,\phi \end{aligned}$$

welche dann in zwei weitere Histogramme eingetragen werden (Abb. 4.9). Auch hier zeigen dann die Stellen der Häufungspunkte die entsprechenden Koordinaten des primären Vertex.



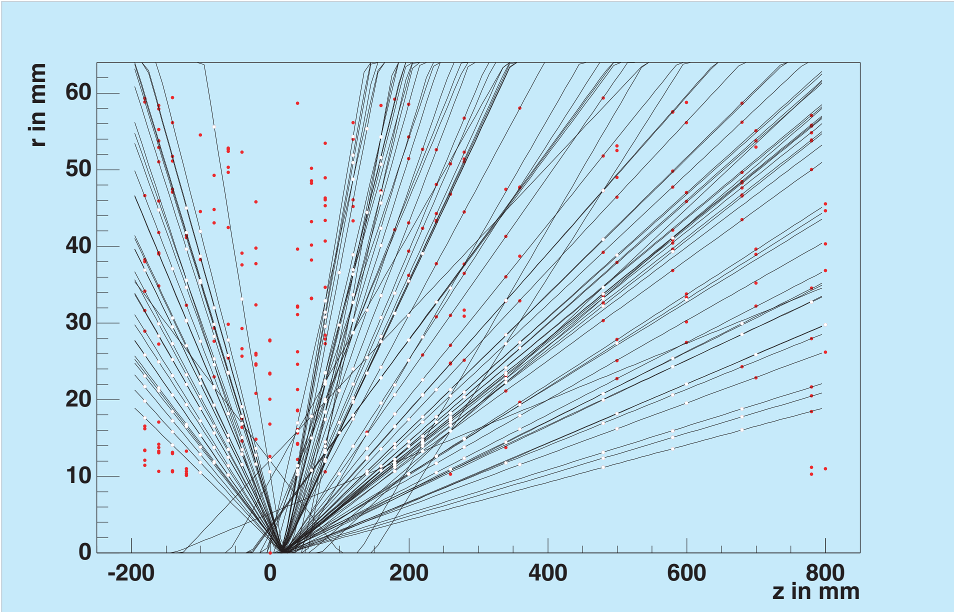


Abbildung 4.7: r-Cluster und 2D-Spuren von $B_d^0 \to \pi^+\pi^-$ Ereignissen. Gezeigt sind zwei ausgewählte Ereignisse mit niedriger und hoher Spurmultiplizität. Jeder Spur ist genau ein Triplett zugeordnet, dessen Cluster weiß markiert sind, während alle übrigen rot dargestellt sind.

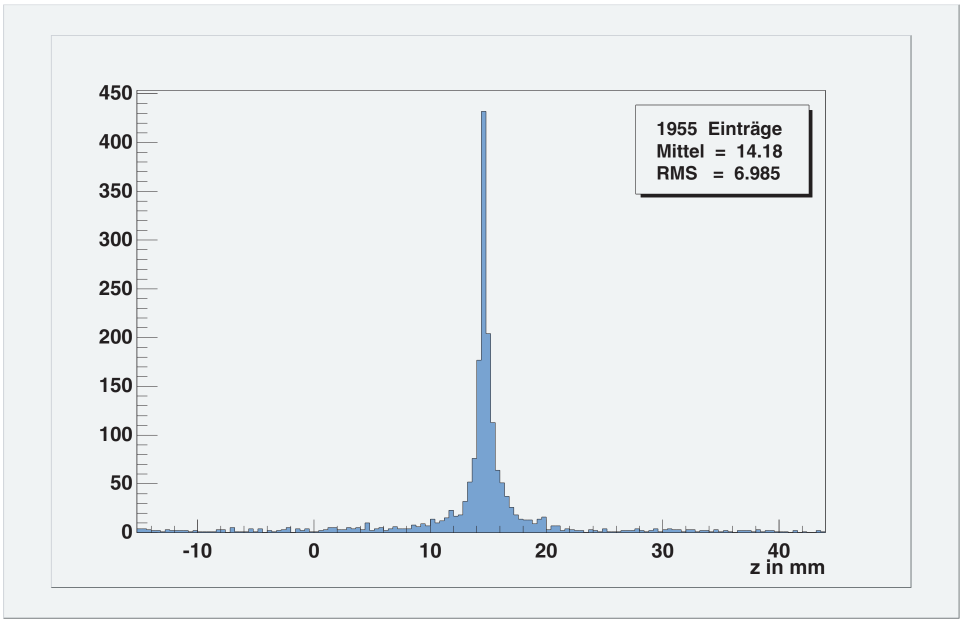


Abbildung 4.8: Histogramm zur Bestimmung der z-Koordinate des primären Vertex. Es enthält die Koordinaten der Schnittpunkte zwischen 2D-Spuren eines Ereignisses.

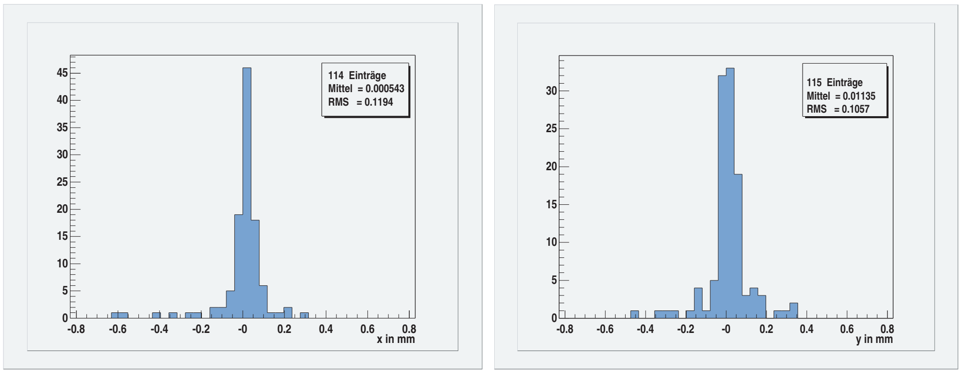


Abbildung 4.9: Histogramme zur Bestimmung der x- und y-Koordinate des primären Vertex. Sie enthalten die Koordinaten aller 2D-Spuren an der Stelle z des Primärvertex.

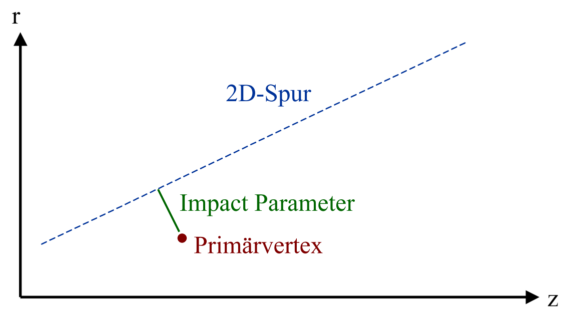


Abbildung 4.10: Geometrische Definition des Impact Parameters von 2D-Spuren.

4.2.3 Spurensuche in drei Dimensionen

Um nach sekundären Vertices im Raum suchen zu können, ist es nun erforderlich, die zu den 2D-Spuren gehörigen ϕ-Cluster zu finden, um so die volle 3D-Information zu erhalten. Ziel muß auch hier sein, den Rechenaufwand möglichst gering zu halten. Deshalb werden nur diejenigen 2D-Spuren bearbeitet, deren Ursprung mit hoher Wahrscheinlichkeit *nicht* im primären Vertex liegt. Alle übrigen Spuren sind später bei der Suche nach Sekundärvertices ohnehin überflüssig und erschweren diese sogar. Als Selektionskriterien dienen die Spurparameter m und b, hauptsächlich aber der Abstand der Spur zum Primärvertex (r_{pv}, z_{pv}), welcher im folgenden als *Impact Parameter* bezeichnet wird (Abb. 4.10) und wie folgt definiert ist:

$$ip = \frac{r_{pv} - m \cdot z_{pv} - b}{\sqrt{m^2 + 1}} \tag{4.4}$$

Für 2D-Spuren mit signifikantem Impact Parameter, muß nun das Triplett aus ϕ-Koordinaten gefunden werden, das zu demjenigen aus den r-Koordinaten paßt. Erste Bedingung hierfür ist, daß die Cluster beider Tripletts in den jeweils selben Stationen und Sektoren liegen. Alle möglichen Kombinationen aus drei solchen ϕ-Clustern werden dann zusammen mit den r-Clustern daraufhin überprüft, ob sie im Rahmen der Ortsauflösung auf einer Gerade im Raum liegen.

Hierzu muß zunächst die Stereowinkel-Korrektur für die ϕ-Koordinaten berechnet werden. Diese hängt vom Abstand zur Strahlachse ab, als welcher die Koordinate des — zunächst potentiell — korrespondierenden r-Clusters verwendet wird. Hierdurch wird die Zahl von Ambiguitäten bei der Zuordnung zwischen den Tripletts verschiedenen Typs verringert.

Unter der Annahme $\phi_1 \approx \phi_2 \approx \phi_3$, was für Spuren aus Primär- und

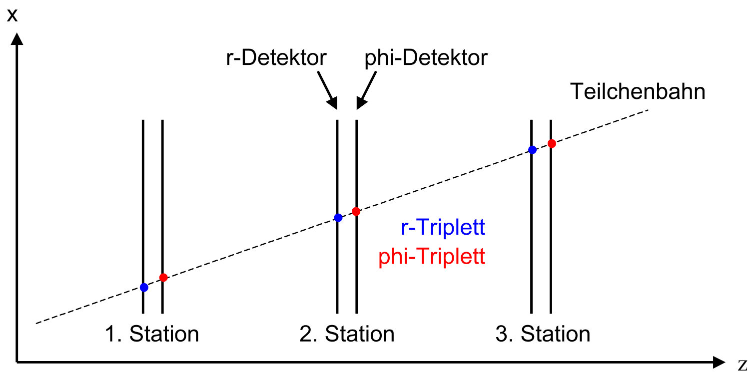


Abbildung 4.11: r- und ϕ-Triplett einer Teilchenbahn, die in der x/z-Ebene liegt.

Sekundärvertices der Fall ist, werden die Meßpunkte (ϕ_1, z_{ϕ_1}) und (ϕ_2, z_{ϕ_2}) der ersten und zweiten Station auf die dritte linear extrapoliert:

$$\phi_3^{extr} \approx \frac{(z_{\phi_3} - z_{\phi_1})(\phi_2 - \phi_1) r_{\phi_2}}{(z_{\phi_2} - z_{\phi_1}) r_{\phi_3}} + \phi_1$$

Dabei sind r_{ϕ_2} und r_{ϕ_3} die auf die zweite und dritte Station extrapolierten r-Koordinaten der 2D-Spur (Abb. 4.11). Die tatsächlich vorhandenen Cluster (ϕ_3, z_{ϕ_3}) müssen dann die folgende Bedingung erfüllen:

$$\begin{aligned} \left|\phi_3^{extr} - \phi_3\right| &< n \cdot \sigma \qquad (4.5) \\ \sigma &= \sqrt{\left(1 - \frac{(z_{\phi_3} - z_{\phi_1}) r_{\phi_2}}{(z_{\phi_2} - z_{\phi_1}) r_{\phi_3}}\right)^2 \sigma_{\phi_1}^2 + \left(\frac{(z_{\phi_3} - z_{\phi_1}) r_{\phi_2}}{(z_{\phi_2} - z_{\phi_1}) r_{\phi_3}}\right)^2 \sigma_{\phi_2}^2 + \sigma_{\phi_3}^2} \end{aligned}$$

Bei $\sigma_{\phi_1,\phi_2,\phi_3}$ handelt es sich um die Ortsauflösungen der ϕ-Koordinaten. Grundsätzlich hängt σ auch von den Auflösungen der r-Koordinaten ab; es hat sich jedoch gezeigt, daß die entsprechenden Terme nur einen verschwindend kleinen Beitrag leisten und deshalb vernachlässigt werden können. Wie bei der Suche nach 2D-Spuren kann auch hier die Akzeptanz über den Parameter n variiert werden. Wenn mehrere ϕ-Tripletts die Kriterien erfüllen, wird dasjenige mit dem kleinsten Fehler (Gl. 4.5) verwendet. Als Spurparameter werden der Ortsvektor des ersten und der Verbindungsvektor zwischen erstem und letztem ϕ-Cluster gespeichert.

4.2.4 Suche nach Sekundärvertices

Unter den 3D-Spuren gilt es nun solche zu finden, die einen gemeinsamen Vertex haben. Im Idealfall wäre hierfür Voraussetzung, daß sich solche Spuren in einem Punkt schneiden. Da man davon in der Praxis nicht ausgehen kann, werden sämtliche Spuren kombiniert und daraufhin untersucht, ob deren geringste räumlich Entfernung einen als Parameter vorgegebenen Wert unterschreitet. Ist dies der Fall, wird der Mittelpunkt der Verbindungsgerade dieses geringsten Abstandes als Vertexposition angenommen. Durch einen Vergleich mit der Position des primären Vertex muß dann noch überprüft werden, ob es sich tatsächlich um einen Sekundärvertex handelt. Dazu wird der Abstand zwischen diesen beiden Punkten berechnet, der einen Schwellwert nicht unterschreiten darf.

Der Zerfall eines B-Mesons in mehr als zwei geladene Teilchen kann durch diese Methode in mehr als einem gefundenen Sekundärvertex resultieren, weil alle möglichen Kombinationen aus zwei dieser Spuren einen solchen erzeugen könnten. Auf eine Routine zur Entfernung dieser mehrfach vorhandenen Vertices wird jedoch verzichtet, denn die größere Anzahl gefundener Sekundärvertices kann als Maß für die Wahrscheinlichkeit verstanden werden, die die tatsächliche Existenz beschreibt.

4.2.5 Die Trigger-Entscheidung

Die Trigger-Entscheidung wird anhand der Zahl gefundener Sekundärvertices getroffen, die einen festgesetzten Wert erreichen muß, damit ein Ereignis akzeptiert wird. Es bestehen jedoch weitere Variablen, die der Algorithmus benutzt, um die Berechnungen bereits vor Ablauf der gesamten Prozedur abbrechen zu können, wenn sinnvolle Ergebnisse bereits frühzeitig ausgeschlossen werden können. Das Ereignis wird dann nicht weiter verfolgt. Solche Schwellen existieren für die Anzahl folgender Objekte:

- r-Cluster des gesamten Ereignisses
- Gefundene 2D-Spuren
- Einträge in den Histogrammen zur Bestimmung der Koordinaten des Primärvertex
- Gefundene 3D-Spuren

Die Level-1 Entscheidung wird im experimentellen Einsatz als Empfehlung an die *Decision Unit* des Trigger-Systems weitergeleitet und zusammen mit den Resultaten der anderen Trigger ausgewertet.

Kapitel 5

Implementierung

Die folgenden Abschnitte widmen sich der konkreten Implementierung des in Kapitel 4 beschriebenen Algorithmus im Rahmen der für solche Zwecke vorgesehenen Programmierumgebung GAUDI, sowie der Erzeugung von simulierten Ereignisdaten. Das daraus resultierende Programm zusammen mit den Simulationsdaten stellt die Quelle für die in Kapitel 6 vorgestellten Ergebnisse dar.

5.1 Entwicklungsumgebung

Die LHCb Software für physikalische Analysen und Detektoroptimierungen basiert zum gegenwärtigen Zeitpunkt auf Fortran, der bislang am häufigsten verwendeten Programmiersprache im Bereich naturwissenschaftlicher Arbeiten. Der Umstieg auf eine neue, objektorientierte Technologie ist jedoch einer der Hauptpunkte der Software-Strategie für LHC Experimente [23].

Die grundsätzlichen Ziele dieser Strategie sind die Wiederverwendbarkeit von Programmcode und die Entwicklung im Rahmen einer einheitlichen Architektur. Die meisten physikalischen Anwendungen tendieren durch den Einbau unzähliger Funktionalitäten zu Übergröße und Unübersichtlichkeit. Die Erhaltung, Wartung und Weiterentwicklung solcher monolitischer „Ungetüme" kostet die Programmierer viel Zeit und erfordert umfangreiche Kenntnisse des gesamten Software-Pakets. Da die Ansprüche an die Software jedoch ständig steigen — die Experimente werden immer komplexer, die Datenmengen immer größer, die Zahl der beteiligten Entwickler wächst ständig — die Geschwindigkeit der Entwicklung jedoch nur auf Kosten der Qualität erhöht werden könnte, muß versucht werden, die Produkte solcher Bemühungen effektiver zu nutzen.

Wiederverwendbare Komponenten können z.B. graphische Ereignisanzeigen, die sowohl im Online- als auch im Offline-Betrieb zum Einsatz kommen, sein oder verallgemeinerte Geometriebeschreibungen für sämtliche Teildetektoren, die dann in der Simulation und Rekonstruktion Verwendung finden. Besonders auch der Zugriff auf Ereignis- und Rekonstruktionsdaten sowie deren Speicherung bieten sich für solch ein Konzept an.

Die Grundlage für diese Art der effektiven Nutzung von Quell-Codes wird durch eine Architektur geschaffen, die von System-Architekten mit projektübergreifenden Kenntnissen festgelegt wird. Inhalt der Architektur sind die Komponenten, durch die die Wiederverwendbarkeit von Code verwirklicht werden soll. Außerdem ein Sammlung von Programmbibliotheken, die zum Beispiel allgemeine Funktionen zu Berechnungen im Rahmen der Hochenergiephysik bereit stellen. Schließlich stellt die Architektur ein *Framework* zur Verfügung. Als solches wird ein Rahmenprogramm bezeichnet, das eine vorgefertigte Sammlung von Bausteinen zur Nutzung, Erweiterung oder Anpassung an spezielle Aufgaben durch Programmierer darstellt. Dem Entwickler physikalischer Algorithmen bleibt so die lästige Aufgabe erspart, jede Anwendung von Grund auf neu zu programmieren, indem ihm Standardaufgaben wie Ein- und Ausgabe, Fehlerbehandlung, Datenverwaltung oder Visualisierungsmechanismen, aber auch die Grundstruktur des Programms vom Framework abgenommen werden. Ein weiterer großer Vorteil liegt in der Wartung, da Änderungen am System in der Regel lediglich Anpassungen des Frameworks erfordern, der Code mit physikalischem Hintergrund hingegen unverändert weiter genutzt werden kann. Das bedeutet für den Physiker, daß er seinen Code grundsätzlich unabhängig von Plattform und Betriebssystem entwickeln und sich ganz auf die eigentliche Aufgabe konzentrieren kann.

All dies läßt sich am besten durch eine objektorientierte Programmiersprache realisieren, da hier durch Klassendefinitionen leicht die Implementierung von Schnittstellendefinitionen getrennt werden kann, was die unbedingte Voraussetzung für wiederverwendbarem Code ist. Es ist außerdem sinnvoll, sich am allgemeinen Trend in der Computer-Industrie zu orientieren, die ebenfalls stärker auf objektorientierte Technologien setzt. Dies erleichtert den Einsatz von Fachkräften und Mitverwendung kommerzieller Software.

Gegenwärtig kommen auf Grund ihrer Reife, Verbreitung und Akzeptanz lediglich zwei objektorientierte Programmiersprachen in Frage: C++ und Java. Für C++ spricht momentan die größere Beliebtheit bei wissenschaftlichen Programmierern, vor allem aber der Geschwindigkeitsvorteil. Dieser begründet sich in der Tatsache, daß durch das Konzept der Plattformunabhängigkeit von Java dessen Code zur Laufzeit in Maschinencode übersetzt werden muß, während dies durch C++ bereits beim Kompilieren geschieht.

5.2 GAUDI

Auf Grund der im vorangehenden Abschnitt angesprochenen Aspekte hat die LHCb Collaboration der Entwicklung und Implementierung einer integrierten Software-Umgebung zur Simulation, Rekonstruktion und Analyse physikalischer Daten in C++ zugestimmt. Seit Februar 1999 steht mit dem GAUDI[1] Data Processing Applications Framework eine solche Umgebung in eingeschränktem Funktionsumfang zur Verfügung. Die vorliegende Arbeit wurde unter Einsatz der Version 4 angefertigt [26].

GAUDI ist ein Framework zur Entwicklung von Anwendungen zur Datenverarbeitung am LHCb Experiment. Ziel ist die Entwicklung einer Anwendungsumgebung, die in möglichst vielen Ebenen der Datenverarbeitung eingesetzt werden kann. Nach Möglichkeit soll das Framework allen Phasen der Datenanalyse als Umgebung dienen: Simulation der Physik und der Detektoren, die Software der Highlevel-Trigger sowie Programme zur Rekonstruktion, physikalischen Analyse und Visualisierung. Auch die Unabhängigkeit von der Einsatzumgebung soll gewährleistet sein. Dies bedeutet, daß die Algorithmen unabhängig davon geschrieben werden, ob sie interaktiv, online oder offline bzw. unter welchem Betriebssystem oder auf welcher Plattform sie laufen. Letzteres ist besonders wichtig, da momentan noch nicht abzusehen ist, welche Technologie zu Beginn des Experiments zum Einsatz kommen wird. Anpassungen sämtlichen Codes der Anwendungen an die sich ständig weiterentwickelnde Hardware kann so vermieden werden.

5.2.1 Struktur des Frameworks

Ein Framework ist die Implementierung einer Architektur, weshalb bei deren Entwicklung auf die Unabhängigkeit von der Vielzahl von Umgebungen und Phasen der Datenanalyse besonderer Wert gelegt wird. Im allgemeinen beschreibt die Architektur die Strukturen des ganzen Systems, die aus den Software-Komponenten, deren externen Eigenschaften sowie ihren Beziehungen untereinander bestehen. Das Framework stellt eine Sammlung von Klassen zum Zugriff auf die Funktionen eines bestimmten Bereichs zur Verfügung, die vom Benutzer bzw. von dessen Anwendung verwendet oder angepaßt werden können. Die Komponenten sind wiederverwendbare Software-Teile mit wohldefinierten und öffentlich zugänglichen Schnittstellen, die eine begrenzte Zahl von Funktionen zur Verfügung stellen. Software-Komponenten folgen Standardkonventionen, um die Wiederverwendbarkeit zu gewährleisten.

[1] Namensgeber ist der spanische Architekt Antoni Gaudì i Cornet (1852-1926), der durch seinen phantasievollen Baustil als Pionier der modernen Architektur gilt. Das Logo des Frameworks stellt sein berühmtestes Werk dar, die Kirche Sagrada Famìlia aus Barcelona.

Trotz des Vorhabens, ein objektorientiertes System zu entwerfen, wird in GAUDI klar zwischen Daten und Algorithmen unterschieden. Die Methoden von Daten-Objekten beschränken sich hauptsächlich auf den Zugriff zu den internen Daten. Die Algorithmus-Objekte arbeiten mit diesen Daten und erzeugen neue. Im allgemeinen wird ein Algorithmus Daten eines Typs einlesen und daraus solche eines anderen Typs berechnen.

Eine weiteres Anliegen der Entwickler ist die strikte Trennung zwischen Daten, die permanent gespeichert werden und solchen, die nur temporär im Speicher für den Zugriff durch Algorithmen bestehen. Diese werden als *Persistant Data* bzw. *Transient Data* bezeichnet. Die jeweils zugehörigen Datenablagen heißen entsprechend *Persistant/Transient Data Store*. Algorithmen sollten nie direkt auf die festgespeicherten Daten zugreifen, sondern allein mit den Übergangsdaten aus dem Transient Store arbeiten. Er dient also lediglich der Übergabe von Daten zwischen verschiedene Teilen einer Anwendung und minimiert so die Abhängigkeit verschiedener Algorithmen voneinander, was die parallele Arbeit mehrerer Entwickler stark erleichtert. Die weiteren Gründe für die klare Trennung sind im wesentlichen folgende:

- Die Entwicklung von Algorithmen erfolgt unabhängig von der verwendeten Technologie zur Speicherung der Daten. Dies ist besonders sinnvoll in Hinblick auf den angestrebten Wechsel von ZEBRA[2] auf ein objektorientiertes Datenbanksystem.
- Die Kriterien zur Optimierung des Zugriffs auf persistente und temporäre Daten unterscheiden sich grundlegend. Bei der Festspeicherung von Daten gilt es, die Ein-/Ausgabegeschwindigkeit durch kompakte Datenpakete und Vermeidung von Datenduplizitäten zu erhöhen. Bei Daten im Transient Store hingegen sind schnelle Ausführungszeiten und übersichtliche Zugriffsmöglichkeiten das Ziel.
- Der Transient Store kann als „Software Bus“ zwischen voneinander unabhängigen Komponenten genutzt werden. Dadurch benötigen externe Komponenten lediglich eine Schnittstelle zum Transient Store um mit den vorhandenen Komponenten zusammenarbeiten zu können, wie z.B. der persistenten Speicherung von Daten.

Bei beiden Formen der Datenspeicherung wird mit drei Hauptkategorien von Datenobjekten gearbeitet. Zum einen mit Ereignisdaten, die den Resultaten von Teilchenkollisionen und deren Folgeprodukten entsprechen (Rekonstruktion, Analyse, etc.), zum anderen mit Detektordaten, die alle zur Interpretation der Ereignisdaten erforderlichen Informationen enthalten

[2] Paket zur dynamischen Speicherverwaltung und Ein-/Ausgabe für Fortran

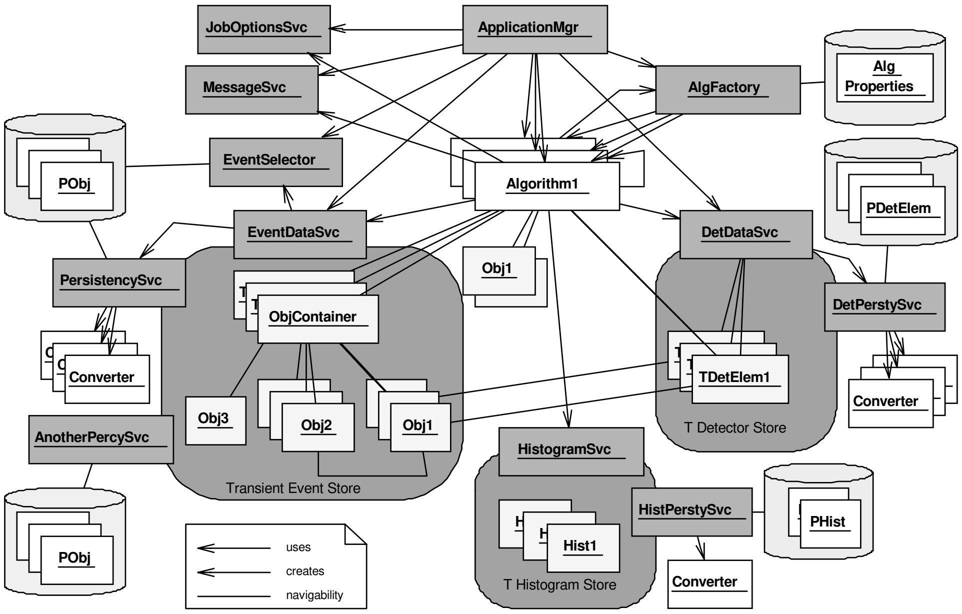


Abbildung 5.1: Objekt-Diagramm der GAUDI-Architektur. [25]

wie Struktur, Geometrie, Kalibration und Ausrichtung der Detektoren sowie Parameter zur Beschreibung von Umwelteinflüssen; und schließlich mit Statistikdaten für Analysezwecke (z.B. Histogramme und N-Tuple).

5.2.2 Komponenten

Alle Komponenten der Architektur bilden wohldefinierte Schnittstellen, um mit anderen Komponenten zu kommunizieren. Diese werden als abstrakte C++-Klassen realisiert. Abbildung 5.1 zeigt einen Überblick der GAUDI-Objekte.

Der *Application Manager (ApplicationMgr)* steuert die Ausführung der Anwendung, die als *Job* bezeichnet wird. Er erstellt alle erforderlichen Dienste und Komponenten und implementiert die Ereignis-Schleife, in der die einzelnen Teile initialisiert und dann die Ausführung der Algorithmen für jedes Ereignis kontrolliert wird. Die Schnittstelle, die alle Algorithmen zur Verfügung stellen müssen, ist das *Algorithm Interface (IAlgorithm)*. Dies besteht im wesentlichen aus drei Funktionen, die vom Application Manager aufgerufen werden, nämlich die Initialisierung zu Beginn eines Jobs *(initialize)*, die Datenverarbeitung einmal pro Ereignis *(execute)* und abschließenden Aufgaben zum Ende des Jobs *(finalize)*. Die Aufgabe von Algorithmus-Komponenten ist im allgemeinen, eine Berechnung auf Eingabedaten durchzuführen und Ausgabedaten zu produzieren.

Der Zweck des *Job Options Service (JobOptionsSvc)* ist es, die Optionen für einen Job an die Komponenten der gesamten Architektur zur verteilen. Bei diesen handelt es sich zum Beispiel um variable Parameter von Algorithmen, die Anzahl der zu bearbeitenden Ereignisse oder Pfade benötigter Laufzeitbibliotheken. Sie sind in einer Text-Datei zusammengefaßt, und können dort vom Benutzer eingestellt werden.

Der *Transient Data Store* ist eine passive Komponente der Architektur, in der Datenobjekte während der Ausführung von Algorithmen temporär gespeichert werden können. Es handelt sich dabei um eine rein logische Baumstruktur, in der Datenobjekte letztendlich nur registriert und nicht hineinkopiert werden. Für die Verwaltung der unterschiedlichen Datentypen ist jeweils eine weitere Komponente verantwortlich. Der *Event Data Service (EventDataSvc)* verwaltet die Ereignisdaten. Er erlaubt es dem Application Manager und den Algorithmen Datenobjekte zu lokalisieren und neue zu registrieren. Die entsprechenden Dienste für Detektordaten und statistische Objekte sind der *Detector Data Service (DetDataSvc)* bzw. *Histogram Data Service (HistogramSvc)*.

Alle vorübergehenden Datenobjekte, die zum Zweck des Austauschs zwischen verschiedenen Komponenten im Transient Store gespeichert werden, sind von einer *Data Object (DataObject)* Basisklasse abgeleitet. Dadurch sind auch die Schnittstellen zu den Diensten standardisiert, mit denen der Datenaustausch über den Transient Store abgewickelt wird. Außerdem wird dessen Verwaltungsstruktur erzwungen.

Die Verantwortung für den Transfer von Ereignisdaten zwischen dem Transient Data Store und permanenten Speichern trägt der *Event Persistency Service (PersistencySvc)*. Er kooperiert eng mit dem Event Data Service für den Fall, daß Daten angefordert werden, die sich noch nicht im Transient Store befinden und deshalb geladen und konvertiert werden müssen. Die Übersetzung von Daten zwischen diesen verschiedenen Formaten geschieht durch Konverter, die auf dem *Converter Interface* basieren. Auch hier existieren die aufgabengleichen Dienste für den Transfer von Detektordaten im *Detector Persistency Service (DetPerstySvc)* und von Statistikdaten im *Histogram Persitancy Service (HistPerstySvc)*.

Der *Event Selector (EventSelctor)* wird dazu verwendet, unter den verfügbaren Ereignissen diejenigen auszuwählen, die vom Application Manager und den von ihm aufgerufenen Algorithmen bearbeitet werden. So muß dem Event Selector beispielsweise der Pfad einer Datei mit den Daten simulierter Ereignisse und die Nummer des ersten zu bearbeitenden Ereignisses mitgeteilt werden.

Über den *Message Service (MessageSvc)* können Informations- oder Fehlermeldungen erzeugt werden, die je nach Konfiguration auf dem Bildschirm angezeigt oder in eine Datei geschrieben werden. Die Nachrichten erhalten einen Prioritätslevel *(Verbose, Debug, Info, Warning, Error, Fatal)*, mit dessen Hilfe die Ausgabe von Meldungen niedriger Priorität unterdrückt werden kann.

5.3 Ereignis-Simulation

Da bislang noch keine Meßwerte von LHCb zur Verfügung stehen und mit solchen auch nicht vor dem Jahr 2005 zu rechnen ist, stützt sich diese Arbeit ausschließlich auf simulierte Daten. Im Experiment werden Ereignisse durch den Teilchenbeschleuniger erzeugt, von den Detektoren gemessen und vom Datenerfassungssystem auf Band geschrieben, wenn für die Auswertung interessante Informationen vermutet werden. Später werden diese Ereignisse dann rekonstruiert, indem die elektronischen Signale der Detektoren in Energien, Impulse und Spuren von Teilchen übersetzt werden. Sodann kann mit der physikalischen Analyse begonnen werden. In der Simulation übernimmt ein Ereignis-Generator die Rolle des Speicherrings und berechnet alle entstehenden Teilchen der primären Wechselwirkungen in einer Strahlkollision. Das Verhalten der Teilchen in den Detektoren und deren Signale werden dann von einem Detektorsimulationsprogramm berechnet. Dessen Ausgabedaten entsprechen in ihrem Format dann denjenigen, die auch im Experiment von der Auslese-Elektronik erwartet werden und können deshalb auch in dieselbe Rekonstruktions- und Analysekette eingespeist werden — mit dem Vorteil, daß das korrekte Ergebnis bereits vorab bekannt ist.

Für das LHCb Experiment existiert das Monte-Carlo-Simulationspaket SICb [30], das auf der Detektorsimulation GEANT basiert [32]. GEANT bietet eine allgemeine Infrastruktur zur Beschreibung von Detektorgeometrien und -materialien. Es simuliert das Verhalten von Teilchen innerhalb des beschriebenen Versuchaufbaus, also deren Bewegung, Zerfall und Wechselwirkung mit Materie, und die Signale der sensitiven Teile des Detektors. Außerdem besteht die Möglichkeit, den Detektoraufbau und Teilchenspuren sichtbar zu machen. Als Ereignisgeneratoren werden die Programme PYTHIA und JETSET verwendet [33].

Die Simulation des Vertex-Detektors im Rahmen von SICb umfaßt das Material, die Geometrie, die Antwort des Detektors auf Signale, die Digitalisierung dieser Signale und die Rekonstruktion von Clustern aus den Signalen. Als Cluster werden ein oder mehrere Detektorsignale zusammengefaßt, die mutmaßlich vom selben Teilchen erzeugt wurden und aus denen eine gewichtete Koordinate berechnet wird. Im folgenden wird daher von r- und ϕ-Clustern die Rede sein, was letztendlich Meßwerten von r- und ϕ-Koordinaten entspricht. Die Ereignisdaten und daraus rekonstruierte

Anzahl getroffener Streifen	1	2	$\geq$3
Innerer r-Detektor	9 μm	6 μm	20 μm
Mittlerer r-Detektor	13 μm	10 μm	30 μm
Äußerer r-Detektor	18 μm	10 μm	40 μm
ϕ-Detektor	$\frac{\text{LokalerStreifenabstand}}{\sqrt{12}}$		

Tabelle 5.1: Ortsauflösungen der r- und ϕ-Detektoren, die im Rahmen der Simulation verwendet werden. [31]

Objekte werden in ZEBRA-Banken gespeichert. Auf die wichtigsten kann innerhalb von GAUDI zugegriffen werden; für sie existiert jeweils ein Konverter, der die Daten von ZEBRA in ein spezielles Format umwandelt, welches durch eine C++-Klasse beschrieben wird, über die auch der Zugriff auf die in der Bank enthaltenen Informationen erfolgt.

Der Trigger-Code verwendet die Banken VSCR (r-Cluster) und VSCP (ϕ-Cluster), um Ereignisdaten, wie sie von den Auslese-Einheiten erwartet werden, zu erstellen. Zur Analyse der simulierten Ereignisse und Bewertung der Ergebnisse werden des weiteren die Bänke ATMC (Monte-Carlo-Spuren) und AVMC (Monte-Carlo-Vertices) benutzt. Der Zugriff erfolgt über die GAUDI-Klassen `VeloCluster`, `MCParticle` und `MCVertex`. Detaillierte Beschreibungen der Banken und der darin enthaltenen Werte können unter [30] gefunden werden.

Als Ortsauflösung dieser simulierten Meßpunkte werden die in Tabelle 5.1 zusammengestellten Werte verwendet [31]. Erzeugt ein Teilchen Signale auf mehr als einem Streifen, kann durch die räumliche Gewichtung der Signalstärken ein bessere Auflösung erzielt werden. Bei mehr als zwei Treffern muß allerdings die höhere Wahrscheinlichkeit berücksichtigt werden, daß die Ursache des Clusters in mehreren Spuren liegt. Bei ϕ-Clustern wird auf eine solche Differenzierung verzichtet, weil es sich dabei fast immer um Einzeltreffer handelt.

5.4 Der Algorithmus

Der gesamte Level-1 Vertex-Trigger-Algorithmus ist als Anwendung unter GAUDI implementiert. Neben dem Trigger-Code enthält diese Anwendung Möglichkeiten zur Analyse der Monte-Carlo-Daten, die dabei helfen sollen, den Algorithmus zu kalibrieren und optimieren. Die folgenden Abschnitte beschreiben alle vorhandenen Unteralgorithmen, deren Zusammenspiel im Rahmen des Frameworks, das verwendete Format für die Ereignisdaten sowie die Möglichkeiten zur Konfiguration.

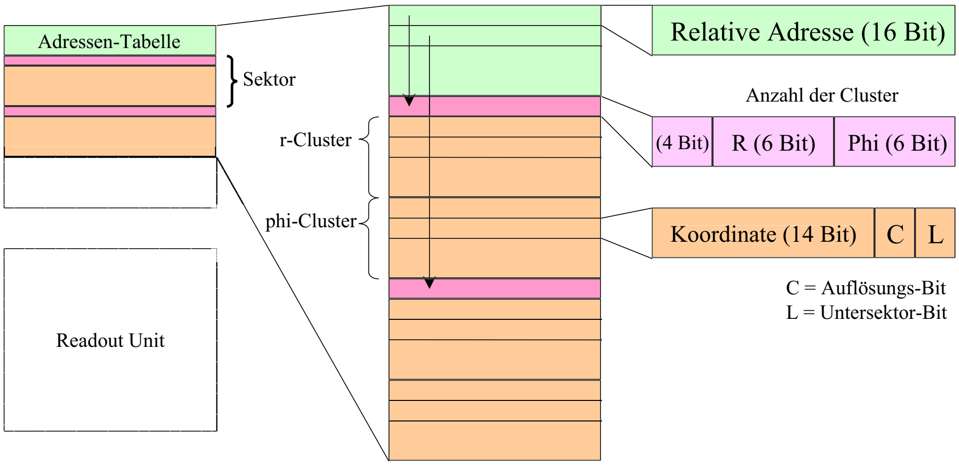


Abbildung 5.2: Format der Meßwerte im Speicher.

5.4.1 Das Format der Ereignisdaten

Das Ziel, die Umsetzung des Algorithmus für den Einsatz sowohl in simulierter als auch in realer Umgebung auszulegen, erfordert die Spezifikation des Formats, in dem die Meßwerte der Detektoren von der Hardware zur Verfügung gestellt werden. Dies wird durch die Readout Units geschehen, welche die Koordinaten aller rekonstruierter Cluster und einige weitere Infomationen direkt in den Speicher der Computer des Level-1 Triggers schreiben. Zum gegenwärtigen Zeitpunkt, zu dem die Hardware noch in Entwicklung ist, wird an dieser Stelle die Möglichkeit geboten, mit verschiedenen Formaten zu experimentieren und deren Einfluß auf die Zugriffsgeschwindigkeit durch die Routinen des Trigger-Codes zu untersuchen. Der momentane Stand orientiert sich an den vorgeschlagenen Schnittstellen Spezifikationen für die Off-Detector Electronic (ODE) [29]. Das verwendete Format ist in Abb. 5.2 schematisch dargestellt.

Jede Readout Unit ist für die Daten einer bestimmten Anzahl von Sektoren der Stationen des Vertex-Detektors verantwortlich. Die Zuordnung ist — zumindest was den Trigger-Code angeht — beliebig in Anzahl und Anordnung der Sektoren. Die Meßdaten dieser Einheiten werden an feste Adressen im physikalischen Adreßraum geschrieben. Der Abstand zwischen den Adressen ist konstant und richtet sich nach der maximal möglichen Datenmenge. Jeder dieser Blöcke beginnt mit einer relativen Adressen-Tabelle, in der für jeden Sektor dieser Einheit ein Eintrag existiert. Die Adressen zeigen auf den Beginn der Daten des zugehörigen Sektors, die in unmittelbarer Folge nach der Tabelle abgelegt sind. Die Daten eines Sektors wiederum beginnen mit einem Datenwort, in dem die Anzahl der für dieses Ereignis von dem entsprechenden Sektor gemessenen r- und ϕ-Cluster gespeichert wird.

Diese Meßwerte folgen als 16-Bit Werte. Der Sinn dieser Anordnung liegt darin, den Zugriff zu beschleunigen. Da die Zahl der gemessenen Cluster pro Sektor über mehrere Ereignisse hinweg nicht konstant bleibt, müßte man ohne Offset-Tabelle zum Zugriff auf Werte eines Sektors mit höherem Index zunächst die Datengröße sämtlicher davor liegender Sektoren auslesen, um die Startadresse der gewünschten Daten berechnen zu können.

Ein weiteres Ziel neben einer hohen Zugriffsgeschwindigkeit durch die Funktionen des Trigger-Codes muß sein, die gesamte Datenmenge pro Ereignis möglichst klein zu halten, um die Transferzeiten des Netzwerks zu reduzieren. Daher wird versucht, die komplette Cluster-Information in einem 16-Bit Wort unterzubringen. Davon werden 14 Bit für die Koordinate verwendet. Bei der gegenwärtig geplanten Gestaltung des Vertex-Detektors (Abschnitt 4.1) liegen die r-Koordinaten im Bereich von 1–6 cm, was dem inneren und äußeren Radius einer Station entspricht. Für diesen Bereich von 5 cm ergibt sich bei einem ganzzahligen 14-Bit Wert eine Auflösung von etwa 3 μm. Da allein die Ortsauflösung der Detektoren im günstigsten Fall 6 μm beträgt (Tab. 5.1), ist dadurch kein merklicher Informationsverlust zu erwarten. Für die ϕ-Koordinaten gilt entsprechend bei einer Spanne von 61° pro Sektor als Abstufung 0,065 mrad gegenüber einer Winkelauflösung von etwa 0,5 mrad am inneren Rand des äußeren Untersektors. Der Algorithmus geht davon aus, daß die Cluster nach der Größe ihrer Koordinaten sortiert sind. Dadurch wird die Suche nach Tripletts entscheidend beschleunigt, denn wenn auf einer Station der erste Cluster mit zu großer Koordinate erreicht wird, brauchen alle folgenden nicht mehr überprüft zu werden.

Von den verbleibenden zwei Bits dient eines zur Unterscheidung verschiedener Ortsauflösungen. Diese hängt nicht nur vom Streifenabstand im Untersektor ab, sondern auch von der Zahl der getroffenen Streifen. Grundsätzlich verbessert sich die Auflösung, wenn durch ein Teilchen mehrere Signale erzeugt werden. Bei größeren Clustern überwiegt jedoch die Wahrscheinlichkeit, daß sie durch mehrere Spuren erzeugt wurden. Diesem Umstand wird durch Annahme einer schlechteren Auflösung Rechnung getragen. Da später in der Experimentalumgebung noch weitere Fehler hinzukommen, die z.B. durch die Ausrichtung des Detektors entstehen, reicht es, die unterschiedlichen Qualitäten der Cluster-Koordinaten in nur zwei Bereiche aufzuteilen. Für die ϕ-Koordinaten wird darauf ganz verzichtet (siehe Abschnitt 5.3). In der vorliegenden Fassung wird daher zwischen Clustern mit nur einem oder zwei bzw. drei oder mehr Treffern unterschieden. Das letzte Bit spezifiziert zusammen mit der Koordinate den Untersektor. Im Falle der r-Detektoren kann so zwischen den beiden inneren Untersektoren unterschieden werden (wodurch die Genauigkeit der ϕ-Information in den r-Clustern verdoppelt wird) und bei ϕ-Clustern zwischen innerem und äußerem Untersektor (zusätzliche r-Information).

Der Zugriff oder Verweis auf bestimmte r- oder ϕ-Cluster im Trigger-Code erfolgt durch Angabe dreier Indizes, nämlich dem der Station, des Sektors innerhalb der Station und des Clusters innerhalb des Sektors. Die Indizes von Station und Sektor müssen dann mit Hilfe einer fest codierten Tabelle in solche für Readout Unit und Sektor umgewandelt werden, die zum Berechnen der Position im Speicher erforderlich sind. Die in Gaudi oder allgemein in C++ üblichen Zeiger oder Referenzen auf Objekte sind an dieser Stelle als Verweis auf Cluster ungeeignet, weil dies für jedes Ereignis die Erzeugung von Cluster-Objekten erfordern würde, was zusätzliche Rechenzeit in Anspruch nimmt.

5.4.2 Die Struktur des Algorithmus

Die gesamte Trigger-Code verteilt sich auf Algorithmus- und Datenobjekte. Algorithmen enthalten die Anweisungen zum Auswerten der Ereignisdaten und werden einmal pro Ereignis ausgeführt. Datenobjekte dienen zum Datenaustausch zwischen den Algorithmen, die bestimmte Datentypen einlesen und neue produzieren. Das Framework stellt für beide Objekttypen Basisklassen zu Verfügung, die die entsprechenden Grundfunktionen zur Verfügung stellen. Beim Start eines Jobs kommen nur diejenigen Algorithmen zur Ausführung, welche in den Job Options als *Top Level* spezifiziert wurden. Hier wird auch die Reihenfolge bestimmt, in der sie vom Application Manager für jedes Ereignis aufgerufen werden. Außerdem besteht die Möglichkeit, Algorithmen direkt aus anderen heraus zu starten und so deren Ausführung von bestimmten Bedingungen abhängig zu machen.

Abbildung 5.3 zeigt sämtliche Algorithmen und Datentypen, die im Rahmen dieser Arbeit entstanden sind, sowie die Pfade der Datenübergabe. Die Klassen sind unterteilt in solche, die ausschließlich mit den reinen Ereignisdaten rechnen bzw. den reinen Trigger-Code darstellen, wie er prinzipiell auch in der Experimentalumgebung laufen könnte, und solche, die zusätzlich Informationen der Monte-Carlo-Simulation auswerten und lediglich zur Analyse des Ereignises und der Ergebnisse dienen. Dies entspricht dem in Kapitel 3 angesprochenen Konzept der einheitlichen Umgebung für den Algorithmus in der Simulation und im Experiment.

5.4.2.1 Daten-Objekte

Alle Daten sind Objekte einer von `DataObject` oder `ContainedObject` abgeleiteten Klasse, je nachdem, ob einzelne Objekte zwischen den Algorithmen übergeben werden sollen oder Listen bzw. Vektoren [26, Kap.6]. Klassen mit dem Präfix RU im Namen enthalten keine Bezüge zu den Monte-Carlo-Daten und können deshalb prinzipiell auch ohne Simulationsumgebung eingesetzt

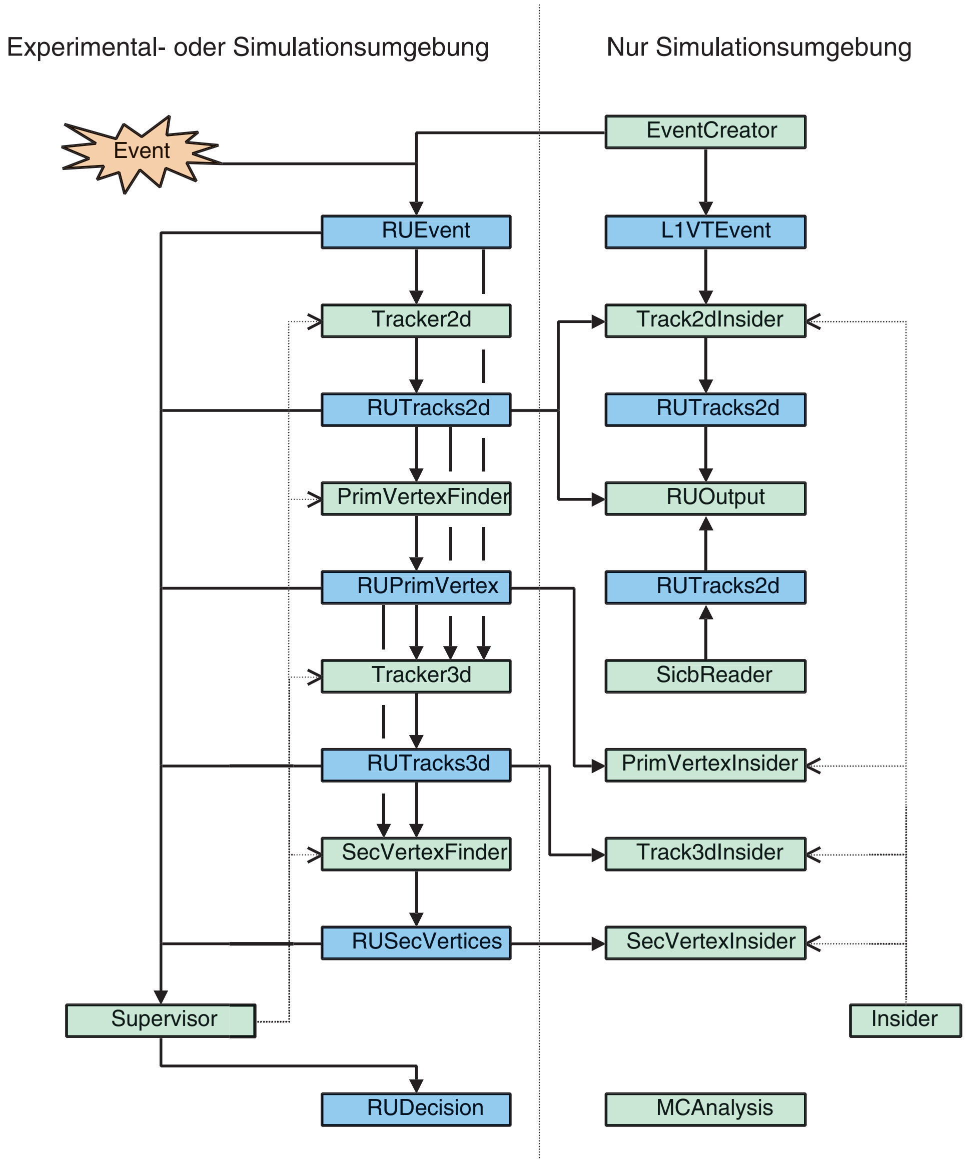


Abbildung 5.3: Sämtliche Algorithmus- (grün) und Datenobjekte (blau) dieser Implementierung. Durchgezogene Pfeile kennzeichnen die Datenpfade, während gestrichelte die Kontrolle von Unteralgorithmen durch Algorithmen der obersten Ebene beschreiben.

werden.[3] Die folgende Aufzählung beschreibt den Inhalt sämtlicher Daten-Objekte:

`RUEvent` — Alle Informationen über die r- und ϕ-Cluster. Objekte dieser Klasse enthalten nicht die Daten selbst, sondern lediglich einen Zeiger auf den Bereich im Speicher, an dem die Informationen abgelegt wurden und deren Format in Abschnitt 5.4.1 beschrieben wurde. Durch diese Auslagerung bleibt die Klasse auch in der Experimentalumgebung einsetzbar, wenn die Daten von der Netzwerk-Hardware in den Speicher geschrieben werden.

`RUTracks2d` — Vektor aus `RUTrack2d`. Diese Klasse enthält die Parameter für 2D-Spuren sowie die Indizes der r-Cluster, aus denen die Spur gebildet wurde.

`RUPrimVertex` — Position des primären Vertex (x-, y- und z-Koordinate).

`RUTracks3d` — Vektor aus `RUTrack3d`. Objekte diesen Typs speichern die Parameter für 3D-Spuren (Start- und Richtungsvektor, siehe Abschnitt 4.2.3) sowie die Indizes der zugrunde liegenden 2D-Spur und der ϕ-Cluster.

`RUSecVertices` — Vektor aus `RUSecVertex`. Mit Objekten dieser Klasse wird die Position eines Sekundärvertices (x-, y- und z-Koordinate) festgehalten, sowie die Indizes zweier 3D-Spuren, aus denen der Vertex berechnet wurde.

`RUDecision` — Enthält die Level-1 Entscheidung.

`L1VTEvent` — Speichert für alle r- und ϕ-Cluster Referenzen auf die zusätzlichen Monte-Carlo-Informationen, die in der Simulation bereitstehen. Ein Objekt dieser Klasse wird parallel zu `RUEvent` erstellt und ermöglicht die Analyse der berechneten Spuren und Vertices.

5.4.2.2 Algorithmus-Objekte

Algorithmen sind Objekte einer von `Algorithm` abgeleiteten Klasse [26, Kap.5]. Die Verwirklichung der in Kapitel 4.2 beschrieben Abschnitte des Level-1 Codes verteilt sich auf folgende Gaudi-Algorithmen:

`Tracker2d` — Erstellt aus den r-Koordinaten des Ereignisses 2D-Spuren (Abschnitt 4.2.1).

[3] RU steht für Readout Unit und soll den Bezug zum Auslesesystem andeuten.

`PrimVertexFinder` — Versucht die Position des primären Vertex aus 2D-Spuren zu berechnen (Abschnitt 4.2.2).

`Tracker3d` — Erzeugt aus 2D-Spuren unter Verwendung der ϕ-Cluster 3D-Spuren, wobei mit Hilfe der Position des Primärvertex eine Selektion bei den 2D-Spuren vorgenommen wird (Abschnitt 4.2.3).

`SecVertexFinder` — Sucht unter den 3D-Spuren unter Berücksichtigung der Position des Primärvertex nach Sekundärvertices. (Abschnitt 4.2.4).

`Supervisor` Steuert den Ablauf der oben aufgeführten Algorithmen und trifft die Trigger-Entscheidung. Kann bereits vor Ablauf der gesamten Kette eine positive oder sinnvolle Entscheidung ausgeschlossen werden, wird auf die weitere Berechnung von Spuren und Vertices verzichtet. Ist dies nicht der Fall, wird am Ende aus den gefundenen Sekundärvertices eine Empfehlung für die Entscheidungseinheit des gesamten Trigger-Systems berechnet.

Alle übrigen Algorithmen können nur unter Einsatz simulierter Daten zur Ausführung kommen, da die physikalischen Hintergrundinformationen zu den Meßwerten benötigt werden. Diese werden durch das Framework als spezielle Datenobjekte bereit gestellt.

`EventCreator` — Liest die simulierten r- und ϕ-Cluster ein, transformiert diese in das beschriebene Datenformat der Hardware und erstellt das zugehörige `RUEvent` Objekt. Außerdem wird ein `L1VTEvent` Objekt erstellt, in dem die Verknüpfungen zu den simulierten Teilchen festgehalten werden, und das Voraussetzung für die im folgenden beschriebenen Analyse-Routinen ist.

`Track2dInsider` — Untersucht einen `RUTracks2d` Vektor auf falsche und fehlende Spuren, aus denen jeweils ein neuer Vektor erstellt wird, der dann durch folgende Algorithmen im Detail untersucht oder visualisiert werden kann. Als falsche Spuren werden solche bezeichnet, deren erster und dritter Triplett-Cluster *nicht* vom selben Teilchen erzeugt wurde — wenn auch nur zum Teil.[4] Da die mittlere Koordinate des Tripletts nicht zur Berechnung der Spurparameter herangezogen wird, braucht diese bei der Überprüfung auch nicht berücksichtigt zu werden. Die fehlenden Spuren gehören zu den Monte-Carlo-Teilchen, deren Spuren mindestens drei aufeinanderfolgende Stationen getroffen haben und deshalb vom Trigger-Algorithmus gefunden werden sollten, die jedoch nicht in den übergebenen Spuren enthalten sind.

[4] Die Ursache eines Clusters kann in mehreren Teilchen liegen, wenn deren Spuren im Detektor nicht weit genug voneinander entfernt sind und deshalb dieselben Streifen ansprechen.

`PrimVertexInsider` — Vergleicht die Koordinaten eines `RUPrimVertex` Objekts mit denen des primären Vertex aus der Monte-Carlo-Simulation, um den Fehler der Positionsbestimmung zu ermitteln.

`Track3dInsider` — Sucht nach falschen Spuren in einem `RUTracks3d` Vektor. Dabei wird dasselbe Verfahren wie für die 2D-Spuren angewendet, nur daß zusätzlich die ϕ-Cluster in die Überprüfung miteinbezogen werden. Nach fehlenden Spuren an dieser Stelle zu suchen, wäre gemessen an den daraus resultierenden Erkenntnissen zu aufwendig, da sämtliche Filtermechanismen bei der Selektion von 2D-Spuren auch in der Analyse implementiert werden müßten.

`SecVertexInsider` — Überprüft den Inhalt eines `RUSecVertices` Vektors. Dabei wird zunächst kontrolliert, ob keine falsche 3D-Spur verwendet wurde, um den Vertex zu bilden; außerdem, ob beide Spuren ihren Ursprung im selben Zerfall haben und es sich bei diesem *nicht* um den Primärvertex handelt; und schließlich, ob es der Zerfallsvertex eines B-Mesons ist.

`Insider` — Startet die oben beschriebenen Analyse-Algorithmen der Reihe nach, so daß diese nicht alle einzeln als Algorithmen der obersten Ebene in den Job Options angegeben werden müssen. Außerdem werden die Zwischenresultate auf den Verbleib der Zerfallsvertices von B-Mesonen hin untersucht.

Die übrigen Algorithmen dienen allein der Analyse der simulierten Ereignisdaten und laufen deshalb unabhängig von den Ergebnissen der Suche nach Spuren und Vertices:

`SicbReader` — Ermöglicht das Anzeigen einiger der mit SICb erstellten ZEBRA-Banken, nämlich der Monte-Carlo-Spuren und Vertices, der r- und ϕ-Cluster sowie sämtliche Ergebnisse des in Fortran geschriebenen Level-1 Algorithmus.

`RUOutput` — Kann die r-Koordinaten aus einem `RUEvent`-Objekt und die Spurparameter eines `RUTracks2d`-Vektors als N-Tuple im HBOOK[5]-Format in eine Datei exportieren. Auf diese Weise können die Cluster und Spuren beispielsweise mit Programmen wie ROOT veranschaulicht werden (Abb. 4.7 zeigt zwei Bilder, die auf diese Art und Weise erzeugt wurden), was mit der aktuellen Version 4 von Gaudi noch nicht möglich (aber in Planung) ist. Als Quelle können die Spuren des Trigger-Codes genauso, wie die in der Analyse ermittelten falschen und fehlenden Spuren sowie die in SICb berechneten, eingesetzt werden.

[5]Datenformat, das von der Analyse-Software PAW verwendet wird.

`MCAnalysis` Führt verschiedene Analysen der Monte-Carlo-Daten durch, die ein wichtiges Werkzeug zur Abstimmung der Parameter des Trigger-Codes darstellen.

5.4.3 Parameter

Der Trigger-Algorithmus ist durch eine Reihe von Parametern konfigurierbar, die als sogenannte *Job Options* in einer Text-Datei zusammengefaßt sind. Diese Werte dienen dazu, Qualität und Geschwindigkeit zu optimieren und durch Schwellen eine sinnvolle Trigger-Entscheidung anhand der berechneten Daten zu treffen. Um geeignete Werte zu finden, helfen spezielle Analysen der Monte-Carlo-Daten, die ebenfalls Teil dieser Arbeit sind, und deren Ergebnisse in Abschnitt 6.1 vorgestellt werden. Dort befindet sich auch eine Auflistung der konkreten Werte (Tab. 6.1), während an dieser Stelle die Parameter nur qualitativ besprochen werden.

Die Suche nach r-Tripletts und somit nach 2D-Spuren kann durch den Akzeptanzfaktor beeinflußt werden. Es handelt sich dabei um den Faktor n aus Gleichung 4.1, der zusammen mit den Ortsauflösungen die Linearitätsbedingung für drei r-Koordinaten definiert. Durch die Vergrößerung dieses Wertes kann die Zahl nicht gefundener Spuren reduziert werden. Es handelt sich dabei hauptsächlich um die Spuren impulsarmer Teilchen, die durch Wechselwirkung mit dem Detektormaterial stärkere Richtungsänderungen erfahren und deshalb größere Abweichungen von einer geraden Bahn aufweisen. Der Gewinn an Spuren läuft allerdings mit einer höheren Rate an falschen Spuren einher, da natürlich auch die Wahrscheinlichkeit für r-Cluster verschiedener Teilchen steigt, auf einer Gerade innerhalb der berechneten Fehlergrenzen zu liegen. Weiterhin kann ein Bereich für die Schnittpunkte der Spuren mit der z-Achse definiert werden. Dieser ergibt sich aus Überlegungen über die möglichen Positionen des primären Wechselwirkungspunktes, den maximal zu erwarteten Zerfallslängen der B-Teilchen sowie dem Winkel der Flugbahnen zur Strahlachse. Auf diese Weise können Spuren eliminiert werden, die ihren Ursprung weder im primären, noch in einem sekundären Vertex haben können und in der Regel ohnehin falsch sind.

Für die Bestimmung der Position des primären Vertex sind die Histogrammparameter von entscheidender Bedeutung für die Geschwindigkeit und sind deshalb ebenfalls variierbar. So können die Grenzen entsprechend dem erwarteten Bereich im Raum für die Position des Wechselwirkungspunktes vorgegeben und durch die Bin-Breite ein Kompromiß zwischen Rechenaufwand und Auflösung gewählt werden. Außerdem gibt es eine minimal erforderliche Zahl von Histogrammeinträgen, bei deren Unterschreiten die Suche nach dem Häufungspunkt nicht gestartet wird, weil mit keinem sinnvollem

Ergebnis zu rechnen ist.

Bei der Suche nach ϕ-Tripletts zur Bildung von 3D-Spuren kann genau wie für die r-Koordinaten die erlaubte Abweichung von geraden Spuren durch den Faktor n aus Gleichung 4.5 gesteuert werden. Auch hier gilt es, einen sinnvollen Kompromiß zwischen den Raten falscher bzw. fehlender Spuren einzustellen. Ein entscheidender Geschwindigkeitsaspekt bei der 3D-Spurensuche ist die Selektion der 2D-Spuren. Da die vollständige Rauminformation nur für Tochterteilchen von B-Mesonen benötigt wird, kann durch die Vorgabe eines Bereichs für den Impact Parameter ein Großteil der 2D-Spuren ausgelassen werden. Je enger die Grenzen gesetzt werden, desto weniger Spuren müssen ausgewertet werden. Allerdings steigt auch die Menge an Spuren aus B-Zerfällen, die verloren gehen und somit für die Bestimmung der zugehörigen Sekundärvertices nicht berücksichtigt werden können.

Für 3D-Spuren, die aus demselben Vertex kommen, wird ein Maximum für den geringsten Abstand der aus den Tripletts rekonstruierten Geraden festgesetzt. Hierdurch können die Raten falscher und nicht gefundener Vertices gegeneinander verschoben werden. Außerdem muß ein ebenfalls einstellbarer räumlicher Abstand zum Primärvertex überschritten werden, bevor ein Schnittpunkt als Sekundärvertex akzeptiert wird.

Schließlich gibt es noch eine Reihe von Parametern, die als Schwellen für eine positive Trigger-Entscheidung verwendet werden. Solche existieren für die Anzahl von r-Clustern, 2D-Spuren und sekundären Vertices. Besonders durch die geforderte Zahl von Clustern können viele Ereignisse ohne B-Zerfälle bereits vor dem Start der Unteralgorithmen erkannt werden, was die Entscheidungszeit auf ein Minimum reduziert. Bei einer zu geringen Menge an 2D-Spuren ist die Positionsbestimmung des Primärvertex nicht sinnvoll, da dies statistisch durch Histogrammauswertung geschieht. Die entscheidende Variable, um das Verhältnis zwischen akzeptierten und abgelehnten Ereignissen des Level 1 Triggers einzustellen, ist letztendlich die Menge der geforderten Sekundärvertices, die als Hinweis auf B-Zerfälle interpretiert wird.

Kapitel 6

Ergebnisse

Die Auswertung der vorgestellten Realisierung des Level-1 Vertex-Trigger Algorithmus findet unter zwei Gesichtspunkten statt. Einerseits gilt es zu prüfen, ob die berechneten Empfehlungen, Ereignisse zu akzeptieren oder zu verwerfen, gründlich genug durchgeführt wurden in Hinblick auf die zugrunde liegende Physik der Ereignisse. Andererseits liegt eine der Hauptanforderungen an den L1 Trigger in einer möglichst hohen Arbeitsgeschwindigkeit, um so die erforderliche Latenz-Zeit des L1 Triggers und den Umfang der zur Berechnung verwendeten Prozessor-Farm und damit die Kosten möglichst gering zu halten. Ein weiterer Teil der Ergebnisse dient als Hilfe, um eine effektive Einstellung der variablen Parameter des Algorithmus zu finden. Es handelt sich dabei um Analysen, die unabhängig von den Resultaten der Spuren- und Vertexsuche die Ereignisdaten der Simulation untersuchen.

Sämtliche in diesem Kapitel vorgestellten Ergebnisse basieren auf Simulationsdaten, die mit SICb v233 erstellt wurden [30]. Dabei wurden zwei unterschiedliche Datensätze mit jeweils 500 Ereignissen ausgewertet, die jeweils eine besonders niedrige bzw. hohe Akzeptanz-Rate erzeugen sollten.[1] Der eine enthält Minimum Bias Ereignisse[2], die möglichst vollständig abgelehnt werden sollten, da sie in fast allen Fällen keine Analyse von B-Physik ermöglichen. Der zweite Datensatz beinhaltet ausschließlich Primärreaktionen, die mindestens einen Zerfall vom Typ $\mathrm{B_d^0} \to \pi^+\pi^-$ erzeugen. Diese Zerfallsart stellt wegen der geringen Zahl an Endzuständen höchste Anforderungen an den Trigger-Algorithmus, da zur Rekonstruktion des Zerfallsvertex lediglich zwei Teilchenspuren zur Verfügung stehen. Daher eignet sich dieser Ereignistyp besonders gut dazu, die Leistungsfähigkeit des Codes bei der Suche nach solchen Vertices zu testen bzw. zu demonstrieren. Beide Datensätze sind frei von Ereignissen mit mehr als einer primären Wechselwirkung.

[1] Es handelt sich dabei um die SICb-Jobs mit den Nummern 15792 und 15836

[2] Als solche werden ungefilterte Ereignisse beliebigen Typs bezeichnet.

Außerdem fehlt die Vorselektion durch den Level-0 Trigger, weswegen die Auswertungen der Minimum Bias Ereignisse — besonders die Geschwindigkeitsmessungen — nur bedingt aussagekräftig sind, da gerade von ihnen nur ein kleiner Bruchteil Level-0 passieren würde. Besonders die kleineren Ereignisse mit wenigen Spuren, die in den Messungen der Berechnungsdauern von Minimum Bias Ereignissen die Statistik verbessern, würden den Level-1 Trigger nicht mehr erreichen.

Die für sämtliche Resultate in diesem Kapitel verwendeten Werte der entscheidenden Algorithmus-Parameter sind in Tabelle 6.1 zusammengestellt. Deren Bedeutung wurde in Abschnitt 5.4.3 besprochen. Diese Einstellung stellt einen Kompromiß dar, der bei einer möglichst hohen Erkennungsrate von B-Zerfällen die Akzeptanz von Minimum Bias Ereignissen minimal halten soll. Außerdem wurde versucht, einen möglichst geringen Rechenaufwand zu erzielen, ohne Akzeptanzraten entscheidend zu verschlechtern. Alle hier vorgestellten Zahlen basieren auf den Jobs, deren Ausgaben auf der beiliegenden CD abgelegt sind (Anhang A). Diese Text-Dateien enthalten auch die vollständige Konfiguration.

6.1 Analyse der Ereignisdaten

Die in diesem Abschnitt vorgestellten Histogramme wurden unabhängig von den Ergebnissen des Trigger-Codes erstellt. Sie dienten zunächst als Hilfe, um einen sinnvollen Wert verschiedener Parameter abschätzen zu können. Jetzt veranschaulichen sie die Wirkung einiger der letztendlich gewählten Bereiche und Grenzwerte aus Tabelle 6.1.

Abbildung 6.1 zeigt die Verteilung der Größe von Ereignissen mit B-Zerfall, wobei als Größe die Anzahl der erzeugten Cluster verstanden wird. Diese beträgt im Mittel ungefähr 740 und kann bis zu 2000 erreichen. Da r- und ϕ-Cluster etwa im Verhältnis 1:1 auftreten, läßt sich durch Betrachtung des Bereichs geringer Cluster-Multiplizität der für die Akzeptanz geforderte Schwellwert von 50 radialen Meßpunkten verstehen, da hierdurch keine B-Zerfälle verloren gehen.

Die örtliche Verteilung entlang der Strahlachse primärer Wechselwirkungen, aus denen B-Mesonen entstehen, zeigt Abbildung 6.2, in der die z-Koordinaten des Primärvertex dargestellt sind. Hieraus kann man den benötigten Bereich für das Histogramm abgelesen, in das die Schnittpunkte der 2D-Spuren zur Rekonstruktion eben dieser Koordinate durch den Algorithmus eingetragen werden. Von dieser Verteilung hängt auch die erstmögliche Station ab, die durch Vorwärts- bzw. Rückwärts-Spuren getroffen werden kann, was in Abbildung 6.3 konkret analysiert wurde.

2D-Spuren	
Erlaubter Bereich für den z-Achsenabschnitt	[-30 cm ; +30 cm]
Akzeptanzfaktor n (Gl. 4.1)	3
Index der erstmöglichen Station für Cluster von Vorwärtsspuren	4
Index der erstmöglichen Station für Cluster von Rückwärtsspuren	5
Minimal erforderliche Anzahl	50

Primärvertex	
Bin-Bereich z-Koordinate	[-100 mm ; +100 mm]
Bin-Breite z-Kooridnate	0,4 mm
Minimal erforderliche Anzahl von z-Einträgen	100
Bin-Bereich x- und y-Koordinate	[-1,5 mm ; +1,5 mm]
Bin-Breite x- und y-Kooridnate	0,4 mm

3D-Spuren	
Bereich für den Impact Parameter	[+0,25 mm ; +4,0 mm]
Akzeptanzfaktor n (Gl. 4.5)	1
Minimal erforderliche Anzahl	2

Sekundärvertices	
Maximale Entfernung zweier Spuren aus demselben Vertex	0,2 mm
Geforderte Entfernung vom Primärvertex	0,6 mm
Minimale Anzahl für positive Trigger-Entscheidung	3

Tabelle 6.1: Einstellungen der Algorithmus-Parameter, mit denen die vorgestellten Ergebnisse erzeugt wurden.

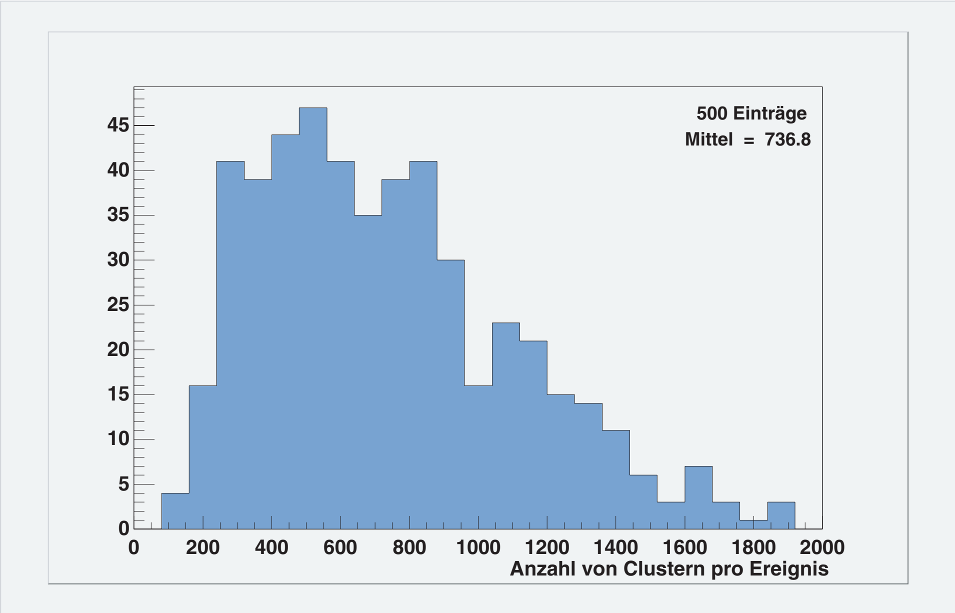


Abbildung 6.1: Anzahl aller Cluster (r und ϕ) pro Ereignis mit B-Zerfall.

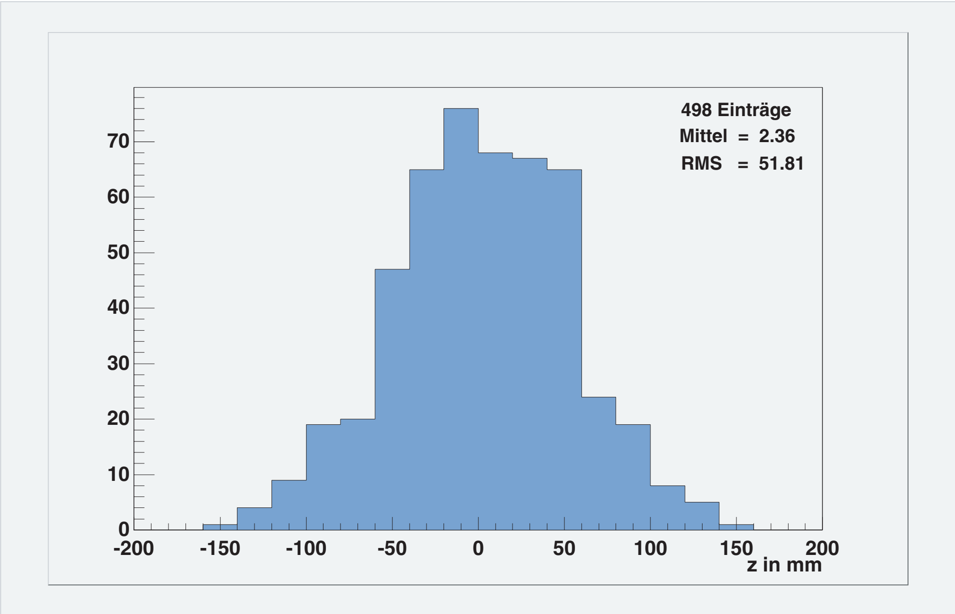


Abbildung 6.2: z-Koordinate des primären Vertex von Ereignissen mit B-Zerfall.

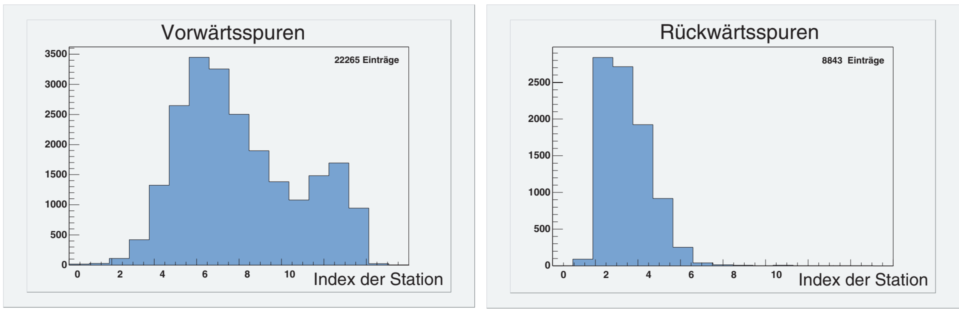


Abbildung 6.3: Index der ersten Station, die durch Spuren von B-Ereignissen getroffen wird.

Mit Hilfe rekonstruierter 2D-Spuren wurde die Grafik in Abbildung 6.4 erstellt. Dazu wurden für alle Teilchen sämtlicher B-Ereignisse mit Meßpunkten auf mindestens drei aufeinanderfolgenden Stationen die Spurparameter gemäß den im Algorithmus verwendeten Gleichungen 4.2 und 4.3 berechnet und der daraus resultierende Schnittpunkt mit der z-Achse untersucht. Man erkennt, daß durch die Einschränkung erlaubter Werte auf den Bereich zwischen -30 und $+30$ cm kaum Spuren verloren gehen, die zur Rekonstruktion von B-Zerfallsvertices benötigt werden und im Diagramm dunkel markiert sind. Allerdings werden eine Vielzahl falscher Spuren eliminiert, die durch zufällig auf einer Gerade liegenden r-Cluster verschiedener Teilchen verursacht wurden und die Strahlachse meistens weit jenseits des genannten Bereichs schneiden.

Mit denselben Spurparametern entstand auch das Diagramm in Abbildung 6.5, das die Impact Parameter in Bezug zum Primärvertex enthält (Gl. 4.4). Auch hier sind die zur Rekonstruktion von Sekundärvertices relevanten Spuren hervorgehoben, die durch geeignete Schnitte angereichert werden sollen. Man erkennt, daß sich die beiden Häufungspunkte auf diese Weise leider nicht voneinander trennen lassen und man durch Eliminierung des Hauptteils der Spuren aus dem Primärvertex auch eine erhebliche Anzahl an Spuren aus B-Vertices verliert. Dies begründet sich hauptsächlich in der Natur des Teilchenzerfalls, dessen exponentielle Verteilung für die B-Mesonen in Abbildung 6.6 dargestellt ist und zeigt, daß trotz einer mittleren Zerfallslänge von etwa 7 mm viele Sekundärvertices in unmittelbarer Umgebung zum Primärvertex liegen.

Hilfreich für die Kalibrierung der Suche nach Sekundärvertices ist die Statistik in Abbildung 6.7. Sie begründet die maximal erlaubte Entfernung zweier rekonstruierter 3D-Spuren aus demselben Vertex, die auf 0,2 mm festgesetzt wurde.

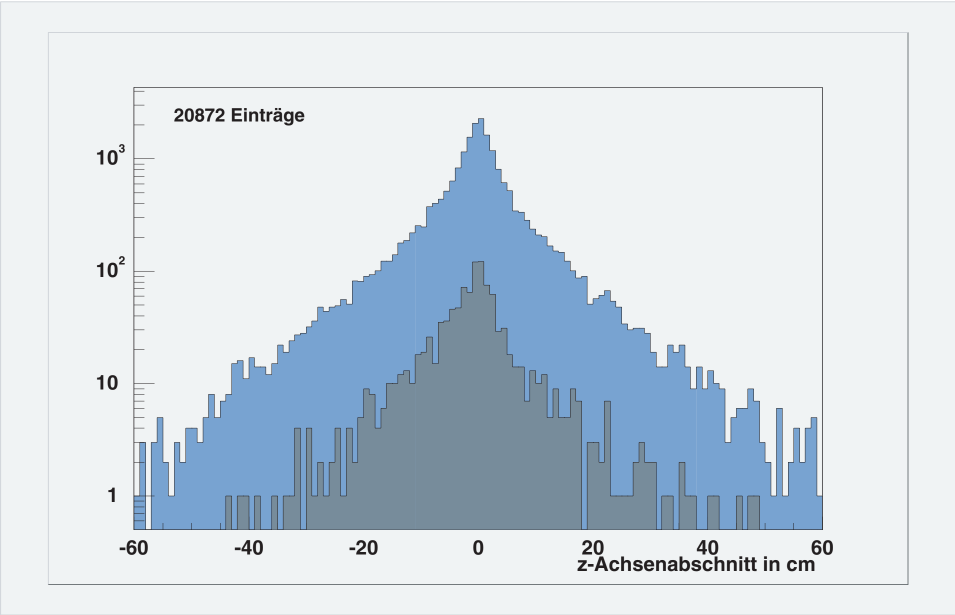


Abbildung 6.4: z-Achsenabschnitt rekonstruierter 2D-Spuren. Der Anteil, der aus Zerfallsvertices von B-Mesonen stammt, ist dunkel markiert.

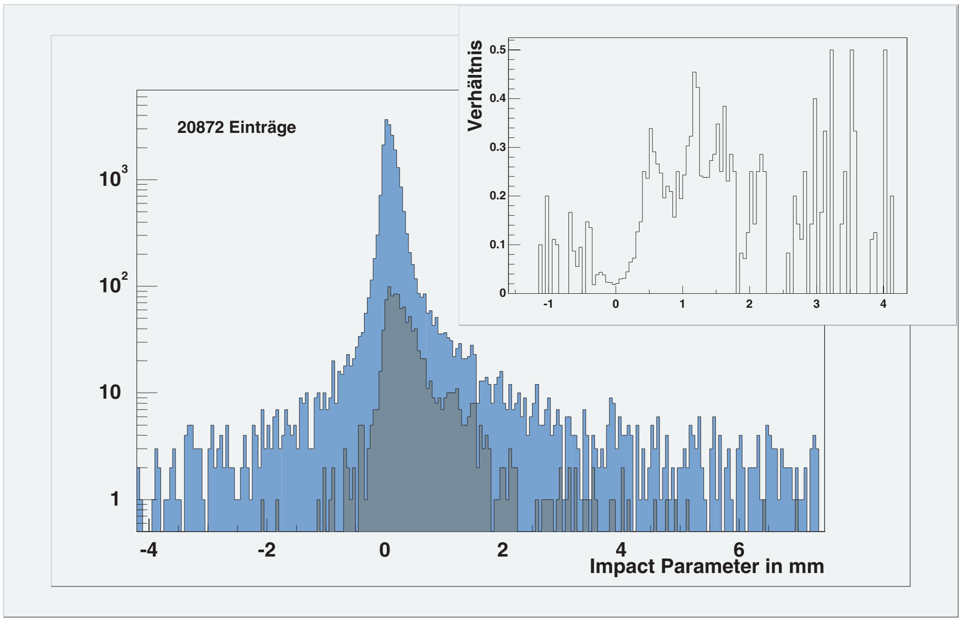


Abbildung 6.5: Impact Parameter rekonstruierter 2D-Spuren aus allen B-Ereignissen. Der Anteil, der aus Zerfallsvertices von B-Mesonen stammt, ist dunkel markiert. Außerdem ist das Verhältnis dieser Spuren zur Gesamtzahl dargestellt.

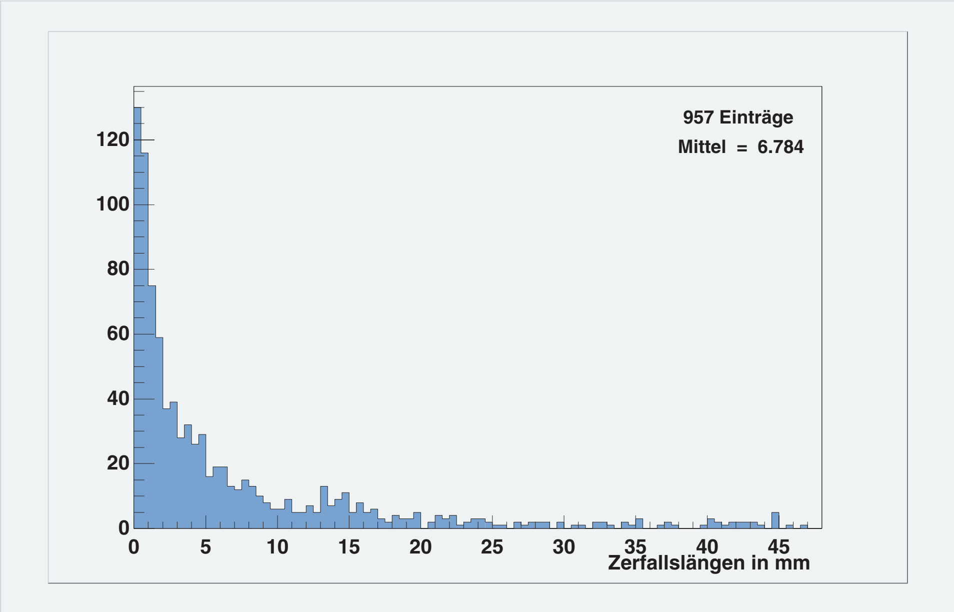


Abbildung 6.6: Zerfallslängen der B-Mesonen.

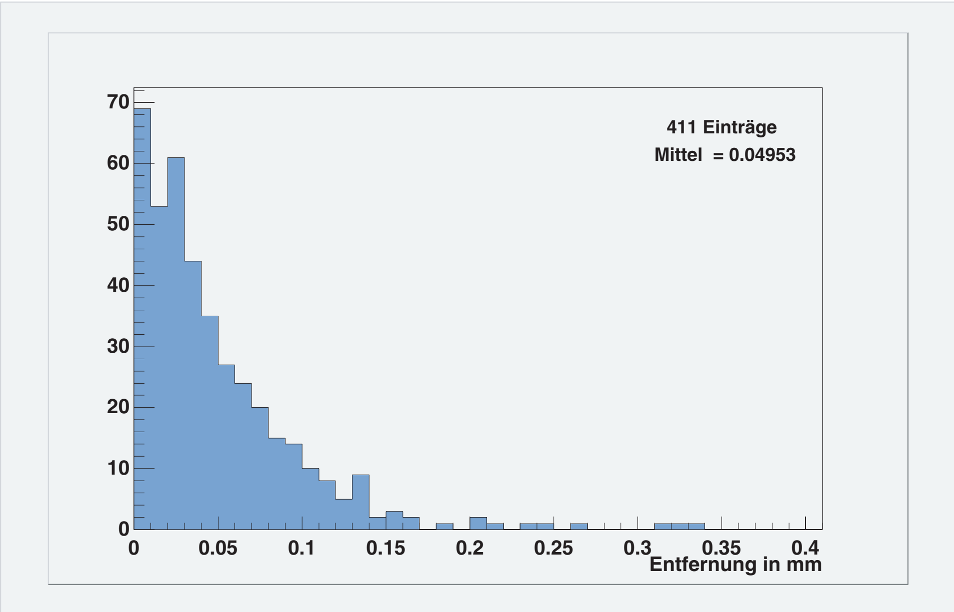


Abbildung 6.7: Minimaler Abstand zwischen rekonstruierten 3D-Spuren, die durch den Zerfall deselben B-Teilchens entstehen.

6.2 Analyse der Resultate

Der Level-1 Trigger sollte im Idealfall nur Ereignisse akzeptieren, in denen zur Auswertung geeignete Zerfälle von B-Mesonen auftreten. Der Trigger-Algorithmus trifft diese Entscheidung anhand der Zahl gefundener Sekundärvertices. Um die Güte dieser Selektion verbessern zu können, ist die Suche nach Fehlerquellen und deren Ursache innerhalb der gesamten Prozedur notwendig.

Die Qualität der Suche nach 2D-Spuren wird durch zwei Werte charakterisiert. Zunächst durch die Effizienz, d.h. den Anteil aller Teilchen mit mindestens 3 Meßpunkten im Vertex-Detektor, deren Spuren vom Algorithmus gefunden wurde. Außerdem durch den Anteil berechneter Spuren, die aus Meßwerten unterschiedlicher Teilchen konstruiert wurden und im folgenden als falsche Spuren bezeichnet werden. Als Durchschnittswerte beider Größen ergeben sich:

2D-Spuren	$\mathrm{B_d^0} \to \pi^+\pi^-$	Minimum Bias
Effizienz	82,2%	83,1%
Anteil falscher Spuren	8,7%	5,2%

In Abb. 6.8 ist die Effizienz in Abhängigkeit vom Teilchenimpuls dargestellt. Dabei ist deutlich zu erkennen, daß dieser Wert für impulsarme Teilchen stark abfällt, weil die Wechselwirkung mit dem Detektormaterial stärkere Richtungsänderungen der Flugbahn verursacht und somit zu Abweichungen von einer geraden Flugbahn führt.

Die ermittelten Raumkoordinaten des Primärvertex werden durch einen Vergleich mit der von der Simulation erzeugten Position analysiert. Dadurch ergeben sich folgende Auflösungen (Abb. 6.9 zeigt die zugrunde liegenden Histogramme für $\mathrm{B_d^0} \to \pi^+\pi^-$ Ereignisse):

Primärvertex	$\mathrm{B_d^0} \to \pi^+\pi^-$	Minimum Bias
σ_x	45 μm	42 μm
σ_y	45 μm	35 μm
σ_z	90 μm	110 μm

Die Werte für Minimum Bias — im besonderen für x und y — sind weniger aussagekräftig, weil die Position nur für etwa 25% der Fälle bestimmt werden kann. Dies liegt an der extrem niedrigen Spurmultiplizität vieler dieser Ereignisse, was auch die schlechtere Auflösung von z erklärt.

Von den 2D-Spuren passieren durchschnittlich nur 11,3% (B-Ereignisse) die Selektion durch den Impact Parameter. Pro 2D-Spur bzw. r-Triplett

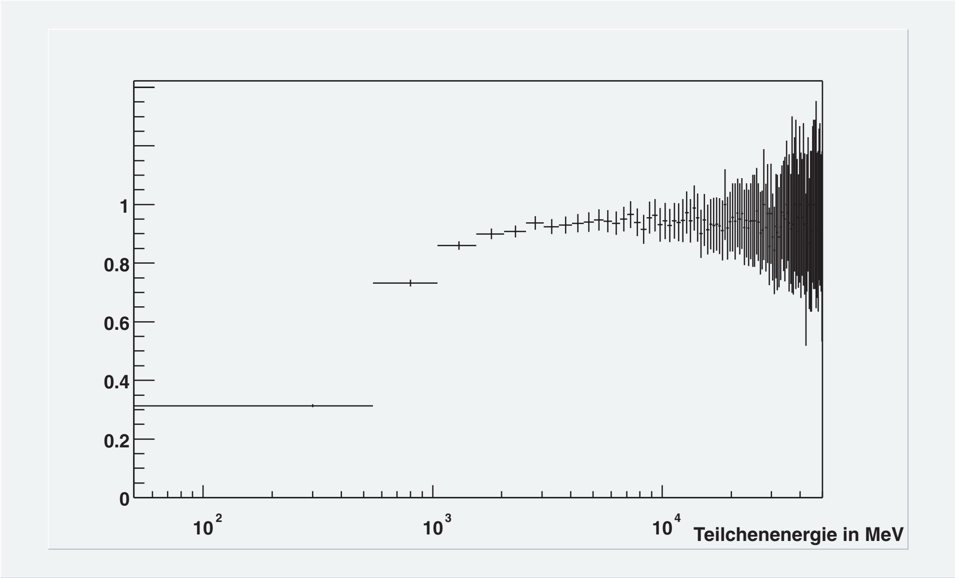


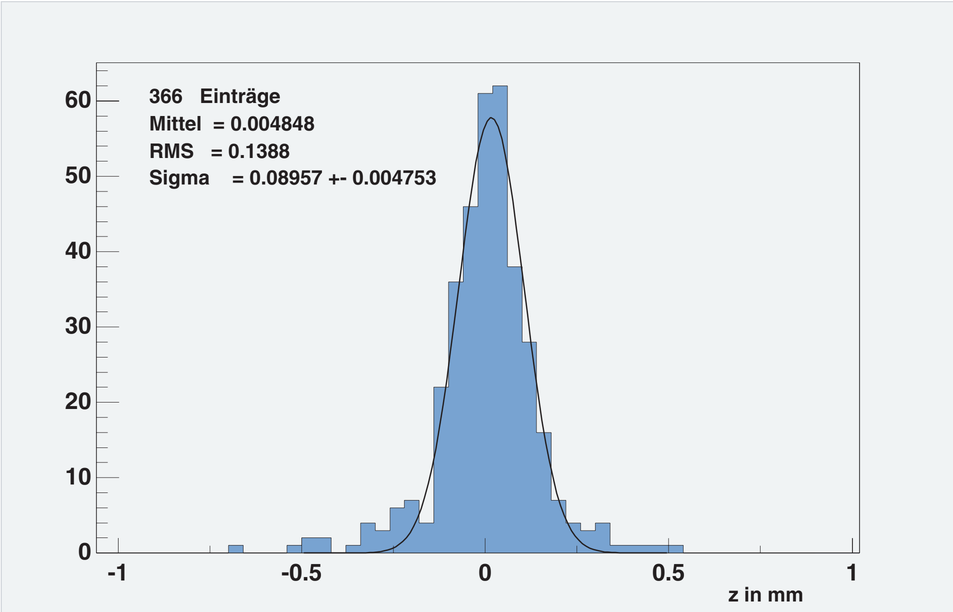
Abbildung 6.8: Effizienz der Suche nach 2D-Spuren in Abhängigkeit vom Teilchenimpuls. Die Referenzmenge bilden alle Spuren mit mindestens 3 aufeinanderfolgenden Treffern in den r-Detektoren.

werden im Schnitt 4,0 passende Kombinationen aus drei ϕ-Clustern gefunden. Für 95,5% findet der Algorithmus mindestens ein passendes ϕ-Triplett und somit eine 3D-Spur. Davon sind 19,3% falsche Spuren, weil entweder die zugrunde liegende 2D-Spur bereits nicht korrekt war, oder die ϕ-Cluster nicht zum selben Teilchen bzw. zu einem anderen als dem der r-Cluster gehören. Für Minimum Bias ergibt sich ein ähnlicher Wert:

3D-Spuren	$\mathrm{B_d^0} \to \pi^+\pi^-$	Minimum Bias
Anteil falscher Spuren	19,3%	15,5%

Im Schnitt werden 4,8 Sekundärvertices pro $\mathrm{B_d^0} \to \pi^+\pi^-$ Ereignis gefunden. Bei dieser Zahl ist zu beachten, daß der Zerfall des korrespondierenden B-Zerfalls häufig in mehr als zwei geladene Teilchen erfolgt und daher zusätzlich mehr als einen Sekundärvertex erzeugen kann (siehe dazu Abschnitt 4.2.4). Dennoch ist ein großer Anteil falscher Vertices enthalten, worauf noch näher eingegangen wird. Die L1-Akzeptanz, was man als Endresultat aller Bemühungen ansehen könnte, fällt wie folgt aus:

Level-1	$\mathrm{B_d^0} \to \pi^+\pi^-$	Minimum Bias
Akzeptierte Ereignisse	165 (33,0%)	26 (5,2%)



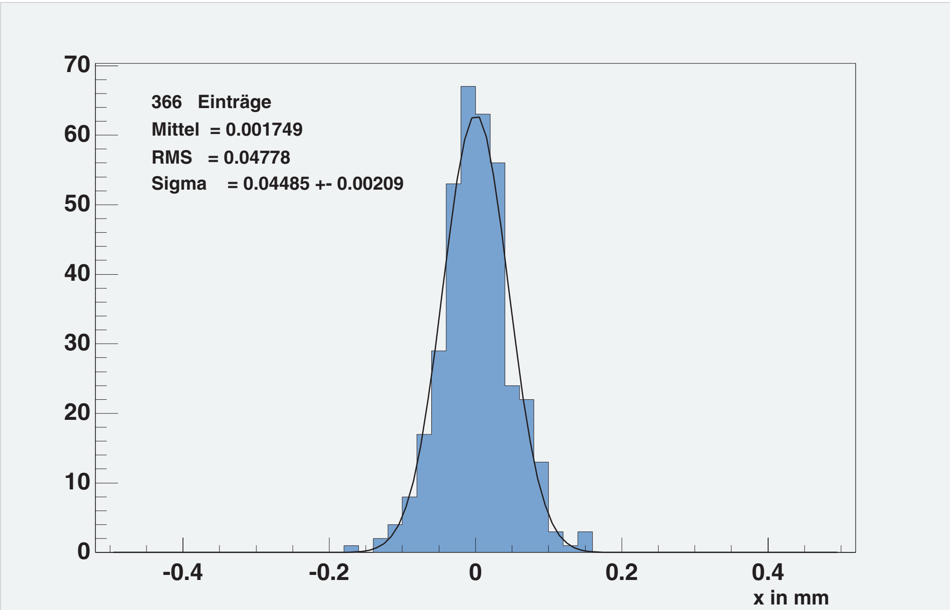


Abbildung 6.9: Fehler der rekonstruierten z- und x-Koordinaten des Primärvertex aller $\mathrm{B}_{\mathrm{d}}^{0} \to \pi^{+}\pi^{-}$ Ereignisse mit angepaßter Gauß-Funktion. Das entsprechende Diagramm für die y-Koordinaten ist qualitativ identisch zu x.

Im Idealfall sollte der Algorithmus für jeden Zerfallvertex eines B-Mesons mindestens einen Sekundärvertex erzeugen. Daß die Realität noch anders aussieht, verdeutlicht die Statistik in Abbildung 6.10. Sie ist Ergebnis einer Analyse, die den Verlust von B-Vertices im Verlauf der Spuren- und Vertexsuche untersucht. In der Anfangsmenge sind dabei sämtliche B-Zerfälle, also auch solche, die aus meßtechnischen Gründen nicht vom Algorithmus gefunden werden können. Im folgenden sind die Ursachen für die einzelnen Balken aus dem Diagramm beschrieben:

Anzahl der r-Cluster — Das Ereignis hat so wenige Meßwerte im Vertex-Detektor erzeugt, daß die verlangte Mindestzahl von r-Clustern nicht erreicht und die Berechnung deswegen abgebrochen wurde. Solche Ereignisse werden sehr wahrscheinlich schon von Level-0 zurückgewiesen.

Suche nach r-Tripletts — Es wurde nicht für wenigstens zwei Spuren von Tochterteilchen des Zerfalls mindestens ein Triplett aus r-Clustern gefunden. Der hohe Wert an dieser Stelle erklärt sich allerdings dadurch, daß von sämtlichen dieser Tochterteilchen 58% weniger als drei Cluster in den r-Detektoren erzeugt haben und vom Algorithmus deshalb prinzipiell nicht gefunden werden können. Lediglich 2% gehen dadurch verloren, daß die Koordinaten im Rahmen ihrer Ortsauflösung und des eingestellten Akzeptanzfaktors nicht auf einer Gerade liegen. Solche Ereignisse könnten auch in der Offline-Analyse kaum rekonstruiert werden und sind daher ohnehin wertlos.

Entfernen überzähliger r-Tripletts — Durch das Entfernen eines von mehreren r-Tripletts, von denen fälschlicherweise angenommen wurde, daß sie dieselbe Teilchenbahn beschreiben, gehen Spuren aus dem B-Vertex verloren. Hierbei muß berücksichtigt werden, daß das verbleibende Triplett eine Spur mit ähnlichen Parametern beschreibt, mit deren Hilfe unter Umständen trotzdem noch der Vertex rekonstruiert werden kann, von dieser Statistik dann aber nicht anerkannt wird.

Kein Primärvertex — Aufgrund zu weniger 2D-Spuren oder Spurpaare mit großem Öffnungswinkel wurde für das Ereignis die Position des Primärvertex nicht berechnet, so daß auch die von Sekundärvertices nicht bestimmt werden konnte. Auch diese Ereignisse werden voraussichtlich Level-0 nicht passieren.

Selektion von 2D-Spuren — Durch die Auswahl von 2D-Spuren, aus denen 3D-Spuren berechnet werden, blieben weniger als zwei Tochterspuren des B-Vertex übrig.

Suche nach 3D-Spuren — Für die 2D-Tochterspuren des B-Zerfalls wurde kein passendes ϕ-Triplett gefunden, oder es war das falsche.

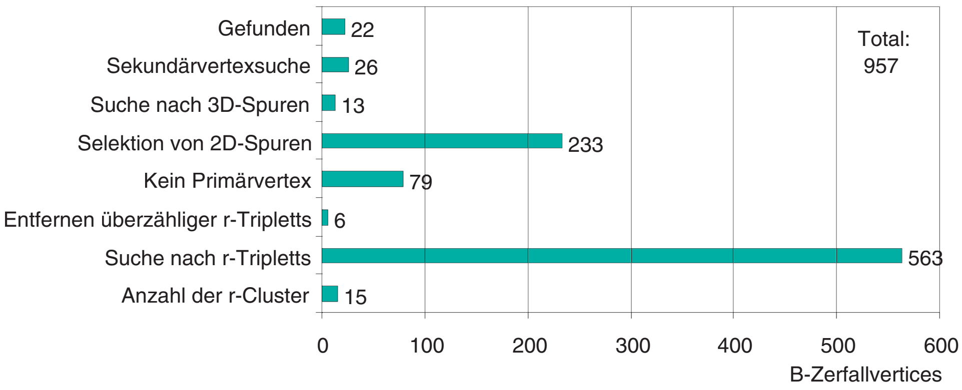


Abbildung 6.10: Statistik über den Verlust von Zerfallsvertices von B-Mesonen im Verlauf der Spuren- und Vertexsuche.

Sekundärvertexsuche — Die 3D-Spuren aus dem B-Vertex erfüllen nicht die geforderten Bedingungen für Sekundärvertices, d.h. deren minimaler Abstand ist zu groß oder die berechnete Position liegt zu nah am Primärvertex.

6.3 Geschwindigkeit

Sämtliche Zeiten, die nun vorgestellt werden, wurden auf einem handelsüblichen PC mit Pentium III 450 MHz CPU, 512 kB Cache und 128 MB Hauptspeicher gemessen, der im folgenden als Referenzsystem bezeichnet wird. Als Betriebssystem wurde Windows NT 4.0 und als Meßmethode der *Time Stamp Counter* des Pentium Prozessors eingesetzt [34]. Dabei handelt es sich um ein 64-Bit Register, dessen Inhalt mit jedem Taktzyklus des Prozessors inkrementiert wird. Um die Rechenzeit eines bestimmten Abschnitts der Anwendung zu messen, wird die Differenz dieses Zählers zu Anfang und Ende gebildet und mit der Dauer eines Zyklus multipliziert, die sich aus der Prozessorfrequenz zu 2,22 ns ergibt. Gemessen wurden nur die Teile der GAUDI-Anwendung, die für den Einsatz in der Experimentalumgebung von Bedeutung sind, also nur der elementare Trigger-Code und keine Ergebnis-Analysen oder das Lesen der Simulationsdaten von Festplatte und deren Konvertierung. Auch die Job-Konfiguration wurde entsprechend angepaßt, so daß keine Bildschirmausgabe erfolgt und keine Histogramme exportiert werden. Aktivitäten des Betriebssystes oder der Hintergrundprozesse während der Datenauswertung wurden jedoch mitgemessen und machen eine exakte Reproduktion der Werte unmöglich. Die Schwankungen lagen auf dem Referenzsystem für Durchschnittswerte pro Ereignis in der Größenordnung

Zeiten in μs	Minimum	Durchschnitt	Maximum
Gesamter Algorithmus	68 (57)	**3468 (945)**	38728 (14013)
2D-Spurensuche	181 (193)	1656 (1008)	11337 (6164)
Primärvertexsuche	51 (51)	993 (635)	8743 (4121)
3D-Spurensuche	98 (87)	906 (476)	18068 (7260)
Sekundärvertexsuche	37 (37)	85 (60)	625 (130)

Tabelle 6.2: Ausführungszeiten pro $B_d^0 \to \pi^+\pi^-$Ereignis (in Klammern für Minimum Bias).

von 100 μs. Mit Fluktuationen dieser Art muß auch auf dem Zielsystem gerechnet werden.

Tabelle 6.2 zeigt die Ergebnisse einer Messung aller 500 Ereignisse vom Typ $B_d^0 \to \pi^+\pi^-$ und Minimum Bias für den gesamten Algorithmus und seine Unterabschnitte. Angegeben sind die minimale, durchschnittliche und maximale Rechenzeit pro Ereignis, wobei für die Zeiten der Unteralgorithmen nur solche Ereignisse berücksichtigt wurden, in denen sie auch zur Ausführung kamen. Daher addieren sich deren Durchschnittszeiten nicht zu der des Gesamtalgorithmus. Die Verteilung der Gesamtdauer ist in Abbildung 6.11 aufgezeichnet. Das zeitliche Gewicht der einzelnen Teile des Gesamtprozedur ist in Abbildung 6.12 graphisch aufbereitet. Hier sind die Unteralgorithmen zum Teil in noch weitere Abschnitte aufgeteilt. Außerdem ist auch die Zeit enthalten, die durch den Transfer mit dem Transient Store und die Steuerung der Unteralgorithmen durch den Application Manager — also durch das Framework — verloren geht.

Einen wichtigen Zusammenhang läßt Abbildung 6.13 erkennen. Dieselben Zeiten sind hier für B-Ereignisse in Abhängigkeit von der Datenmenge aufgetragen. Als Grundlage für die Berechnung der Ereignisgröße dient das in Abschnitt 5.4.1 vorgestellte Datenformat, womit sich die Größe wie folgt berechnet:

$$\text{Clusteranzahl} \times 2\,\text{Bytes} + \underbrace{\text{Sektoranzahl} \times 4\,\text{Bytes}}_{=408\,\text{Bytes}} \qquad (6.1)$$

Man sieht, daß die für die Berechnung der Trigger-Entscheidung benötigte Zeit näherungsweise exponentiell mit der Cluster-Multiplizität ansteigt. Dies bedeutet, daß ernsthafte Probleme zu erwarten sind, falls die Ereignisgrößen im Experiment deutlich höher sein werden, als dies von der Simulation derzeit vorausgesagt wird.

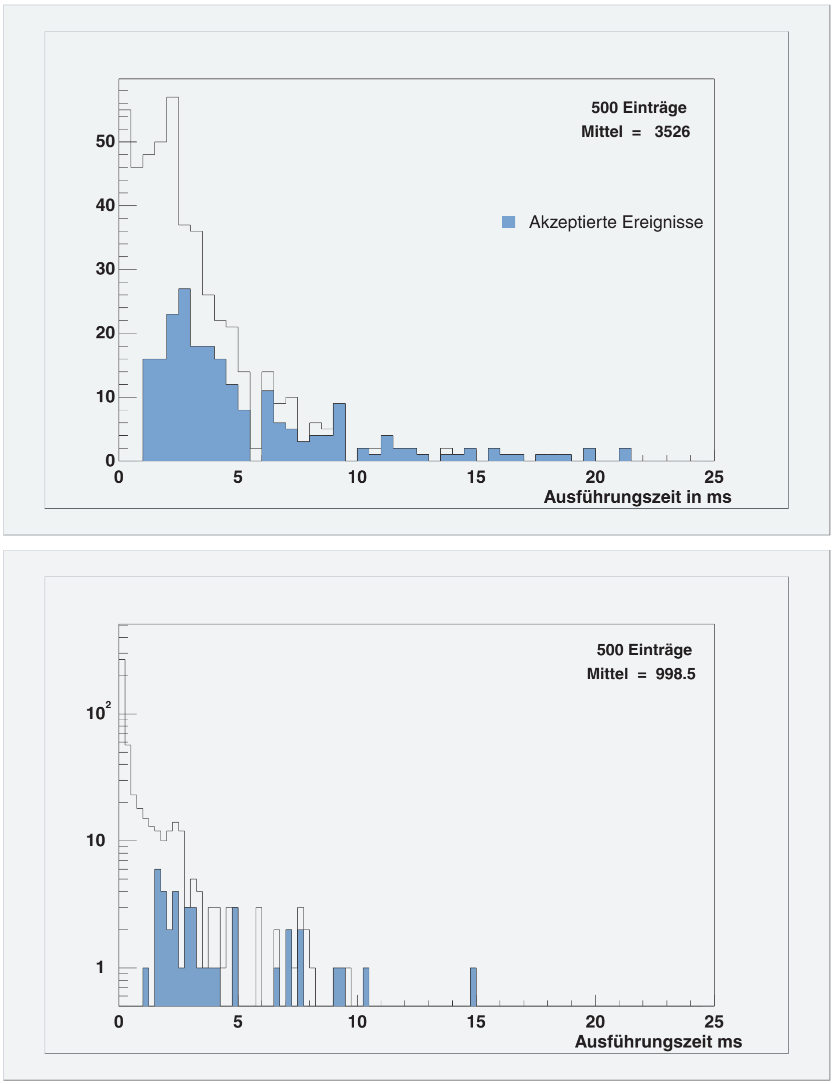


Abbildung 6.11: Rechenzeit des Vertex-Trigger-Algorithmus pro Ereignis auf dem Referenz-System. Der Anteil akzeptierter Ereignisse ist dunkel markiert.

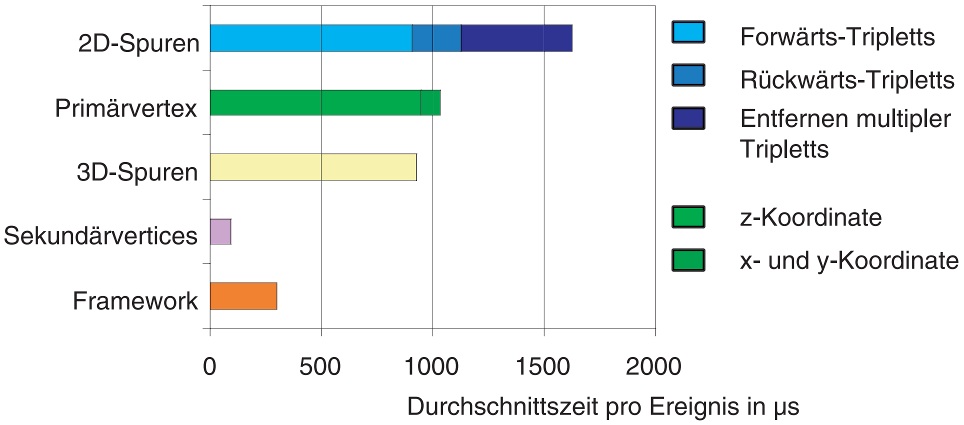
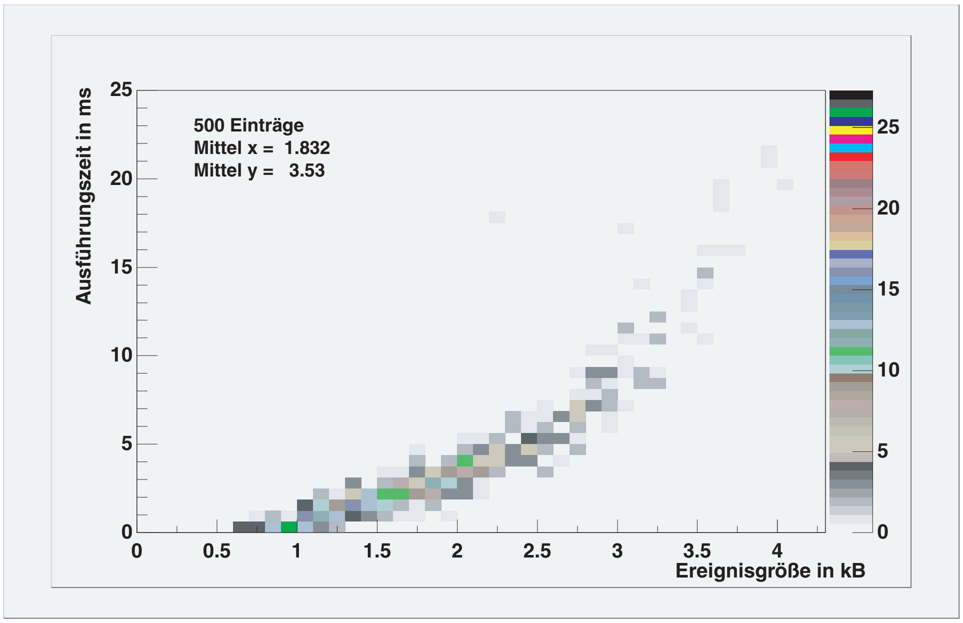


Abbildung 6.12: Durchschnittlicher Zeitverbrauch pro Ereignis der Unteralgorithmen und des Frameworks.



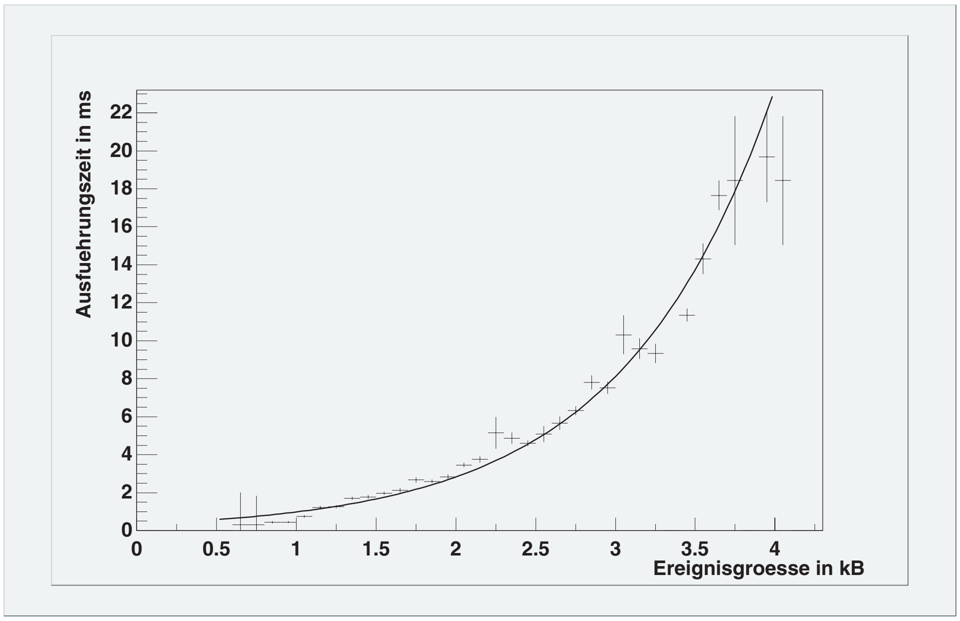


Abbildung 6.13: Berechnungsdauer von B-Ereignissen auf dem Referenzsystem in Abhängigkeit von der Ereignisgröße in zweidimensionaler Darstellung und als Projektion mit angepaßter Exponentialfunktion.

Kapitel 7

Zusammenfassung und Ausblick

Mit dieser Arbeit ist es gelungen, eine vollständige und funktionstüchtige Implementierung eines Vertex-Trigger-Algorithmus für LHCb im Rahmen der neu entwickelten Programmierumgebung GAUDI sowie eine Möglichkeit zu detaillierten Messungen seines Zeitverhaltens auf einem Referenz-System zu schaffen. Darüberhinaus wurden umfangreiche Analyse-Routinen zur Untersuchung der simulierten Ereignisse in Hinblick auf die Gestaltung und Konfiguration des Algorithmus sowie zur qualitativen Bewertung seiner Ergebnisse entwickelt. Die bislang daraus gewonnenen Erkenntnisse werden im folgenden kurz zusammengefaßt und diskutiert, bevor abschließend einige Probleme angesprochen werden.

Bei der Bewertung der erreichten Auflösungen und gefundenen Sekundärvertices darf nicht vergessen werden, daß der Vertex-Trigger-Algorithmus nicht den Anspruch erhebt, die Orte der Wechselwirkungen möglichst exakt zu rekonstruieren. Dies ist Aufgabe der Offline-Auswertung, bei der Geschwindigkeit nicht mehr die entscheidende Größe darstellt, weil mit aufgezeichneten Daten gearbeitet werden kann. Der Trigger muß vielmehr in möglichst kurzer Zeit einen zuverlässigen Hinweis auf die Existenz von B-Mesonen liefern.

Die Auswertungen der Ereignisse mit einem Zerfall vom Typ $\mathrm{B_d^0} \to \pi^+\pi^-$ resultierten in einer Akzeptanz-Rate von 33,0%, für Minimum Bias ergaben sich 5,2%. Während der Wert für B-Ereignisse als durchaus akzeptabel angesehen werden darf — da eine viel höhere Quote bei Zerfällen in nur zwei geladene Teilchen, von denen nur ein geringer Anteil in signifikanter Entfernung zum primären Vertex entsteht und damit nur sehr schwierig zu identifizieren ist — liegt die Akzeptanz für Minimum Bias eindeutig zu hoch. Zudem ist absehbar, daß dieser Wert noch weiter ansteigen wird, sobald Daten mit Level-0 Filterung untersucht werden. Dann wird sich nämlich die durchschnittliche Spurmultiplizität und dadurch bedingt auch die Zahl der gefundenen Sekundärvertices deutlich vergrößern.

Ein Vergleich mit den Resultaten des in SICb implementierten L1-Codes zeigt, daß mit der grundsätzlichen Funktionsweise bessere Ergebnisse produziert werden können. Dessen Akzeptanz liegt für dieselben B-Ereignisse bei vergleichbaren 35,6%, präsentiert sich aber effektiver bei der Abweisung von Minimum Bias, die nur zu 1,4% akzeptiert werden. Das Problem der hier vorgestellten Implementierung zusammen mit den verwendeten Einstellungen der Parameter liegt in der hohen Rate falscher Sekundärvertices, die einerseits die Akzeptanz von B-Ereignissen akzeptabel erscheinen lassen — trotz der unbefriedigenden Rate gefundener B-Zerfallsvertices — andererseits aber Minimum Bias Ereignisse mit hoher Spurmultiplizität mit zu großer Freizügigkeit akzeptieren. Es ist daher notwendig, in weiterführenden Analysen die Ursachen hierfür zu finden und zu eliminieren. Grundsätzliche Schwierigkeiten sind hierbei nicht zu erwarten, im Rahmen dieser Arbeit fehlte allein die Zeit dazu.

Als durchschnittliche Dauer, die zur Berechnung der Level-1 Entscheidung für Ereignisse mit Zerfällen vom Typ $\mathrm{B_d^0} \rightarrow \pi^+\pi^-$ auf dem Referenz-System benötigt wurde, ergaben sich etwa 3,5 ms. Dieser Wert ist allerdings nur für die Dimension des Computer-Clusters von Bedeutung. Die Leistungen der einzelnen Rechner sollte möglichst für alle Ereignisse ausreichen, weswegen momentan die entscheidendere Zeit für den Entwurf des Triggers die Maximaldauer ist. Diese betrug für besonders spurreiche Ereignisse bis zu 20 ms. Auch wenn die Dimensionen der geplanten Prozessor-Farm und die Latenz-Zeit für Level-1 noch nicht endgültig feststehen, kann bereits mit Sicherheit konstatiert werden, daß sich diese Zeiten noch erheblich reduzieren müssen. Derzeit sind als Latenz 1024 μs vorgesehen. Nach dem gegenwärtigen Stand der Dinge, was die vorausgesagten Geschwindigkeiten der Verteilung von Ereignisdaten auf die einzelnen Computer und die Strategie zur Pufferung dieser Daten angeht, stünden davon im günstigsten Fall nur etwa 887 μs für den Algorithmus als Rechenzeit zur Verfügung, unter Umständen sogar nur knapp 500 [35]. Allerdings wird bereits über eine Verdoppelung der Latenz-Zeit auf 2 ms diskutiert, wodurch aus den 887 dann 1912 μs würden.

Man darf damit rechnen, daß sich diese Anforderungen an den Algorithmus in Zukunft realisieren lassen. Zunächst kann davon ausgegangen werden, daß sich die Rechenleistung handelsüblicher PCs bis zum geplanten Einsatz im Jahr 2005 in etwa verzehnfachen wird [39]. Darüberhinaus sollte sich die Leistung des Trigger-Codes in Hinblick auf seine Geschwindigkeit noch so weit verbessern lassen, daß sich die benötigten Berechnungsdauern ungefähr halbieren. Anlaß zu dieser Prognose geben unter anderem folgende Punkte:

- Die Resultate gezielter Code-Optimierung in der 2D-Spurensuche, besonders durch eine ausgefeiltere Kombinationsmethode bei der Triplett-Suche, ergaben Geschwindigkeitsvorteile von etwa 50%. Im 3D-Bereich, wo dies bislang noch nicht durchgeführt wurde, darf mit ähnlichen Verbesserungen gerechnet werden.

- Durch Profiler-Analysen wurde die Erkenntnis gewonnen, daß die meiste Zeit bei der Triplett-Suche durch das Lesen von Cluster-Koordinaten verloren geht. An dieser Stelle muß die Zahl der nötigen Rechenoperationen (Umwandlung der Indizes, Berechnung der Clusteradresse, Umrechnung des verwendeten Integer-Formats in Fließkomma-Koordinaten) durch ein geschickteres Verfahren reduziert werden.

- Die besonders rechenintensiven Fehlerformeln, die für jedes potentielle Triplett neu berechnet werden müssen, können durch wesentlich effektivere Fehlertabellen ersetzt werden.

- Die Histogramm-Methode zur Positionsbestimmung des Primärvertex ist bei weitem noch nicht optimal. Bei der Erstellung der 2D-Spuren könnte beispielsweise der Mittelwert der z-Achsenabschnitte berechnet und dadurch der Bereich für die z-Koordinate bereits abgeschätzt werden. Die Dimension des Histogramms und somit die benötigte Zeit für dessen Initialisierung und die Suche nach Häufungspunkten ließe sich auf diese Weise beschleunigen.

- Die im Framework verbrauchte Zeit wird sich mindestens halbieren, wenn dessen Laufzeitbibliotheken auch als zeitoptimierte Compilierungen zur Verfügung stehen.

Unter Annahme der angestellten Schätzung lägen die durchschnittlichen Zeiten unter 200 μs und die maximalen unter 1 ms, was zumindest im Fall einer Latenz-Zeit von 2 ms mehr als ausreichend wäre. Weiteres Potential für eine Beschleunigung darf auch in einer grundsätzlichen Änderung der Strategie bei der Suche nach Sekundärvertices und einem für den Algorithmus günstigerem Format der Ereignisdaten vermutet werden. Hierzu wurden allerdings noch keine Untersuchungen angestellt, daher wäre eine Abschätzung des dadurch zu erwartenden Zeitgewinns an dieser Stelle zu verwegen. Voraussetzung für all diese Überlegungen ist allerdings, daß sich die tatsächlichen Ereignisse im Experiment in der Anzahl ihrer Treffer auf den Stationen des Vertex-Detektors nicht wesentlich von denen in der Simulation vorausgesagten Werten abheben, da der Algorithmus in seiner gegenwärtigen Form darauf mit einem näherungsweise exponentiellen Anstieg seiner Ausführungsdauer reagieren würde. Die Architektur sollte daher auf jeden Fall flexibel genug sein, um in diesem Fall an die gestiegenen Ansprüche angepaßt werden zu können, auch wenn dadurch ungeplante Kosten entstünden.

Wenn in der endgültigen Implementierung die Berechnung eines Ereignisses dennoch länger als die vorgegebene Maximalzeit dauern sollte, müßte sie abgebrochen werden. Sollte eine Analyse neuer Daten ohne B-Zerfall ergeben, daß deren Berechnungsdauern stets unter dieser Zeit liegen, könnten solche Ereignisse dann grundsätzlich akzeptiert werden. Dies erscheint jedoch unwahrscheinlich, was wiederum bedeutet, daß diese Ereignisse abgelehnt werden müßten. Besonders große B-Ereignisse würden dann von Level-1 nicht erkannt.

Es muß noch darauf hingewiesen werden, daß manche Ergebnisse zum Teil — hauptsächlich für den Fall der Auswertung von Minimum Bias Ereignissen — nur bedingt aussagekräftig sind, was den späteren Einsatz des Algorithmus in der Experimentalumgebung angeht. Der Grund hierfür liegt darin, daß die verwendeten Simulationsdaten keine Selektion durch den Level-0 Trigger erfahren haben und frei von Ereignissen mit mehr als einer primären Wechselwirkung sind. Dadurch erreichen zum Beispiel besonders kleine Minimum Bias Ereignisse den L1-Algorithmus und schönen an dieser Stelle die Statistik der Zeitmessungen. Neuere und realitätsnähere Daten können zwar bereits durch den Einsatz von SICb erzeugt werden, allerdings gelang es der Software-Gruppe am CERN bis dato nicht, einen fehlerfreien Zugriff auf diese Ereignisse im Rahmen von GAUDI zu ermöglichen [36]. Dabei handelt es sich um die Auswirkung einer Gegebenheit, welche die gesamte Entstehung dieser Arbeit begleitete, nämlich daß sich dieses Rahmenprogramm zeitgleich mit dem Algorithmus in Entwicklung befand. Da es bei Software-Projekten dieses Umfangs unmöglich ist, von Beginn an absolut fehlerfreie und problemlos zu installierende Versionen anzubieten, kam es gelegentlich zu leichten Verzögerungen beim Umstieg auf aktuelle Versionen oder beim Zugriff auf neue Simulationsdaten. Solche Anlaufschwierigkeiten zu beheben, setzt den praktischen Einsatz voraus, weswegen diese Arbeit durchaus als Hilfe hierzu verstanden werden darf. Auch die Tatsache, daß GAUDI außerhalb der Software-Umgebung vom CERN eingesetzt wurde, war nicht ausschließlich hilfreich. Zwar wurde durch die lokale Präsenz der Simulationsdaten deren Auswertung beschleunigt, der Zugriff auf größere Mengen von Daten und Ereignisse anderen Typs gestaltete sich jedoch infrastrukturell bedingt zuweilen etwas umständlich.

Für die weitere Entwicklung des Algorithmus ist vorgesehen, daß als nächstes neuere Daten bearbeitet werden, und das in größerem Umfang, als bislang geschehen ist. Dazu sollte der Algorithmus für den Einsatz auf Computern am CERN gerüstet werden. Dies erfordert auch den Einsatz der Detector Description Data Base[1], so daß bei Änderungen im Design des Vertex-Detektors der Code unverändert bleiben kann. Außerdem sollte der

[1]Datenbank, in der die physikalische und logische Beschreibung der Detektoren für den Zugriff innerhalb von GAUDI gespeichert ist.

Mechanismus zur Zeitmessung, der momentan Pentium-Prozessoren voraussetzt, architekturunabhängig gestaltet werden, um zum Beispiel Messungen mit Alpha-CPUs zu ermöglichen. Im weiteren gilt es auch zu untersuchen, wie sich die Belastung des Speicherbusses durch Aktivitäten der Netzwerkkarte auf das Geschwindigkeitsverhalten auswirkt.

Abschließend läßt sich festhalten, daß trotz der angesprochenen Schwierigkeiten eine Vorstellung der zur Berechnung benötigten Zeiten und deren Abhängigkeit von der Ereignisgröße gewonnen und somit für die Gestaltung der Hardware wertvolle Ergebnisse erzielt wurden. Außerdem existiert nun eine Grundlage für weiterführende Untersuchungen und Optimierungen des Level-1 Vertex-Trigger-Algorithmus.

Anhang A

Die CD

Auf der beiliegenden CD befindet sich eine Auswahl der im Laufe dieser Arbeit entstandenen Dateien, unter anderem der Algorithmus als ausführbares Programm. Die untersuchten Simulationsdaten finden hier keinen Platz, da deren Größe für 500 Ereignisse im Bereich von 1 GB liegt. Eine detaillierte Gebrauchsanweisung für den Einsatz des Algorithmus entsteht parallel zu dieser Arbeit und wird gesondert erscheinen. An dieser Stelle sei auf die Dateien `Readme.txt` in den einzelnen Verzeichnissen verwiesen. In den Verzeichnissen auf der CD befinden sich folgende Inhalte:

Diplomarbeit — Diese Arbeit als Postscript-Datei mit allen zugehörigen LaTeX-Quelltexten und sämtlichen Abbildungen im EPS-Format (Encapsulated Postscript).

Ergebnisse — Die Text-Ausgaben und Dateien mit Histogrammen der Jobs, mit denen die vorgestellten Ergebnisse erhalten wurden.

Quelltexte — Sämtliche Quelltexte der Implementierung, die zugehörigen Konfigurationsdateien für CMT[1], die Projektdateien für Microsoft Visual C++ und Makros für ROOT[2], mit denen die meisten der vorgestellten Histogramme erzeugt wurden.

WinNT — Der Vertex-Trigger-Algorithmus als ausführbare Datei für Windows NT, die benötigten Gaudi-Laufzeitbibliotheken und die Konfigurationsdateien.

[1] Configuration Management Tool, mit dem die Project- und Makefiles für Gaudi-Anwendungen erstellt werden

[2] Für den Einsatz in der Hochenergiephysik entwickeltes Programm zur Datenanalyse und Erstellung von Grafiken

Arbeiten am Quellcode setzen die Installation von GAUDI (mindestens Version 4) voraus, während für die Ausführung der Programme aus dem WinNT-Verzeichnis lediglich der Zugriff auf Simulationsdaten ermöglicht werden muß.

Anhang B

Glossar

ALICE (**A** **L**arge **I**on **C**ollider **E**xperiment) — Geplanter Detektor am LHC zur Untersuchung von Schwerionen-Kollisionen.

ATLAS (**A** **T**oroidal **L**HC **A**pparatu**s**) — Geplanter Detektor am LHC zur Untersuchung von Protonen-Kollisionen.

BABAR — Detektor am PEP-II zur Erforschung von B-Mesonen.

BELLE — Experiment am KEKB zur Untersuchung der Zerfälle von B-Mesonen.

BTeV — Geplanter Detektor am Tevatron zur Erforschung der Zerfälle von B-Mesonen.

CDF (**C**ollider **D**etector at **F**ermilab) — Detektor am Tevatron zur Untersuchung von Kollisionen zwischen Protonen und Antiprotonen.

CERN (**C**entre **E**uropéenne de **R**echerche **N**ucleaire) — Europäisches Kernforschungszentrum bei Genf (Schweiz).

CMS (**C**ompact **M**uon **S**olenoid) — Geplanter Detektor am LHC zur Untersuchung von Protonen-Kollisionen.

D0 — Detektor am Tevatron zur Untersuchung von Kollisionen zwischen Protonen und Antiprotonen.

DESY (**D**eutsches **E**lektronen-**Sy**nchrotron) — Zentrum für Grundlagenforschung in Hamburg und Zeuthen (Deutschland).

ECAL (**E**lectromagnetic **Cal**orimeter) — Kalorimeter zur Messung elektromagnetisch wechselwirkender Teilchen.

Fermilab (**Fermi** National Accelerator **Lab**oratory) — Forschungseinrichtung für Hochenergiephysik bei Chicago (USA).

GAUDI (LHCb Data Processing Applications Framework) — Rahmenprogramm für Anwendungen zur Verarbeitung von Daten des LHCb-Experiments.

HCAL (**H**adronic **Cal**orimeter) — Kalorimeter zur Messung von hadronisch wechselwirkenden Teilchen.

HERA (**H**adron **E**lectron **R**ing **A**ccelerator) — Speicherring für Protonen und Elektronen am DESY.

HERA-B — Detektor am Speicherring HERA zur Erforschung der Zerfälle von B-Mesonen.

KEK — Forschungsorganisation für Hochenergiephysik in Tsukuba (Japan).

KEKB — Asymmetrischer Speicherring für Elektronen und Positronen am KEK für die Erzeugung von B-Mesonen.

LEP (**L**arge **E**lectron **P**ositron Collider) — Speicherring für Elektronen und Positronen am CERN.

LHC (**L**arge **H**adron **C**ollider) — Geplanter Speicherring für Protonen und Ionen am CERN.

LHCb (**L**arge **H**adron **C**ollider **B**eauty Experiment) — Geplanter Detektor am LHC zur Präzisionsmessung von CP-Verletzung und schwacher Zerfälle von B-Mesonen.

PEP-II (**P**ositron **E**lectron **P**roject **II**) — Asymmetrischer Speicherring für Elektronen und Positronen am SLAC.

PS (**P**roton **S**ynchrotron) — Vorbeschleuniger für den SPS.

RICH (**R**ing **I**maging **Ch**erenkov Counter) — Spezieller Čerenkov-Zähler.

SICb — Programmpaket zur Monte-Carlo-Simulation von Ereignissen für das LHCb Experiment.

SLAC (**S**tanford **L**inear **A**ccelerator **C**enter) — Forschungseinrichtung der Stanford University (USA).

SPS (**S**uper **P**roton **S**ynchrotron) — Vorbeschleuniger für den LEP/LHC.

Tevatron — Speicherring für Protonen und Antiprotonen am Fermilab.

Literaturverzeichnis

[1] B. R. Martin und G. Shaw, *Particle Physics*, John Wiley & Sons 1992

[2] R. K. Bock, H. Grote, D. Notz und M. Regler, *Data analysis techniques for high-energy physics experiments*, Cambridge University Press 1990

[3] Salvatore Mele, *Experimental Constraints of the Cabbibo-Kobayashi-Maskawa Matrix*, CERN

[4] Rick Jesik, *D0 Expectations and Aspirations for B Physics in Run II*, Workshop on B Physics at the Tevatron, 23. September 1999

[5] F. Eisele, *Experimente zur Untersuchung der CP-Verletzung mit B-Mesonen an Hadronmaschinen*, http://www1.physi.uni-heidelberg.de/~eisele/bphysik.htm

[6] *BABAR Collaboration Homepage*, http://www.slac.stanford.edu/BFROOT

[7] *BELLE Homepage*, http://bsunsrv1.kek.jp

[8] *HERA-B Homepage*, http://www-hera-b.desy.de

[9] *CDF Homepage*, http://www-cdf.fnal.gov

[10] *The D0 Experiment*, http://www-d0.fnal.gov

[11] *The BTeV Project at Fermilab*, http://www-btev.fnal.gov

[12] *The Large Hadron Collider Project*, http://lhc.web.cern.ch/lhc

[13] *ALICE Homepage*, http://alice.web.cern.ch/Alice

[14] *ATLAS General Sites*, http://atlas.web.cern.ch/Atlas

[15] *CMS Homepage*, http://cmsinfo.cern.ch

[16] *LHCb Homepage*, http://lhcb.web.cern.ch/lhcb

[17] *LHCb Technical Proposal — A Large Hadron Collider Beauty Experiment for Precision Measurements of CP Violation and Rare Decays*, CERN/LHCC 98-4, 20. Februar 1998

[18] A. Wiebalck, *Untersuchung von Kompressionsverfahren für die Daten der ALICE TPC*, 5. Oktober 2000

[19] K. Kleinknecht, *Detektoren für Teilchenstrahlung*, B. G. Teubner Stuttgart 1984

[20] A. Walsch, *Level-1 Trigger Status Report*, 26. September 2000, http://lhcb-trig.web.cern.ch/lhcb-trig/pdf/2000_09_26_walsch.pdf

[21] T. Bowcock, M. McCubbin, C. Parkes, H. Dijkstra, F. Fiedler, B. Hay, T. Ruf, N. van Bakel, J. van den Brand, M. Ferro-Luzzi, T. Ketel und P. Koppenburg, *VELO Geometry Optimisation*, LHCb 2000-90 VELO, 10. Oktober 2000

[22] Hans Dijkstra und Thomas Ruf, *The L1 Vertex Trigger Algorithm and its Performance*, LHCb 98-006 TRIG, 22. Januar 1998

[23] Eric van Herwijnen, *LHCb Software Strategy*, LHCb 98-025 COMP, 6. Februar 1998

[24] *LHCb Computing Tasks*, LHCb 98-042 COMP, 9. März 1998

[25] *GAUDI — Architecture Design Document*, LHCb 98-064 COMP, 24. November 1998

[26] *GAUDI — User Guide Version 4*, 14. April 2000

[27] M. Frank und B. Hay, *Accessing Level 1 and Velo SICb data within the Gaudi Framework — Classes and access methods*, LHCb 2000-009 COMP, 7. Februar 2000

[28] M. Bouianov, *Development of the Level-1 Vertex Trigger Software in Gaudi Data Processing Applications Framework*

[29] Y. Ermoline, V. Lindenstruth und A. Walsch, *LHCb Level-1 Vertex Topology Trigger, Requirements and Interface Specifications*, LHCb 99-031, 23. August 1999

[30] *SICb User Guide*, 19. Juni 1997, http://lhcb-comp.web.cern.ch/lhcb-comp/SICB/html/sicbug.html

[31] D. Steele, *The $R-\Phi$ Silicon Vertex Detector Simulation as Implemented in SICB*, LHCb 98-004 TRAC, 20. Januar 1998

[32] *GEANT — Detector Description and Simulation Tool*, CERN Program Library Long Writeup W5013, Oktober 1994

[33] Torbjörn Sjöstrand, *PYTHIA 5.7 and JETSET 7.4 – Physics and Manual*, Dezember 1998

[34] *Intel Architecture Software Developer's Manual, Volume 2: Instruction Set Reference*, 1997

[35] Markus Schulz, *Präsentation zum L1 Status*, 26. September 2000, http://lhcb-trig.web.cern.ch/lhcb-trig/pdf/2000_09_26_schultz.pdf

[36] Bruce Hay, LHCb Software-Gruppe am CERN, Persönliche Korrespondenz

[37] John L. Hennessy und David A. Patterson, *Computer Architecture — A Quantitative Approach*, Second Edition, 1996

[38] Gregory F. Pfister, *In Search of Clusters — The ongoing Battle in Lowly Parallel Computing*, Second Edition, 1998

[39] *What is Moore's Law*, http://www.intel.com/intel/museum/25anniv/hof/moore.htm

Danksagung

An dieser Stelle möchte ich mich bei allen Mitgliedern unseres Lehrstuhls, meinen Freunden und Bekannten sowie meiner Familie bedanken, die mir direkt oder indirekt beim Zustandekommen dieser Arbeit geholfen haben.

An erster Stelle sind dies Prof. Volker Lindenstruth, der mir — wo immer dies möglich war — bei der Beseitigung von Problemen aller Art zur Seite stand und durch seinen Optimismus und seine Zielstrebigkeit wesentlichen Anteil am Fortschritt dieser Arbeit hatte, und insbesondere mein Betreuer Dr. Markus Schulz, der Garant für das Gelingen meiner Präsentationen am CERN war und mich mit unzähligen wichtigen Hinweisen während der gesamten Arbeit versorgt hat, die ich ausnahmslos dankbar entgegegenommen und — abgesehen von einigen Vorschlägen zur farblichen Gestaltung mancher Abbildungen — so gut wie möglich verwertet habe.

Unmöglich wäre das Gelingen auch ohne die Hilfe vieler Mitarbeiter am CERN gewesen, die mir die benötigten Simulationsdaten beschafft und mich bei der Installation und Anwendung von GAUDI unterstützt haben. Dabei sind Markus Frank und Bruce Hay hervorzuheben, die nicht müde wurden, meine unzähligen Fragen zu beantworten und mir so die Lösung vieler Probleme ermöglichten.

Weiterhin möchte ich Sorina Popescu danken, die mir während ihres leider nur kurzen Aufenthalts an unserem Lehrstuhl dazu verhalf, mich in die Methoden der qualitativen Analyse meiner Ergebnisse einzuarbeiten, und darüberhinaus durch ihre angenehme Art für eine willkommene Abwechslung während der Arbeit sorgte.

Besonders hilfreich waren auch alle, die neben meinem Betreuer die vorliegende Ausarbeitung zur Korrektur gelesen haben. Hier möchte ich Alexander Walsch nennen, der mich darüberhinaus mit wichtigen Informationen zur Entwicklung der Trigger-Hardware versorgt hat, und meinen Vater, ohne dessen Einsatz eine Arbeit mit wesentlich mehr Fehlern „enstanden" wäre.

Nicht zuletzt danke ich auch den Sekretärinnen und dem Verwaltungspersonal an unserem Institut für die Unterstützung bei organisatorischen und verwaltungstechnischen Angelegenheiten, ganz besonders Béatrice Bähr, die Problemen jeder Art mit guter Laune und raschen, unkomplizierten Lösungen begegnete.

Abschließend gilt mein spezieller Dank Oliver Klimm, ohne dessen Hinweis in letzter Minute ich unter dem Zeitdruck des Abgabetermins diese Danksagung vergessen hätte.