RUPRECHT-KARLS-UNIVERSITÄT HEIDELBERG

Arne Wiebalck

Untersuchung von Kompressionsverfahren für die Daten der ALICE TPC

Diplomarbeit

HD-KIP-00-50

KIRCHHOFF-INSTITUT FÜR PHYSIK

Untersuchung von Kompressionsverfahren für die Daten der ALICE TPC

Die Diplomarbeit wurde von Arne Wiebalck ausgeführt am

Kirchhoff-Institut für Physik

unter der Betreuung von

Herrn Prof. Dr. Volker Lindenstruth

Inhalt

In dieser Arbeit werden verschiedene Verfahren zur Kompression von Daten der Zeit-Projektions-Kammer des am Teilchenforschungszentrum CERN geplanten Schwerionen-Experiments ALICE untersucht. Neben der verlustlosen Entropie-Codierung nach Huffman wird der Schwerpunkt auf die verlustbehaftete Vektorquantisierung gelegt. Bereits existierende Bild- und Videokompressionsverfahren werden ebenfalls hinsichtlich ihrer Anwendbarkeit untersucht. Für alle verlustbehafteten Verfahren wird die Datenqualität nach Encodierung und Decodierung durch Analyse und Vergleich mit den Resultaten der unbehandelten Daten bestimmt. Die Huffman-Codierung reduziert die Datenmenge auf etwa zwei Drittel, die Vektorquantisierung um einen Faktor 2, ohne die Datenqualität wesentlich zu beeinträchtigen. Bild- und Videokompressionsverfahren sind aufgrund ihrer starken Auswirkungen auf die Qualität der Daten nicht anwendbar.

Abstract

In this thesis several techniques for the compression of data produced by the time projection chamber of the planned heavy ion experiment ALICE at the particle physics center CERN are examined. Apart from the entropy coding by Huffman the emphasis is put on lossy vector quantization. Image and video compression techniques are likewise examined regarding their applicability. For all lossy procedures the data quality is evaluated after encoding and decoding by analysis and comparison with the results of the untreated data. The Huffman coding reduces the data volume to about two thirds, vector quantization by a factor of 2, without impairing the data quality substantially. Image and video compression techniques are not applicable due to their strong impact on the data.

Inhaltsverzeichnis

Kapitel 1

Einleitung

Jedem Anfang wohnt ein Zauber inne.
Hermann Hesse (1877-1962)

Im Jahre 1929 gelang dem amerikanischen Astronom Edwin P. Hubble der experimentelle Beweis, daß sich das gesamte Universum ausdehnt. Er kombinierte dazu die Ergebnisse der Arbeiten von Henrietta Leavitt zur Entfernungsbestimmung von Sternen sowie diejenigen von Vesto Slipher über die Rotverschiebung in den Spektren sich entfernender Objekte. Diese Erkenntnisse bilden eine Säule des Standardmodells der Kosmologie: der Urknall-Theorie.

Dieser Theorie zufolge entstand das Universum vor etwa 15 Milliarden Jahren aus einer unendlich heißen und dichten Singularität, die sich explosionsartig ausdehnte. Durch diese rapide Expansion kühlte sich das Universum bis auf etwa 3 Grad über dem absoluten Nullpunkt ab, seinen heutigen Wert.

Das junge Universum durchlief dabei verschiedene Phasenübergänge. So befanden sich in der Anfangsphase alle Teilchen im thermodynamischen Gleichgewicht, die Energie reichte aus, um alle Teilchen ineinander umzuwandeln. Es gab keine Unterschiede zwischen Leptonen und Quarks, die im Standardmodell der Teilchenphysik zusammen mit den Austauschteilchen (dem Photon, den Gluonen und den Eichbosonen $W^{\pm}$und Z^0), die die Kräfte zwischen ihnen vermitteln, die Bausteine der Materie bilden. Die Gleichheit aller Teilchen implizierte die Gleichheit aller Kräfte. Bereits nach etwa 10^{-35} Sekunden jedoch war die Temperatur so weit gefallen, daß die Entkopplung von starker und elektroschwacher Kraft stattfand. Nach 10^{-11} Sekunden koppelte sich dann die schwache von der elektromagnetischen Kraft ab. 10^{-6} Sekunden nach dem Urknall und bei einer Temperatur von ca. 150 MeV[1] befand sich das Universum in einem Zustand, in dem Quarks und Gluonen sich frei bewegen konnten, bevor die Energieskala erreicht wurde, in der sie Bindungszustände in Form von Mesonen und Baryonen eingehen.[2] Dieser Zustand freier Quarks und Gluonen

[1]Die Angabe der Temperatur in Elektronenvolt ist hier üblich. 1 eV entspricht dabei etwa 11600 K.
[2]Mesonen sind Teilchen, die aus einem Quark-Antiquark-Paar aufgebaut sind und ganzzahligen Spin

vor der Bildung von Hadronen in der sogenannten Hadron-Ära wird in Analogie zur Atomphysik und der Dissoziation von Atomen in Elektronen und Kerne als Quark-Gluon-Plasma bezeichnet.

Zur Erzeugung, Nachweis und Untersuchung dieses Zustandes der Materie in der Frühphase des Universums, werden in Schwerionenexperimenten Atomkerne bei so hohen Energien miteinander zur Kollision gebracht, daß die Kerne in Protonen und Neutronen aufbrechen. Bei Energiedichten, die 20fach größer sind als diejenige normaler Materie, und Temperaturen,[3] die mehr als 100000 mal so groß sind wie die im Kern der Sonne, brechen auch die Nukleonen noch in ihre Bestandteile, Quarks und Gluonen, auf. Diese können nun nicht mehr einzelnen Nukleonen zugeordnet werden. Es ensteht ein Quark-Gluon-Plasma.

Abb. 1.1 zeigt in Abhängigkeit von der Temperatur und dem baryochemischen Potential, einer Größe, die in diesem Zusammenhang üblicherweise verwendet wird und eng mit der Dichte der Baryonen zusammenhängt, schematisch das Phasendiagramm der Kernmaterie mit dem Verlauf der Phasengrenze zwischen Hadronengas und Quark-Gluon-Plasma.

Die Theorie der statistischen Quantenchromodynamik (QCD)[4] sagt als kritische Temperatur für diesen Phasenübergang in Kern-Kern-Stößen einen Wert von 150 - 170 MeV voraus. Die experimentelle Untersuchung ist ein wichtiger Test dieses Aspektes der Theorie der starken Wechselwirkung. Ferner ist dieser Phasenübergang der einzige mit Laborenergien zugängliche, bei dem Quantenfelder von Elementarteilchen involviert sind, und somit das Verständnis seiner Dynamik von generellem Interesse.

Ähnlich wie beim Universum kühlt auch hier das Quark-Gluon-Plasma aus. Es bilden sich Hadronen, die in den Detektoren nachgewiesen werden können und aus denen Rückschlüsse auf die Zustände während der Kollision gezogen werden müssen.

So geschieht der Nachweis des Quark-Gluon-Plasmas indirekt neben anderen Kriterien über die Produktionsrate verschiedener Teilchen. Die Vektormesonen J/ψ zum Beispiel entstehen zu Anfang einer Kollision, wenn die kollidierenden Teilchen noch ihre volle Energie besitzen. Bei Verwendung schwererer Kerne und Erhöhung der Zentralität der Kollision, wird ein vermindertes Auftreten der J/ψs gemessen.[5] Die Erklärung findet sich im Phasenübergang zum Quark-Gluon-Plasma: Ein Teil der J/ψs wird hierbei zerstört und ihre Gesamtzahl liegt am Ende der Kollision unter derjenigen für Kollisionen, bei denen die Phasengrenze nicht überschritten wurde. Ein Signal bei der Entstehung des Quark-Gluon-Plasmas ist also die Unterdrückung dieser Mesonen.

---

tragen, Baryonen bestehen aus drei Quarks und haben halbzahligen Spin. Mesonen und Baryonen werden zu den Hadronen, den Teilchen, die der starken Wechselwirkung unterliegen, zusammengefaßt.

[3] Von *Temperaturen* darf man bei diesen Kollisionen deshalb sprechen, weil die Dichte der Materie und damit die Zahl der Wechselwirkungen so hoch ist, daß sich sehr schnell ($\tau \leq 1$ fm/c $\approx 3 \cdot 10^{-24}$ s) ein thermisches Gleichgewicht ausbilden kann.

[4] Bei diesen Temperaturen ist die effektive Kopplungskonstante der QCD in etwa 1, so daß störungstheoretische Techniken nicht anwendbar sind.

[5] Diese Unterdrückung der Charmonium-Zustände ist ein Ergebnis der Messungen aus Schwefel-Uran-Kollisionen beim CERN-Experiment NA38 und aus Blei-Blei-Kollisionen beim NA50-Experiment, das ebenfalls am CERN durchgeführt wurde.

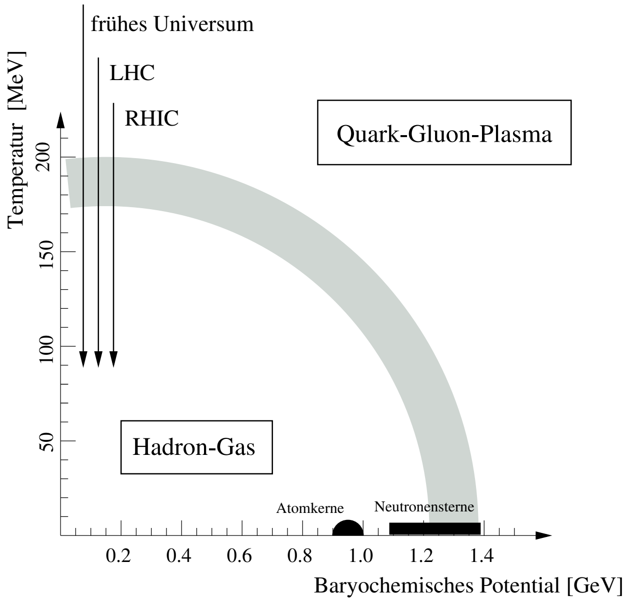


Abbildung 1.1: Phasendiagramm der Kernmaterie. Die Pfeile kennzeichnen die Orte des Phasenübergangs beim Urknall und bei den Beschleunigern LHC und RHIC nahe einem baryochemischen Potential von 0.

Andere Teilchen treten vermehrt auf. So wird in Kern-Kern-Reaktionen zum Beispiel gegenüber Elektron-Elektron- oder auch Proton-Proton-Kollisionen eine Anreicherung mit Teilchen gemessen, die Strange-Quarks enthalten. Da nach der Formation von Hadronen keine Strange-Quarks mehr erzeugt werden können, müssen diese vorher in einer Phase wechselwirkender Quark-Gluon-Materie entstanden sein. Der Anstieg in der Produktion von diesen Teilchen ist also ebenfalls ein Hinweis auf die Existenz des Quark-Gluon-Plasmas.[6]

In verschiedenen Experimenten[7] am Teilchenforschungszentrum CERN[8] bei Genf konnten Indizien für die Existenz einer solchen Phase gefunden werden, so daß im Februar 2000 offiziell die Existenz dieses neuen Zustandes der Materie bekannt gegeben wurde [19].

Neue Schwerionenexperimente am RHIC-Kollider[9] am Brookhaven National Laboratory und am geplanten LHC-Beschleuniger[10] am CERN werden diese Untersuchungen bei weit höheren Energien fortführen, um das Quark-Gluon-Plasma im Detail zu studieren. Da hier Experimente bei weit höheren Energien durchgeführt werden, erwartet man auch direktere

[6] Diese Anreicherung von Hadronen, die Strange-Quarks enthalten, konnte bei den CERN-Experimenten NA49, NA50 und WA97 festgestellt werden.

[7] Beteiligt waren hieran die Experimente NA44, NA45/CERES, NA49, NA50, NA52/NEWMASS, WA97/NA57 und WA98.

[8] **C**onseil **E**uropéen pour la **R**echerche **N**ucléaire (European Organization for Nuclear Research).

[9] **R**elativistic **H**eavy **I**on **C**ollider.

[10] **L**arge **H**adron **C**ollider.

Hinweise auf die Formation eines Quark-Gluon-Plasmas. Hierzu zählt z.B. die Messung von Photonen, die in der heißen Anfangsphase direkt von den Quarks im Plasma emittiert werden.[11]

Bei diesen Experimenten werden pro Kollision Zehntausende von Teilchen entstehen, die in den Detektoren nachgewiesen werden müssen. Um diese hohe Teilchenmultiplizität zu bewältigen, wird insbesondere eine der Hauptkomponenten des Schwerionen-Experiments ALICE am LHC, die Zeit-Projektionskammer, mit einer hohen räumlichen und zeitlichen Granularität arbeiten. Dies führt zu Datenmengen von mehr als 10 Gigabyte pro Sekunde, die über ein Netzwerk verschickt und gespeichert werden müssen.

Ziel dieser Arbeit ist die Untersuchung verschiedener Kompressionsverfahren und, falls sie verlustbehaftet sind, ihre Auswirkungen auf die Datenqualität. Die Datenmenge dieses Subdetektors soll so weit wie möglich reduziert werden, um sowohl die Anforderung an das Netzwerk als auch die Kosten zur Speicherung zu minimieren.

Die Arbeit gliedert sich wie folgt: Im zweiten Kapitel werden der Beschleuniger LHC und das Experiment ALICE kurz vorgestellt. Es folgt im dritten Kapitel eine Einführung in verschiedene Kompressionsverfahren, bevor im vierten Kapitel verschiedene Techniken auf simulierte Daten der Zeit-Projektions-Kammer des ALICE-Experiments angewendet werden. Im fünften Kapitel werden dann die Auswirkungen verlustbehafteter Kompression auf die in den Daten enthaltene Physik untersucht. Die erhaltenen Ergebnisse werden in Kapitel 6 zusammengestellt und es wird ein kurzer Ausblick gegeben.

[11] Untersuchungen dieser Art wurden bereits bei den CERN-Experimenten WA98, NA45 und NA50 am Super Proton Synchrotron (SPS) durchgeführt. Allerdings sind sie wegen des großen Untergrundes aus anderen Quellen sehr schwierig und so konnte bisher der Nachweis dieser thermischen Strahlung des Plasmas nicht mit Sicherheit erbracht werden.

Kapitel 2

Das Experiment

Es genügt nicht, zum Fluß zu kommen
mit dem Wunsche, Fische zu fangen.
Man muß auch das Netz mitbringen.
Chinesisches Sprichwort

2.1 Der LHC-Beschleuniger

Der Betriebsbeginn des großen Hadron-Kolliders (Large Hadron Collider, LHC) [28] am CERN ist für das Jahr 2005 vorgesehen. Er wird im bestehenden 27 km langen Tunnel des Elektron-Positron-Beschleunigers LEP[1] installiert werden. Protonen werden dann bei Schwerpunktsenergien von 14 TeV, Blei-Ionen bei Schwerpunktsenergien von etwa 1150 TeV miteinander zur Kollision gebracht.

Bereits bestehende Beschleuniger wie der PS[2] und der SPS[3] werden die Protonen und Schwerionen vorbeschleunigen, bevor sie in den LHC injiziert werden. Um die Strahlen im LHC auf ihrer Bahn zu halten, werden Magnetfelder von mehr als 8 Tesla, was mehr als dem 100.000fachen des Erdmagnetfeldes entspricht, benötigt. Felder dieser Stärke werden beim LHC durch supraleitende Magnete erzeugt. So wird der LHC mit seiner Länge von 27 km die größte supraleitende Anlage der Welt sein.

Protonen oder Ionen werden in 2 getrennten Strahlen in entgegengesetzter Richtung beschleunigt und an den Wechselwirkungspunkten der 4 Experimente zusammengeführt. Die Experimente ATLAS (**A** **T**oroidal **L**HC **A**pparatu**S**) und CMS (**C**ompact **M**uon **S**olenoid) sind für den Proton-Betrieb ausgelegt. Hauptaufgabe dieser beiden Experimente ist die Untersuchung der Natur der Masse, insbesondere die Erzeugung und die Untersuchung des Higgs-Bosons sowie der Test theoretischer Modelle jenseits des Standardmodells wie z.B.

[1] **L**arge **E**lectron **P**ositron Collider.
[2] **P**roton **S**ynchrotron.
[3] **S**uper **P**roton **S**ynchrotron.

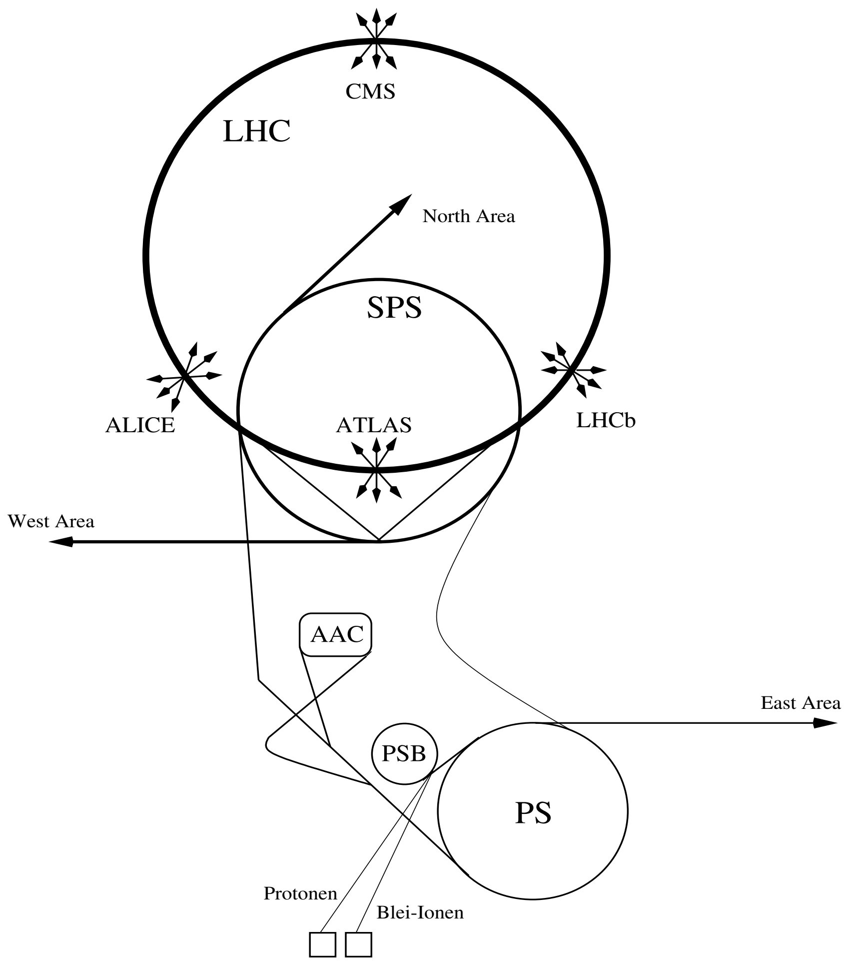


Abbildung 2.1: Schematische Darstellung des LHC-Beschleunigers mit den Vorbeschleunigern PS und SPS und den vier Experimenten ALICE, ATLAS, CMS und LHCb. AAC: Antiproton Accumulator Complex, PSB: Proton Synchrotron Booster. Die Abbildung ist nicht maßstäblich.

der Supersymmetrie. Das LHCb-Experiment sucht nach der CP-Verletzung im B-Meson-System und wird so einen Beitrag zur Frage nach dem Ungleichgewicht zwischen Materie und Anti-Materie im Universum liefern. Das Experiment ALICE ist für den Schwerionen-Modus ausgelegt und soll das Quark-Gluon-Plasma näher untersuchen. Es wird im folgenden Abschnitt detaillierter vorgestellt.

Abbildung 2.1 zeigt schematisch die LHC-Ring-Anlage mit den Vorbeschleunigern PS und SPS sowie den vier Experimenten.

Die technischen Parameter des LHC übertreffen diejenigen bereits existierender Anlagen bei weitem. So wird die Kollisionsenergie für die Konstituenten des Protons, Gluonen und Quarks, 10 mal höher sein als beim Fermilab-Beschleuniger Tevatron in Chicago, für Blei-

Energie pro Ladung	7 TeV/Ladung
Energie pro Nukleon	2.76 TeV/u
Schwerpunktsenergie	1148 TeV
Dipolfeld	8.386 T
Strahldurchmesser am Wechselwirkungspunkt	15 μm
Paketlänge	7.5 cm
Zahl der Pakete	608
Zahl der Ionen pro Paket	$6.3 \cdot 10^7$
Umlauffrequenz	11.2 kHz
Luminosität	$0.85 \cdot 10^{27}$cm^{-2}s^{-1}
Fülldauer	9.8 min
Strahllebensdauer	10 h

Tabelle 2.1: LHC-Parameter in Pb-Pb-Modus.

Ionen wird sie die beim Schwerionenkollider RHIC erreichten Energien sogar um das 30fache übertreffen.

Die Luminosität wird im Schwerionen-Modus 1 Größenordnung höher liegen als etwa bei RHIC, im Proton-Modus wird die Luminosität bestehender Beschleuniger um 2 Größenordnungen übertroffen. Der resultierende sehr große Strahlstrom von mehr als 0.5 A im Proton-Modus ist eine besondere Herausforderung. Zum ersten Mal werden diese hohen Ströme bei so hohen Energien zusammen mit einer so großen supraleitenden Anlage verwendet. Dies stellt besondere Anforderungen an die verwendete Beschleunigertechnik, um z.B. die effiziente Bündelung des Strahls zu gewährleisten.

Einige Parameter des LHC im Schwerionen-Modus sind in Tabelle 2.1 nochmals mit ihren Zahlenwerten aufgeführt.

2.2 Der ALICE - Detektor

Die große Herausforderung in Schwerionen-Experimenten ist die hohe Teilchenzahl, die bei jeder Kollision entstehen. So werden bei ALICE bis zu 20000 Teilchen pro Kollision erwartet. Diese müssen zu einem überwiegenden Teil identifiziert werden, um ein Bild von den Vorgängen während der Kollision zu erhalten.

Der Detektor ALICE (**A** **L**arge **I**on **C**ollider **E**xperiment) [2] ist optimiert für das Studium solcher Schwerionenkollisionen. Er soll die Untersuchung von Materie bei extrem hohen Energiedichten und Temperaturen nahe einem baryochemischen Potential von 0 ermöglichen und so die Untersuchung des Quark-Gluon-Plasmas gewährleisten.[4]

Zu diesem Zweck wird der Detektor aus mehreren Komponenten aufgebaut, die im Zusammenspiel die Anforderungen eines Schwerionen-Experiments bewältigen sollen.

Der Detektor besteht aus drei Hauptkomponenten: dem Zentralbereich mit dem Magneten des LEP-Experiments L3 und verschiedenen Subdetektoren (ITS, TPC, TRD, TOF, PHOS und HMPID), dem Vorwärts-Myon-Spektrometer und den übrigen Vorwärtsdetektoren (ZDC, FMD, CASTOR und PMD).

Diese Hauptkomponenten decken verschiedene Bereiche der Pseudorapidität η[5] ab. So liegt die Akzeptanz im Zentralbereich bei Werten von $|\eta| < 0.9$ über den vollen Azimuth, für das Vorwärts-Myon-Spektrometer gilt $2.5 < \eta < 4$ und für die übrigen Vorwärtsdetektoren muß $\eta > 4$ sein.

Abb. 2.2 zeigt den schematischen Aufbau des Detektors. Insgesamt wird der Detektor etwa 15 m hoch und 25 m lang sein und ein Gewicht von 10000 t haben.

Die einzelnen Komponenten mit ihren jeweiligen Detektoren sollen im folgenden kurz vorgestellt werden.

2.2.1 Der Zentralbereich

Das Inner Tracking System (ITS)

Das Inner Tracking System (ITS) [3] wird aus sechs Siliziumschichten bestehen, die zylindrisch um den Kollisionspunkt herum angeordnet sind. Je nach Schicht sind die verwendeten Techniken Silizium-Pixel-, Silizium-Streifen- oder Silizium-Drift-Detektoren.

Die Aufgaben dieses innersten Detektors sind die Bestimmung des Primärvertex sowie der Sekundärvertices, die zur Rekonstruktion der Zerfälle von Teilchen, die Charm- oder Strange-Quarks enthalten, nötig sind. Weitere Anforderungen sind die Identifikation sowie

[4] Auch im Proton-Proton-Modus des LHC wird ALICE Daten nehmen, allerdings erfordern die Eigenschaften des Detektors hierbei eine verminderte Luminosität. Diese kann z.B. durch Versetzung der Strahlen gegeneinander reduziert werden.
Weiterhin sind für die Erzeugung von Referenzdaten auch Deuterium-Deuterium- und $\alpha - \alpha$-Kollisionen, Kollisionen mit mittelschweren Ionen (wie Ar und O) sowie asymmetrische Kollisionen (p-A) vorgesehen.

[5] Die Pseudorapidität η ist das gängige Maß für die Beschreibung der Winkel θ relativ zum Strahlrohr. Sie ist definiert als $\eta = -\ln(\tan(\theta/2))$.

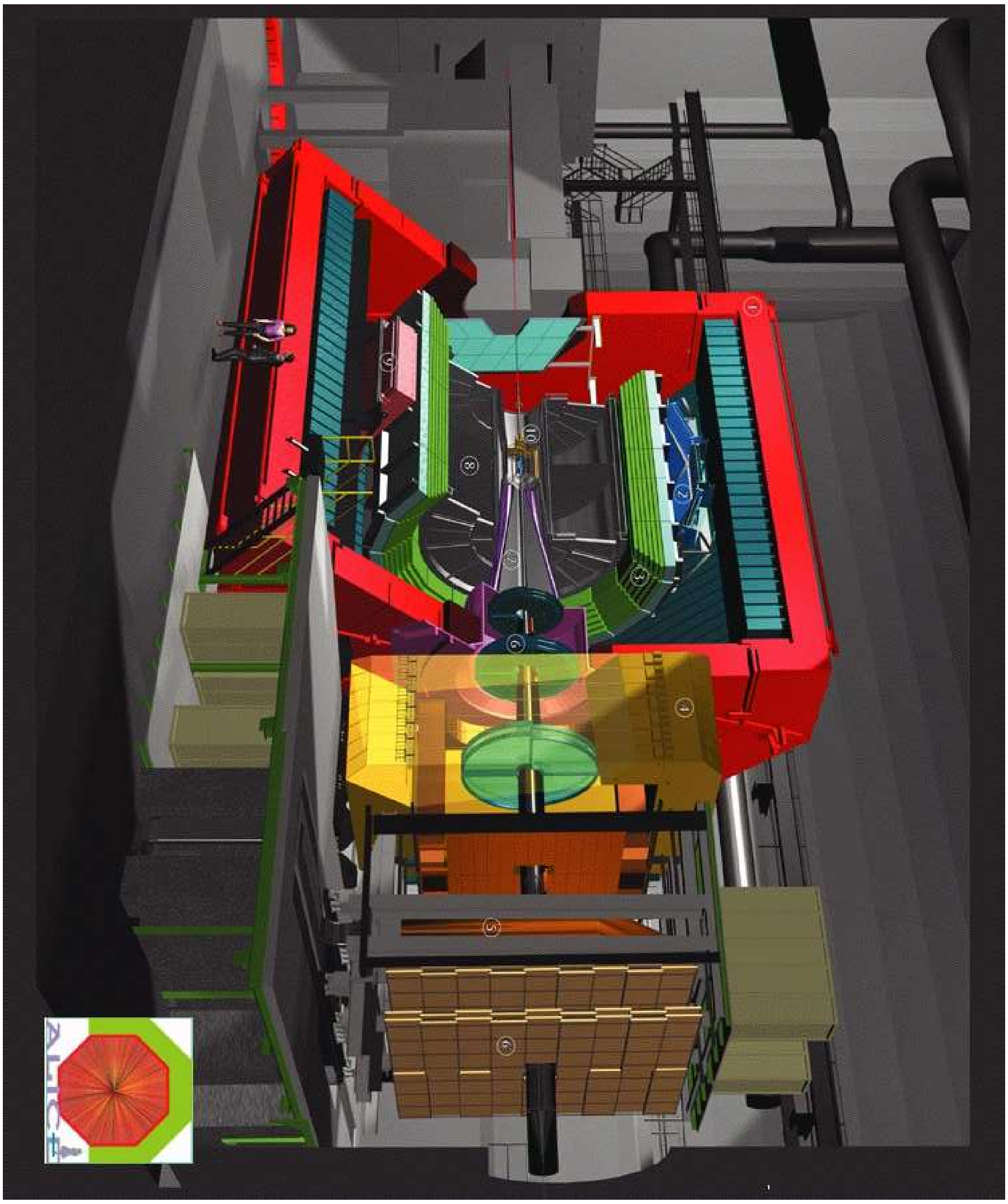


Abbildung 2.2: Aufbau des ALICE-Detektors.
1: L3-Magnet, 2: HMPID, 3: TOF, 4: Dipolmagnet, 5: Myon-Filter, 6: Spur-Kammern, 6’: Trigger-Kammern, 7: Absorber, 8: TPC, 9: PHOS, 10: ITS.

das Auffinden der Spuren von Teilchen mit niedrigem Impuls. Im Zusammenwirken mit der Zeit-Projektions-Kammer soll die Impuls- und Winkelbestimmung verbessert werden.

Der innere Radius (∼3 cm) ist der kleinstmögliche, der mit den Ausmaßen des Strahlrohrs verträglich ist, der äußere (∼ 50 cm) ist durch die gemeinsame Spurerkennung mit der TPC bestimmt. Am äußeren Radius hat der ITS-Detektor eine Länge in z-Richtung von etwa 1 m. Es ergibt sich insgesamt eine aktive Fläche von knapp 7 m^2.

Die Zeit-Projektions-Kammer (TPC)

Die Zeit-Projektions-Kammer (Time Projection Chamber, TPC) [4] ist das Hauptwerkzeug im ALICE-Experiment zur Spurerkennung und zur Teilchenidentifizierung. Da sie den mit Abstand größten Anteil am Datenvolumen hat[6] und daher der Test von Kompressionsverfahren besonders für diesen Detektor relevant ist,[7] soll er hier etwas detaillierter als die übrigen vorgestellt werden, zumal die Komprimierung der Daten dieses Detektors Gegenstand dieser Arbeit ist.

Im Zusammenspiel mit den übrigen Detektoren des Zentralbereiches muß die TPC die Impulsmessung und -auflösung, die Teilchenidentifikation, die Vertexbestimmung und andere physikalische Größen im Akzeptanzbereich $\mid \eta \mid < 0.9$ mit hinreichender Genauigkeit gewährleisten. Der Bereich des Transversalimpulses $P_t < 10$ GeV/c ist hierbei durch das Magnetfeld des verwendeten Magneten vom L3-Experiment bestimmt.

Bei maximal erwarteten Teilchenmultiplizitäten von 8000 geladenen Teilchen pro Rapiditätseinheit werden bis zu 20000 geladene primäre und sekundäre Spuren in der TPC erwartet. Die daraus resultierende Auslastung der TPC, definiert als der Anteil der Pads und Zeitbins, deren ADC Werte über der Nullunterdrückungsschwelle liegen, bewegt sich dabei zwischen 15% am äußeren und 40% am inneren Radius.

Die TPC ist so konstruiert, daß sie die nötigen Anforderungen erfüllen kann: Sie besteht aus zwei konzentrischen Zylindern mit Durchmessern von 1140 bzw. 5560 mm und einer Länge von 5100 mm. Der innere Radius ist bestimmt durch die maximal akzeptable Hit-Dichte, während der äußere Radius durch die minimale Spurlänge, die für eine dE/dx-Auflösung von 10% nötig ist, festgelegt wird. An beiden Enden sind die Zylinder durch die Auslese-Endkappen miteinander verbunden. Parallel zu den Endkappen in der Mitte des Driftvolumens befindet sich die zentrale Hochspannungselektrode. Zwischen ihr und den Endkappen besteht ein elektrisches Feld, in welchem die beim Durchgang geladener Teilchen durch Ionisation entstandenen Elektronen in weniger als 100 μs zu den beiden Endkappen driften. Die Endkappen sind jeweils in 18 Sektoren mit je einer inneren und einer äuße-

---

[6] Die TPC liefert etwa 66 MByte/Ereignis während alle übrigen Detektoren zusammen nur etwa 10 MByte/Ereignis liefern werden.

[7] Auch für die Daten zweier Teil-Detektoren des ITS-Detektors werden z. Zt. Kompressionsverfahren untersucht. Die Signale der beiden Silizium-Drift-Detektoren können als 256x256 Pixel große Bilder mit in 10 Bit codierten Intensitätsstufen interpretiert werden und so werden hieran Bildkompressionsverfahren getestet [1].

ren Kammer aufgeteilt, auf denen Vieldraht-Proportional-Kammern mit Kathoden-Pads als Auslese-Einheiten angebracht sind. Auf diesen werden die von den Elektronen erzeugten Signale registriert und durch die Front-End-Elektronik verarbeitet. Dazu gehören etwa die Vorverstärkung, die Analog-Digital-Konversion oder die Pulsformung. Insgesamt hat die TPC etwa 570000 solcher Pads in drei verschiedenen Größen, die in ungefähr 500 Zeitschritten ausgelesen werden. Die maximale Auslesefrequenz liegt dabei bei 200 Hz.

Eine Diskussion der für die Auflösung dieses Subdetektors relevanten Größen findet sich in Kap. 2.4; auf die produzierten Datenmengen wird in Kap. 3.1 näher eingegangen.

Der Übergangsstrahlungs-Detektor (TRD)

Übergangsstrahlung wird von Teilchen emittiert, wenn sie die Grenze zwischen zwei Medien mit unterschiedlichen Brechungsindizes überqueren. Da die Wahrscheinlichkeit für die Abstrahlung relativ gering ist, werden in Übergangsstrahlungsdetektoren meist viele solcher Übergänge hintereinander verwendet. Der ALICE-TRD [5] unterteilt sich deshalb in 540 Sub-Detektoren: 18 Unterteilungen in azimuthaler Richtung, 5 entlang der Strahlachse und 6 aus Radiator[8] und Driftbereich bestehenden Kammern für jedes dieser 90 Module.

Der Nachweis geschieht über die Primärionisation der geladenen Teilchen, die Identifikation über die Energiedeposition der Photonen.

Bei Teilchen mit Lorentzfaktoren $\gamma > 1000$ liegt die Übergangsstrahlung im weichen Röntgenbereich zwischen 2 und 30 keV. Dies wird zur Elektron-Pion-Trennung bei Impulsen über 0.5 GeV/c genutzt.[9]

Der Elektronenkanal ist besonders wichtig für die Unterscheidung zwischen direkt produzierten J/ψs und solchen, die z.B. beim Zerfall von B-Mesonen entstanden sind (siehe Einleitung). Desweiteren soll die gute Elektronen-Nachweis-Effizienz zum Studium von in Blei-Blei-Kollisionen produzierten leichten und schweren Vektormeson-Resonanzen sowie zur Untersuchung von Open-Charm- und Open-Beauty-Produktion genutzt werden.

Da die Driftzeit nur etwa 2 μs beträgt, wird der TRD als schneller[10] Detektor auch als Trigger eingesetzt werden.

---

[8]Wie der Radiator aufgebaut sein wird, steht zum Zeitpunkt dieser Arbeit noch nicht fest. Untersucht werden als mögliche Konfigurationen die Schichtung von Folien sowie die Anwendung eines schaumartigen Materials, in dem der Übergang dann zwischen dem Schaum selbst und eingeschlossenen Gasblasen stattfinden würde.

[9]γ = E/m $\cong$ P/m; mit m_e=511 keV folgt für γ=1000 ein Impuls für Elektronen von etwa 0.5 GeV/c. Aufgrund der sehr viel größeren Masse emittieren Pionen keine Photonen und können so von den Elektronen unterschieden werden.

[10]Zu den schnellen Detektoren gehören die Trigger-Detektoren wie die Myon-Trigger-Kammern, das ZDC oder der FMD, während zu den langsamen Detektoren die Driftkammern wie das ITS oder die TPC zählen.

Der Flugzeit-Detektor (TOF)

Zur Bestimmung der Teilchenmassen wird ein Flugzeit-Detektor (Time Of Flight, TOF) [6] eingesetzt werden. Während bei Teilchen mit kleinerer Energie die Identifikation über die spezifische Ionisation dE/dx erfolgt, werden Teilchen mit höherer Energie über die Zeit, die sie vom Kollisionspunkt zum Flugzeit-Detektor benötigen, bestimmt. Der ITS-Detektor und die TPC können im Verbund über die spezifische Ionisation dE/dx die K/π - Trennung bis zu Impulsen von 0.5 GeV/c gewährleisten. Ab diesem Bereich bis zu Impulsen von etwa 2.5 GeV/c übernimmt diese Aufgabe der TOF-Detektor.

Dieser Detektor wird aus 160.000 Parallel-Platten-Zähler bestehen, die zylindrisch über etwa 150 Quadratmeter verteilt sein werden. Die erwartete Zeit-Auflösung liegt bei etwa 150 ps.

Das Photon-Spektrometer (PHOS)

Das Photon-Spektrometer (PHOton Spectrometer, PHOS) [7] ist ein elektromagnetisches Kalorimeter aus Blei-Wolfram-Kristallen mit einer hohen Granularität. Es ist optimiert für den Nachweis hochenergetischer Photonen, π^0s und ηs. Die Photonen geben Aufschluß über die anfänglichen Bedingungen in der heißen und dichten Materie, die bei den ultrarelativistischen Kern-Kern-Stößen entsteht. Die Pionen und die ηs hingegen werden unter anderem Signale zur Untersuchung der chiralen Symmetrie liefern.

Der PHOS-Detektor wird unterhalb der übrigen Detektoren des Zentralbereichs installiert sein. Mit seiner etwa 8 m^2 großen Fläche wird er den Bereich $\mid \eta \mid < 0.12$ sowie in azimuthaler Richtung 100° abdecken können.

Der Detektor zur Identifizierung hochenergetischer Teilchen (HMPID)

Der HMPID-Detektor (High Momentum Particle IDentification, HMPID) [8] besteht aus sieben Modulen, die oberhalb des TOF-Detektors nahe am inneren Radius des L3-Magneten angebracht sind, siehe auch Abb. 2.2. Dieser Detektor wurde zur Identifizierung hochenergetischer Teilchen, also Pionen, Protonen und Kaonen mit Impulsen zwischen 1 und 5 GeV/c, entworfen. Er ist als Ring-Imaging-Čerenkovzähler (RICH) konzipiert und kann so aus der Messung des Öffnungswinkels der Čerenkov-Photonen die Geschwindigkeit der Teilchen und bei bekanntem Impuls ihre Identität ermitteln.

Der Magnet

Umgeben werden die Detektoren des Zentralbereichs von dem Magneten des L3-Experiments, das sich zur Zeit an der Stelle im LEP-Tunnel befindet, an der ALICE stehen wird.

Die vorgesehene Feldstärke von 0.2 Tesla ist ein Kompromiß zwischen Impulsauflösung, Akzeptanz kleiner Impulse und der Spurerkennungseffizienz in der TPC. Mit dieser Feldstärke wird die Spurerkennung in der TPC bis zu Impulsen von 0.1 GeV/c sichergestellt. Kleinere Impulse werden, wie bereits erwähnt, vom ITS vermessen.

2.2.2 Das Myon-Spektrometer

Das Myon-Spektrometer [9] wird Myon-Paare nachweisen, die aus Quarkonia-Zerfällen stammen. Die Untersuchung von Quarkonia-Zuständen ist ein wichtiges Hilfsmittel, um die frühsten und heißesten Zustände einer Kern-Kern-Kollision auszumessen.

Das Spektrometer besteht aus mehreren Komponenten: Zur Unterdrückung von Hadronen und Photonen, die direkt vom Wechselwirkungspunkt kommen, befindet sich noch innerhalb des Magneten der Front-Absorber. Über seine gesamte Länge hin ist das Spektrometer außerdem durch einen das Strahlrohr umgebenden Absorber geschützt. Hinter dem Front-Absorber sind teils vor, teils in und zum Teil hinter einem Dipolmagneten die Spurkammern angebracht. Hinter einem zweiten Absorber, der zusammen mit dem ersten Myonen mit Impulsen bis zu 4 GeV/c stoppt, liegen die Trigger-Kammern. Da der Transversalimpuls von emittierten Myonen aus J/ψ - Zerfällen oder Υ- Resonanzen meist größer ist als der der Myonen aus Hintergrund-Zerfällen, wie denen des π oder des K, können die Trigger-Kammern durch einen Schnitt auf den Impuls die interessanten Ereignisse selektieren.

2.2.3 Die übrigen Vorwärtsdetektoren

Der Vorwärts-Multiplizitäts-Detektor (FMD)

Um die Verteilung der Multiplizität außerhalb des Akzeptanzbereiches der Detektoren des Zentralbereiches zu bestimmen, wird ein spezieller Vorwärts-Multiplizitäts-Detektor (Forward Multiplicity Detector, FMD) [12] eingesetzt. Weitere wichtige Größen wie die z-Position des Vertex oder der Ereignis-Zeitpunkt T_0 - die Auflösung liegt hier bei weniger als 100 ps - können ebenfalls mit diesem Detektor genau vermessen werden.

Bestehend aus 7 Scheiben, die in Abständen von etwa 42 bis 230 cm symmetrisch[11] um den Wechselwirkungspunkt das Strahlrohr umgeben, wird dieser Detektor den Rapiditätsbereich zwischen $\mid \eta \mid = 1.6$ und $\mid \eta \mid = 4.7$ abdecken.

Die hier verwendete Technik werden Mikro-Kanal-Platten (Micro-Channel-Plate, MCP) sein, schmale Bleiglasplatten mit vielen Poren kleinen Durchmessers, die sich innerhalb von Vakuum-Containern befinden. Die von durchfliegenden Teilchen produzierte Ladung wird in den Bleiglas-Kanälen verstärkt und dann 0.5 mm hinter den MCPs registriert.

[11] Bis auf die siebte Scheibe, die dem Myon-Spektrometer gegenüber sitzt.

Der Photon-Multiplizitäts-Detektor (PMD)

Gegenüber dem Myon-Spektrometer wird sich ein Photon-Multiplizitäts-Detektor (Photon Multiplicity Detector, PMD) [13] befinden. Knapp 6 m vom Wechselwirkungspunkt entfernt deckt dieser Detektor den Bereich von $1.8 \leq \eta \leq 2.6$ über den vollen Azimuth ab.

Photonen, die ein Konverter-Material durchqueren, initiieren einen elektromagnetischen Schauer, der mittels eines Schauerdetektors nachgewiesen werden kann. Geladene Teilchen hingegen werden durch spezielle Veto-Detektoren vor dem Konverter registriert. Die vorgesehene Technik, sowohl für den Veto- als auch für den Schauer-Detektor, ist eine Anordnung aus hexagonalen Proportional-Kammern.

Die Messung der Multipizität und der räumlichen Verteilung der Photonen können zur Charakterisierung eines Ereignisses und zur Bestimmung der transversalen Energie genutzt werden.

Das CASTOR - Kalorimeter

Für das Studium von Phänomenen, die in und durch die Bedingungen bei sehr hohem baryochemischen Potential in der extremen Vorwärtsrichtung ($5.6 \leq \eta \leq 7.2$) produziert werden, wird der CASTOR-Detektor (Centauro And STrange Object Research, CASTOR) [10] entwickelt. Die physikalische Motivation sind Experimente mit kosmischer Strahlung, die eine Reihe von bisher noch unverstandenen Ereignissen nachweisen konnten. Diese Ereignisse lassen sich eventuell bei LHC-Energien unter genau definierten Bedingungen reproduzieren und mittels des CASTOR-Detektors studieren. Als Beispiel seien hier nur die sogenannten 'Centauro'-Ereignisse mit sehr kleiner Multiplizität, starker Unterdrückung der elektromagnetischen Komponente und hohem mittlerem Transversalimpuls $<p_t>$ erwähnt.

Das CASTOR-Kalorimeter unterteilt sich in einen elektromagnetischen und einen hadronischen Teil, die jeweils longitudinal in die aktiven Silizium-Schichten und Wolfram-Absorbern segmentiert sind. Diese longitudinale Schichtung gewährleistet die genaue Messung der entstehenden Kaskaden. Das erzeugte Čerenkov-Licht wird dann über Lichtleiter aus Plastik ausgelesen.

Der Detektor wird sich in 10,5 m Entfernung vom Wechselwirkungspunkt befinden und bei einer Länge von 1 m einen äußeren Radius von 25 cm haben.

Das Null-Grad-Kalorimeter (ZDC)

Zum Nachweis der Nukleonen, die nicht an der Kollision teilgenommen haben, dienen die Null-Grad-Kalorimeter (Zero Degree Calorimeter, ZDC) [11]. Aus der transversalen Energie E_{Trans} dieser Nukleonen lassen sich Rückschlüsse auf die Zentralität der Kollision ziehen. Die Zentralität ist ein wichtiges Maß für die in einer Kollision erzielte Energiedichte und deren genaue Bestimmung essentiell für das genaue Ausmessen von Schwellwerten, die die

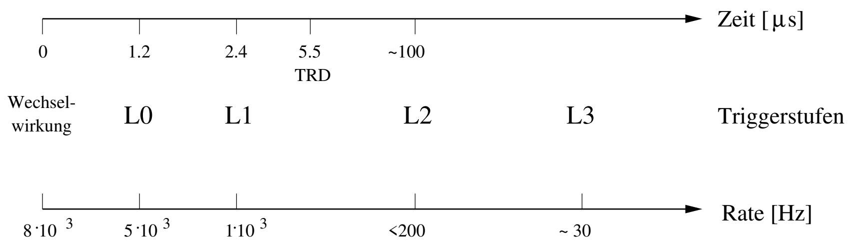


Abbildung 2.3: Latenzen und Raten der einzelnen Triggerstufen. Für den Level-3 Trigger gibt es keine feste Latenz, die Ausleserate wird durch die zur Verfügung stehende Bandbreite festgelegt.

Formation eines Quark-Gluon-Plasmas charakterisieren.

Die Kalorimeter werden etwa 115 m vom Wechselwirkungspunkt entfernt stehen. Der Nachweis der Nukleonen geschieht über abgestrahltes Čerenkov-Licht in Quartz-Glas.

2.3 Das Trigger-System

Die mittlere Ereignisrate im Pb-Pb-Modus am LHC wird etwa 10^4 Kollisionen pro Sekunde sein, wovon etwa 100 Ereignisse in der Sekunde wegen ihres kleinen Stoßparameters zu den interessantesten mit maximaler Teilchenproduktion zählen.

Das Trigger-System von ALICE ist insofern auf diese Gegebenheiten abgestimmt, als daß es den Schwerpunkt auf die Bestimmung der Zentralität eines Ereignisses legt. Da die verwendeten Detektoren unterschiedliche Totzeiten haben, wird ein mehrstufiges System zum Einsatz kommen, das sich hierarchisch in vier Ebenen gliedert.

Im Level-0 Trigger werden die Daten des FMD verwendet und nach der festen Zeit von 1.2 μs entschieden, ob eine Kollision stattgefunden hat, die die Kriterien dieser Trigger-Stufe erfüllt. Im einzelnen muß hier die Multiplizität über einer Schwelle liegen, der Wechselwirkungspunkt nahe genug am nominellen Ort sein und die Vorwärts-Rückwärts-Verteilung der Spuren muß konsistent sein mit derjenigen einer Strahl-Strahl-Wechselwirkung. Der Level-0 Trigger löst dann gegebenenfalls die Auslese-Elektronik verschiedener Sub-Detektoren aus. Die Raten, bei denen der Level-0 Trigger arbeitet, liegen zwischen $5{\cdot}10^4$ Hz (für den Proton-Proton-Betrieb) und $5{\cdot}10^3$ Hz (für den Blei-Blei-Betrieb). Integriert in den Level-0 Trigger ist der TRD, der nach Elektronen mit hohem Transversalimpuls suchen wird und gegebenenfalls die Auslese der TPC auslösen kann. Das Zeitlimit für den TRD bis zur Entscheidung beträgt 5.5 μs.

In die Entscheidung der nächsten Stufe, des Level-1 Triggers, halten zusätzlich die Informationen aus den Null-Grad-Kalorimetern und den Myon-Trigger-Kammern Einzug. Auf dieser Ebene werden so strengere Kriterien an die Zentralität (z.B. aus E_{Trans}, siehe den Abschnitt über das ZDC) und die Multiplizität gelegt, sowie Ereignisse mit zwei Myonen, deren Impulse oberhalb einer Schwelle liegen und die - wie oben bereits ausgeführt - besonders interessant für das Studium von Quarkonia-Zuständen sind, ausgewählt. Die Entscheidung des Level-1 Triggers fällt nach 2.4 μs und führt zu Ereignisraten von etwa 1000 Hz. Alle Sub-Detektoren, deren Auslese-Elektronik nicht vom Level-0 Trigger ausgelöst wurde, erhalten nun vom Level-1 Trigger ein entsprechendes Signal.

Während der Driftzeit der Elektronen in der TPC von $\sim 100\ \mu s$ können die Entscheidungen des Level-2 Triggers getroffen werden. Hier werden Daten von den Trigger-Detektoren hinzugenommen, die nicht Bestandteil der ersten beiden Stufen waren, wie z.B. diejenigen des PHOS-Detektors. Im Level-2 Trigger werden dann detailliertere Analysen vorgenommen, wie etwa ein Schnitt auf die Masse eines Di-Myon-Systems (Suche nach J/ψ) oder die Untersuchung der Cluster im FMD. Die Latenz des Level-2 Triggers ist nicht festgelegt, da die verschiedenen Algorithmen unterschiedlich viel Zeit benötigen werden.

Die Daten des schnellen TRD werden zusammen mit denen der TPC direkt an die Prozessoren des Level-3 Triggers gesendet. Die Handhabung der TPC-Daten ($\sim$ 13 GB/s) in Echtzeit geschieht durch symmetrische Parallelverarbeitung in der Level-3 Prozessor-Farm, die aus etwa 500 PCs bestehen wird, die durch ein Hochgeschwindigkeits-Netzwerk mit hoher Bandbreite und geringer Latenz verbunden sein werden. Hier werden Aufgaben wie die Online-Spurerkennung oder die Datenkompression durchgeführt. Die maximale Arbeitsrate des Level-3 Triggers liegt bei 200 Hz.

Das Trigger-System von ALICE ermöglicht durch die Verwendung unterschiedlicher Trigger die Auslese ausgewählter Detektoren. So werden für spezielle Untersuchungen nur der Myon-Arm und einige der schnellen Detektoren ausgelesen, die langsamere TPC hingegen nicht.

Nachdem die Trigger-Entscheidungen gefallen sind, sorgt das Datenakquisitionssystem für die Auslese, Verarbeitung und den Transport der Daten zum Archivierungssystem.

Abb. 2.3 zeigt ein Schema der Latenzen und der Raten der einzelnen Triggerstufen.

2.4 Orts- und Impulsauflösung der TPC

Da in dieser Arbeit auch verlustbehaftete Kompressionsverfahren für die Daten der TPC untersucht werden, soll in diesem Abschnitt genauer erläutert werden, welche Größen für die Orts- und Impulsauflösung dieses Detektors von Relevanz sind.

Durchquert ein geladenes Teilchen ein Magnetfeld, so wird seine Bahn aufgrund der Lorentzkraft gekrümmt sein, wenn sein Geschwindigkeitsvektor eine Komponente senkrecht zum B-Feld hat. Aus der Krümmung ρ der Bahn und der bekannten Magnetfeldstärke B kann die zum B-Feld senkrechte Komponente des Impulses aus

$$p_t = \frac{0.3B}{\rho} \tag{2.1}$$

bestimmt werden.

Die Ortsauflösung in azimuthaler Richtung $r\delta\varphi$ hängt mit der relativen p_t-Auflösung über

$$\frac{\triangle p_t}{p_t} = \frac{p_t r\delta\varphi}{0.3BL^2}\sqrt{\frac{720}{N+4}} \tag{2.2}$$

zusammen, siehe [25]. Hierin sind L die sichtbare Gesamtlänge der Spur und N die Zahl der zugeordneten Raumpunkte. L und B sind durch die Parameter der ALICE TPC und des Magneten gegeben. N und $r\delta\varphi$ sind Parameter der Auslese-Einheiten, also der Vieldraht-Proportional-Kammern mit den Kathoden-Pads.

Verschiedene Faktoren gehen hier in die Ortsauflösung in azimuthaler Richtung ein [4]:

- die Verbreiterung der durch Ionisation erzeugten Elektronenwolken während der Drift (diese Verbreiterung wird durch das zur Drift parallele B-Feld reduziert, indem es die Elektronen auf schraubenförmige Bahnen um die Feldlinien zwingt),
- Fluktuationen in der Gasverstärkung einzelner Elektronen,
- die Produktion von Sekundärelektronen (sog. Landau-Fluktuationen),
- die Anzahl der Elektronen, die zur Messung eines Raumpunktes beitragen,
- das Signal-zu-Rausch-Verhältnis und
- Effekte durch die Digitalisierung und die Anlegung einer Schwelle.

Einige dieser Effekte hängen eng mit der Geometrie der Auslese-Pads zusammen, so z.B. die Anzahl der Elektronen, die zur Bestimmung eines Raumpunktes beitragen. Aus diesem Grunde wurden verschiedene Pad-Geometrien bezüglich ihrer Ortsauflösung getestet.

Für die TPC ergeben sich mit der gewählten Pad-Geometrie bei gut getrennten Spuren minimal ionisierender Teilchen in azimuthaler Richtung Auflösungen von 1150 μm (innere Kammer) bzw. 800-850 μm (äußere Kammer) sowie in Drift-Richtung 1250 μm (innen) bzw. 1100 μm (außen). Die erwartete relative Transversalimpuls-Auflösung liegt bei 2.1%.

Kapitel 3

Datenkompression

Ich habe diesen Brief länger als gewöhnlich gemacht,
da mir die Zeit fehlt, ihn zu kürzen.
Blaise Pascal (1623-1662)

3.1 Motivation für Datenkompression bei ALICE

Da die TPC etwa 570000 Pads hat und die Anzahl der Zeit-Bins pro Pad etwa 500 beträgt, erhält man unter der Annahme einer nicht-linearen Kompression der 10 Bit Daten in 8 Bit Werte eine Roh-Event-Größe von 285 MB.

Unter Berücksichtigung der Tatsache, daß nicht alle Teilchenbahnen physikalisch gleichermaßer interessant sind, weil sie z.B. nicht aus einem harten Streuprozeß stammen und so unter kleinem Winkel zum Strahlrohr die TPC durchqueren, wird aus den Daten ein 45-Grad-Konus herausgeschnitten, was eine Reduktion der Datenmenge um etwa 50 % bewirkt. Durch Nullunterdrückung, welche den Abzug eines Pedestals (in den verwendeten Daten die ADC Werte 1 und 2), das eindimensionale Cluster-Finden in der Zeitrichtung und die Lauflängencodierung von Nullintervallen beinhaltet, kann die Größe eines Events auf etwa 66 MB ± 15 % reduziert werden.

Betrachtet man nun die maximale Auslesefrequenz der TPC von 200 Hz, so erhält man einen Datenstrom von etwa 13 GB pro Sekunde.

Zwei Punkte machen die Notwendigkeit einer Datenkompression deutlich: zum ersten das Problem, solch eine Menge von Daten über ein Netzwerk zu verschicken. Zur Zeit gibt es noch kein lokales Netzwerk, das solch eine Bandbreite liefern kann.

Zum zweiten treten natürlich enorme Kosten zur Speicherung der anfallenden Daten auf. Da mit einer Datenmenge von 2 PB pro Jahr gerechnet werden muß, erhält man mit einem Preis von etwa 2 CHF/GB Kosten zur Speicherung auf Band von etwa 4 Mio. CHF pro Jahr.

Beide Aspekte verdeutlichen die Notwendigkeit der Datenkompression.

Die Verwendung eines Kompressionsverfahrens auf die Daten bringt aber auch Nachteile mit sich: Der wahlfreie Zugriff, um zum z.B. detaillierte Analysen an einzelnen Spuren durchzuführen, für die die zu der Spur gehörenden Cluster[1] oder deren Pads von Bedeutung sind, ist dann nur noch nach einer Decodierung möglich. So kann z.B. nicht direkt auf die einzelnen ADC Werte eines bestimmten Pads in einer beliebigen Pad-Reihe zugegriffen werden, ohne wesentlich größere Einheiten, wie z.B. ganze Sektoren oder Kammern, zu decodieren. Für die genauere Untersuchung einzelner Spuren müßten eventuell mehrere Sektoren decodiert werden, wovon dann allerdings nur ein sehr geringer Bruchteil an Information verwendet wird. Dieser Nachteil kann jedoch durch geschicktes Wählen der zu codierenden Einheiten und durch zusätzlich abgespeicherte Informationen, in denen z.B. der Abstand bis zur nächsten Einheit steht, zumindest teilweise relativiert werden.

3.2 Allgemeines

Die Datenkompression spielt eine zentrale Rolle in der Informationstheorie, die als Teilzweig der Mathematik ihren Anfang in den späten 40er Jahren des vorigen Jahrhunderts mit den Arbeiten von Claude Shannon in den Bell Laboratorien gefunden hat.

In der Informationstheorie wird der Gehalt einer Nachricht über den Begriff der Entropie, der der Thermodynamik entliehen ist und dort eine ähnliche Bedeutung hat, bestimmt. Je größer die Entropie einer Nachricht ist, desto größer ist ihr Informationsgehalt. Die Entropie eines einzelnen Zeichens i wird über seine Auftrittswahrscheinlichkeit p_i bestimmt:

$$H = - p_i \, log_2 \, p_i \tag{3.1}$$

Da die Entropie eine additive Größe ist, wird die Entropie einer Zeichenkette aus der Summe über die Entropien der einzelnen Zeichen berechnet.

Ziel der Datenkompression ist nun die Minimierung der zur Übertragung oder Speicherung nötigen Anzahl von Bits. Die Möglichkeit der Kompression liegt in der Differenz zwischen der Zahl der Bits zur Darstellung eines Zeichens und seiner Informationsentropie. Somit gibt es eine untere Schranke, unter die hinaus die Daten nicht ohne Informationsverlust komprimiert werden können.

Man unterteilt daher die Datenkompression in zwei große Familien: die verlustlosen (lossless) und die verlustbehafteten (lossy) Verfahren, wobei letztere unter dem Zugeständnis eines Verlustes an Rekonstruktionsgenauigkeit erhebliche höhere Kompressionsraten erreichen können.

[1] Wenn es nicht explizit anders angegeben ist, bezeichnet ein *Cluster* stets eine Menge zusammenhängender ADC Werte in den Pad-Reihen-Zeit-Ebenen, die über der Nullunterdrückungsschwelle liegen.

Verlustlose Algorithmen werden vor allem dort eingesetzt, wo der Verlust eines Bits schwerwiegende Folgen haben könnte, wie bei der Kompression von Texten (z.B. Quelltexte) oder Programmen. Der weiter unten erklärte Algorithmus von Lempel und Ziv, der z.B. in den UNIX-Programmen *gzip* oder *compress* verwendet wird, ist ein Beispiel für ein solches verlustloses Verfahren.

Verlustbehaftete Techniken hingegen finden vorrangig Verwendung bei der Kompression von Bildern oder Tönen, bei denen der in Kauf genommene Verlust an Information so bestimmt wird, daß er für den Benutzer nicht oder kaum zu erkennen und somit irrelevant ist. Ein Beispiel für ein lossy Verfahren ist der bekannte JPEG-Algorithmus. Mit ihm können Bilder auf bis zu unter 10% der ursprünglichen Größe reduziert werden, wobei nur ein geringer Verlust in der Auflösung in Kauf zu nehmen ist.

Eine weitere wichtige Unterscheidung wird zwischen statistischen und Wörterbuch - basierten Verfahren gemacht. Statistische Verfahren weisen einzelnen Zeichen oder Gruppen von Zeichen Symbole zu, wobei die häufiger auftretenden Zeichen die kürzeren Symbole erhalten. Die Zuordnung geschieht über eine Wahrscheinlichkeitstabelle. Diese wird entweder vorher mit Trainingsdaten oder direkt während der Codierung erstellt (adaptive Verfahren). Die Huffman-Codierung ist wohl der berühmteste Vertreter dieser Gattung.

Wörterbuch-basierte (dictionary-based) Verfahren hingegen lesen Ketten von Zeichen ein und benutzen ein Wörterbuch zur Codierung. Das Wörterbuch besteht aus vorher eingelesenen Zeichenketten und kann statisch oder dynamisch sein.

Als Ausgabe wird z.B. ein Index auf den entsprechenden Eintrag zurückgeliefert (ähnlich wie in einer wissenschaftlichen Arbeit bei einem Zitat nicht die gesamte Quelle, sondern nur der Verweis auf den Eintrag im Literaturverzeichnis angegeben wird).

Um die Vielfalt der verschiedenen Kompressionsverfahren anzudeuten, sollen im folgenden einige Algorithmen kurz vorgestellt werden.

3.3 Elementare Kompressionsalgorithmen

3.3.1 Lauflängencodierung

Der wohl intuitiv einfachste verlustlose Kompressionsalgorithmus ist die Lauflängencodierung (Run Length Encoding, RLE). Hierbei werden Zeichen durch sich selbst und die Anzahl des Auftretens hintereinander codiert, d.h. die Zeichenkette

0 0 0 0 0 1 1 1 1 1 1 wird zu

0 5 1 6 codiert.

Dieser Algorithmus führt nicht zwangläufig zu einer Kompression, sondern kann sogar zu einer Vergrößerung des Datenvolumens führen, wie folgendes Beispiel zeigt:

1 2 3 4 5 6 7 8 9 0 wird zu
1 1 2 1 3 1 4 1 5 1 6 1 7 1 8 1 9 1 0 1 codiert.

Hier sieht man exemplarisch, daß die Verwendbarkeit und Effizienz der Algorithmen entscheidend von der Struktur der Daten abhängen kann.

3.3.2 Shannon-Fano-Codierung

Der unter Shannon-Fano-Codierung bekannte Algorithmus gehört zur Gruppe der statistischen Verfahren und wurde nahezu gleichzeitig von Claude Shannon und R.M. Fano Ende der 40er Jahre des vorigen Jahrhunderts entwickelt ([20], [37]). Er funktioniert nach folgendem einfachen Schema: zu einer gegebenen Menge von Symbolen wird die Häufigkeit des Auftretens einzelner Zeichen bestimmt. Dann werden die Zeichen gemäß ihrer Häufigkeit sortiert und die entstandene Liste so in zwei Hälften geteilt, daß die Gesamtwahrscheinlichkeit des Auftretens von Zeichen in beiden Teilen möglichst gleich groß ist. Der ersten Hälfte wird eine 0, der zweiten eine 1 zugewiesen. Dann wird das Verfahren so lange rekursiv mit den beiden Unterhälften fortgesetzt, bis in den Listen nur noch einzelne Zeichen als Blätter des Baumes stehen. Der Code für ein Zeichen ergibt sich dann als die Aneinanderreihung der 0 und 1 in den Knoten beim Durchlaufen des Baumes von der Wurzel zum entsprechenden Blatt.

3.3.3 Huffman-Codierung

Das Verfahren von Shannon-Fano wurde schon bald nach seiner Einführung in seiner Effizienz durch den Algorithmus von Huffman übertroffen [24]. Bei diesem Algorithmus wird im Gegensatz zum vorigen der Binärbaum von unten nach oben aufgebaut: Die Häufigkeit des Auftretens einzelner Zeichen wird ermittelt und jeweils die zwei Knoten mit dem geringsten Gewicht zu einem zusammengefaßt, der als Gewicht die Summe der beiden Ausgangsknoten erhält. Dem Ausgangsknoten mit dem größeren Gewicht wird die 0, dem anderen die 1 zugewiesen. Dieses Verfahren wird wiederholt, bis nur noch zwei Knoten übrig sind, die dann zur Wurzel des Baumes zusammengefaßt werden. Abb. 3.1 zeigt die Erstellung eines solchen Baums für den Ausdruck “Shannon-Fano”. Gemäß ihrer Auftrittswahrscheinlichkeit erhalten die einzelnen Zeichen unterschiedlich lange Codes. Da bei der Huffman-Codierung kein Code das Präfix eines anderen Codes ist, kann die Decodierung als Datenstrom ohne Probleme durchgeführt werden.

Die Codierung mittels dieses Verfahrens ist hinsichtlich des Rechenaufwands sehr günstig: Nach dem Erstellen des Huffman-Baumes besteht sie nur noch aus dem Ablesen des zu einem Zeichen gehörenden Symbol aus einer Tabelle. Somit kann die Codierung also sehr schnell erfolgen. Die Decodierung hingegen geschieht durch das Durchlaufen des aufgebauten

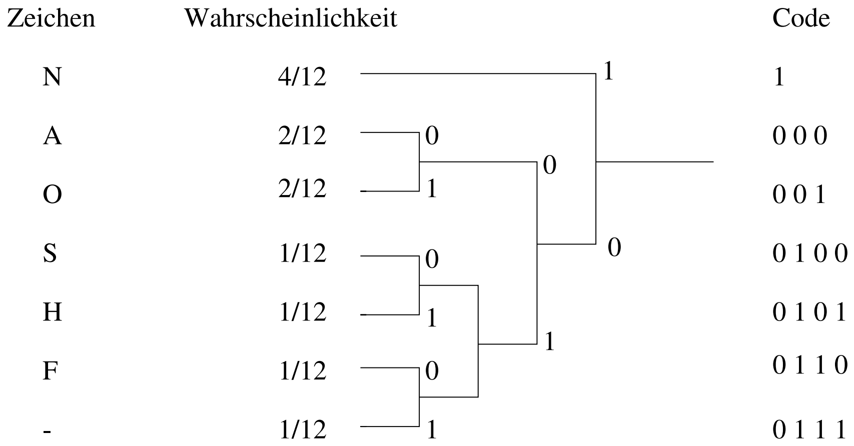


Abbildung 3.1: Konstruktion eines Huffman-Baums für den Term “Shannon-Fano”.

Binärbaumes und dauert - abhängig von der Art der Daten und somit der Struktur des Baumes - entsprechend länger.

Da der größere Zeitaufwand für die Decodierung für das Experiment kein wesentliches Gewicht hat, weil nur während der Datennahme, jedoch nicht bei der späteren Analyse, die verfügbare Zeit, die Daten zu bearbeiten, begrenzt ist, erweist sich die Huffman-Codierung aufgrund ihrer guten Kompressionseigenschaften als ein Verfahren, das für die Kompression der Daten in Frage kommt.

Dieses Verfahren kommt in dieser Arbeit sowohl allein als auch im Zusammenwirken mit der Vektorquantisierung zur Anwendung.

3.3.4 Arithmetische Codierung

Während bei der Huffman oder Shannon-Fano-Codierung jedem Zeichen eine ganze Zahl von Bits zugeordnet wird, kann mit der Arithmetischen Codierung diese offensichtliche Schwäche (denn nicht jedes Symbol hat als Wahrscheinlichkeit eine negative Potenz von 2) umgangen werden. Jedem Zeichen des Eingabe-Stroms wird gemäß seiner Wahrscheinlichkeit ein Subintervall im Bereich [0,1[zugeordnet. Wie der Eingabe-Strom aus der Konkatenation von Zeichen entsteht, so wird die Ausgabe durch ein immer kleiner werdendes Intervall zwischen 0 und 1 definiert, so daß am Ende eine Gleitkommazahl die gesamte Nachricht codiert. Mit dieser Methode erreicht man nahezu die maximale Informationsentropie.

Trotz der meist etwas besseren Kompressionseigenschaften im Vergleich zur Huffman-Codierung wurde auf eine nähergehende Untersuchung der Arithmetischen Codierung verzichtet, da die Anzahl der Operationen pro zu codierenden Zeichen hier erheblich höher ist. So muß für jedes Zeichen bei der Codierung das entsprechende Intervall berechnet werden, was mehrere Operationen impliziert und somit signifikant länger dauert als die Codierung mittels Huffman.

Zum Vergleich der Laufzeiten der beiden Algorithmen werden die Implementationen aus [33] mit den simulierten Daten, die auch die Grundlage für den Test der anderen Verfahren bilden, getestet. Die Huffman-Codierung ist hierbei etwa doppelt so schnell, bei einem Unterschied in der Kompressionsrate von weniger als 0.5%.[2]

3.3.5 LZ77, LZ78 und LZW

Alle bisher genannten Verfahren codieren einzelne Zeichen. Im allgemeinen erreichen Methoden, die ganze Zeichen*ketten* codieren, bessere Ergebnisse, da die Auftrittswahrscheinlichkeit von Zeichenketten stärker variieren als diejenige einzelner Zeichen.

Der unter dem Namen LZ77 bekannte Algorithmus, benannt nach den Entwicklern A. Lempel und J. Ziv und dem Erscheinungsjahr 1977 [41], ist ein Wörterbuch-basierter Algorithmus. Die Grundidee ist hierbei, einen Teil der vorher eingelesenen Zeichen als Wörterbuch zu verwenden. Der Encoder erhält ein Fenster im Eingabe-Datenstrom und verschiebt das Fenster von links nach rechts, nachdem eine Zeichenkette (oder auch nur ein einzelnes Zeichen) codiert wurde ("Sliding-Window-Technik"). Dieses Fenster wird in zwei Teile unterteilt: in den "Search Buffer" mit bereits codierten Zeichen und den "Lookahead Buffer" mit den als nächstes zu codierenden Zeichen.

Bei der Codierung wird der Search-Buffer rückwärts nach dem zu codierenden Zeichen und möglichst vielen mit den nächsten Symbolen übereinstimmenden Zeichen durchsucht. Als Ausgabe erhält man einen Code, der aus drei Teilen besteht: einem Abstand (Offset), einer Länge und dem nächsten Symbol. Der Abstand enthält die Entfernung vom Anfang des Search-Buffers, und die Länge ist die Anzahl der Zeichen der gefundenen identischen Zeichenkette. Abb. 3.2 soll dieses Vorgehen verdeutlichen. Offensichtlich funktioniert dieser Algorithmus am besten, wenn gleiche Muster in den Eingabe-Daten nicht zu weit voneinander entfernt auftreten, da sie sonst aus dem Wörterbuch herausfallen. Der LZ77-Algorithmus wird zusammen mit einem statischen Huffman-Verfahren z.B. in dem Programm *gzip* verwendet.

Beim LZ78-Verfahren gibt es ein Wörterbuch aus vorher eingegebenen Zeichenketten, mit dem die eingelesenen Zeichenketten verglichen werden. Als Ausgabe erhält man den Index der entsprechenden Zeichenkette sowie das erste Zeichen, das die Gleichheit verletzt. Die eingelesene Zeichenkette wird mitsamt diesem Zeichen als neuer Eintrag dem Wörterbuch hinzugefügt. Somit wird nie etwas aus dem Wörterbuch gelöscht, was zum einen ein Vorteil gegenüber LZ77 ist, zum anderen aber das Problem des wachsenden Speicherbedarfs mit sich bringt.

[2]Die im Appendix A von [36] angegebenen Tests von verschiedenen Verfahren an ausführbaren, Graphik- und Textdateien zeigen bei Unterschieden in der gemittelten Kompressionsrate von etwa 2% einen etwa dreimal so großen Zeitbedarf bei der Arithmetischen Codierung wie bei der Huffman-Codierung.

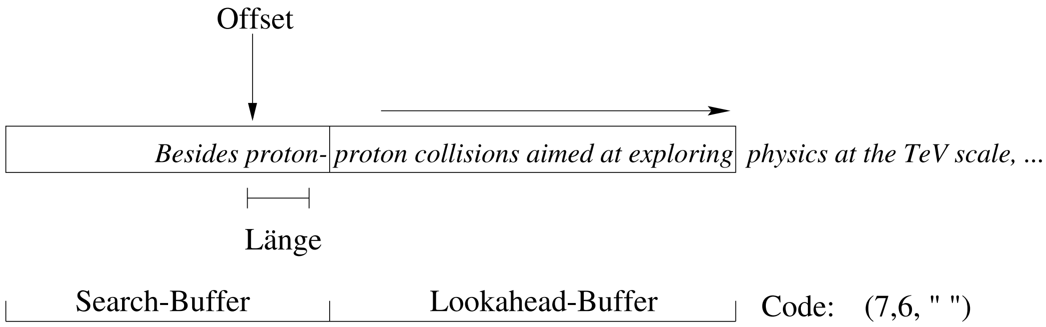


Abbildung 3.2: Vorgehen beim LZ77-Algorithmus am Beispiel eines Ausschnittes aus dem ALICE Technical Proposal. Ebenfalls angegeben ist der erzeugte Code für den dargestellten Schritt.

Eine Variante des LZ78-Algorithmus ist das LZW-Verfahren [40]. Hier liest der Encoder die Eingabe Zeichen für Zeichen ein und fügt sie zu einer Zeichenkette zusammen. Nach jedem angefügten Symbol wird das Wörterbuch nach der Zeichenkette durchsucht. Hat die Zeichenkette einen Eintrag im Wörterbuch, wird erneut ein Zeichen eingelesen und angefügt. Ist kein Eintrag vorhanden, wird der Wörterbuchindex für die bis dahin zusammengefügte Zeichenfolge ohne das zuletzt angehängte Zeichen als Code ausgegeben. Dann wird ein neuer Eintrag im Wörterbuch mit der Zeichenfolge inklusive dem zuletzt eingelesenen Zeichen angelegt und eine neue Zeichenfolge mit disem Zeichen initialisiert. Der ausgegebene Code ist also im Vergleich zum LZ78-Verfahren um einen weiteren Teil verkürzt worden.

Das V.42bis Protokoll für Modems zum Beispiel benutzt eine Variante vom LZW.

Neben den hier vorgestellten gibt es noch eine ganze Reihe anderer Varianten der Lempel-Ziv-Codierung. Für eine Vertiefung sei etwa auf [36] verwiesen.

3.4 Vektorquantisierung

Ein Vektorquantisierer ähnelt dem bereits vorgestellten LZW-Algorithmus. Auch hier wird zur Codierung in einem Wörterbuch nach einem passenden Eintrag gesucht, mit dem der zu codierende Vektor identifiziert werden kann. Ist ein entsprechender Eintrag gefunden worden, wird sein Index als Code abgespeichert. Konnte kein Eintrag gefunden werden, kann z.B. der Vektor im Original oder der Eintrag im Wörterbuch mit der geringsten Abweichung von diesem Vektor bezüglich eines bestimmten Abstandsmaßes als Code verwendet werden. Letzteres entspricht einer verlustbehafteten Komprimierung wie sie in abgewandelter Form in dieser Arbeit zum Einsatz kommen soll. Das genaue Vorgehen wird in Abschnitt 4.3.2 erläutert.

Mathematisch betrachtet ist ein Vektorquantisierer eine Abbildung VQ vom k-dimensionalen Raum in einen Unterraum CB, der N k-dimensionale Elemente enthält:

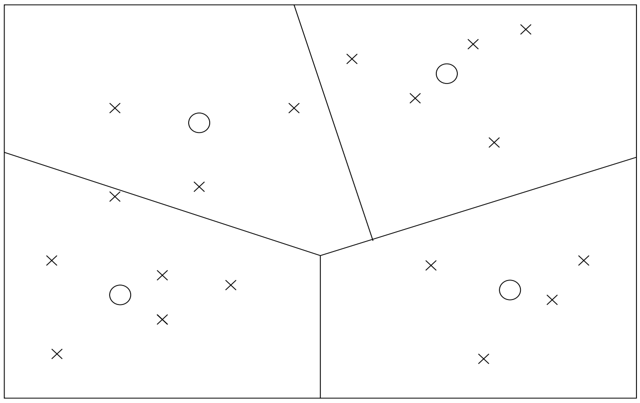


Abbildung 3.3: Beispiel einer Aufteilung des 2-dimensionalen Raumes in Zellen. Repräsentationsvektoren ($\circ$) und Daten-Vektoren ($\times$) sind ebenfalls eingezeichnet.

$$VQ\ :\ \Re^k\ \rightarrow\ CB \tag{3.2}$$

Den Unterraum CB nennt man das Code-Buch, seine Elemente Code-Vektoren oder Code-Wörter. Die Auflösung oder Code-Rate r eines Vektorquantisierers ist definiert als

$$r\ =\ \frac{1}{k}\, log_2 N \tag{3.3}$$

r ist die Anzahl der Bits, die zur Beschreibung einer Komponente des zu codierenden Vektors benötigt werden. Je größer das Code-Buch und je kleiner die Dimension, desto genauer können die Eingabe-Vektoren wiedergegeben werden.

Jedem Code-Vektor $\overrightarrow{y_i}(\overrightarrow{x})$ wird eine sogenannte Zelle R_i im k-dimensionalen Raum zugeordnet:

$$R_i\ =\ \{\ \overrightarrow{x}\ \in\ \Re^k\ :\ VQ(\overrightarrow{x})\ =\ \overrightarrow{y_i}\ \} \tag{3.4}$$

$\overrightarrow{y_i}(\overrightarrow{x})$ ist der Repräsentationsvektor der Zelle, meist wird hier der Schwerpunkt der zugeordneten Vektoren gewählt, wobei die Zuordnung über ein Abstandmaß d erfolgt.

Abb. 3.3 zeigt ein 2-dimensionales Beispiel der Aufteilung des Raumes in Zellen, die durch ihre Schwerpunkte ($\circ$) repräsentiert werden. Ebenfalls eingezeichnet sind einige Daten-Vektoren ($\times$).

Strukturell läßt sich ein Vektorquantisierer auf niederster Ebene in zwei Komponenten zerlegen: einen Encoder, der zu jedem gegebenen Eingabevektor $\overrightarrow{x}$ den Index i der zugehörigen Zelle R_i bestimmt, und einen Decoder, der zu einem gegebenen Index i den entsprechenden Vektor $\overrightarrow{y}_i$ liefert, der diese Zelle repräsentiert.

Qualität

Die Qualität eines Vektorquantisierers wird im allgemeinen über den Mittelwert einer Kostenfunktion oder eines Abstandmaßes d über alle n Trainingsvektoren bestimmt:

$$d \,:\, \Re^k \times CB \rightarrow \Re \tag{3.5}$$

$$\overline{d} = \frac{1}{n} \sum_{i=1}^{n} d(\,\vec{x_i}\,,\, \vec{y}(\vec{x_i})\,) \tag{3.6}$$

Als Maß wird häufig der quadratische euklidische Abstand verwendet:

$$d(\vec{x}\,,\,\vec{y}\,) = \sum_{j=1}^{k} (\,x_j - y_j\,)^2 \tag{3.7}$$

Abhängig von der Anwendung kann jedoch auch ein anderes Abstandsmaß sinnvoll sein. So wird in dieser Arbeit ein lineares Maß verwendet, wie im folgenden Abschnitt beschrieben.

Code-Buch-Training

Die Effizienz eines Vektorquantisierers hängt entscheidend von den Eigenschaften des Code-Buchs ab, dessen Elemente den zu codierenden Daten so gut wie möglich angepaßt werden müssen.

Der große Vorteil eines Vektorquantisierers gegenüber einem Skalarquantisierer liegt in der Möglichkeit, Korrelationen zwischen den aufeinanderfolgenden Datenpunkten auszunutzen. Ähnlich wie in der deutschen Sprache auf ein q sehr häufig ein u folgt, sollten auch die ADC Werte eines Pulses auf den Pads der TPC wiederkehrende Formen haben, auf die das Code-Buch angepaßt werden kann. Abb. 4.2 zeigt Beispielpulse.

Der Standardalgorithmus zum Trainieren eines Vektorquantisierers ist der von S. P. Lloyd entwickelte und nach ihm benannte Algorithmus [30], der auch unter dem Namen LBG-Algorithmus [29] in der Informationstheorie bekannt ist.

Kernstück des Algorithmus ist die *Lloyd-Iteration* für ein gegebenes Code-Buch CB_m:

1. Sei ein Code-Buch $CB_m = \{\,\vec{y_i}\,;\, i = 1, \ldots, N\,\}$ gegeben. Finde diejenige optimale Einteilung des Eingaberaums in Zellen gemäß der *Nächster-Nachbar-Bedingung*, d.h. ordne jedem Trainingsvektor $\vec{x}$ den Index i derjenigen Zelle zu, deren Repräsentationsvektor bezüglich des Abstandsmaßes D den geringsten Abstand vom Eingabe-Vektor hat. (Sind zwei Zellen gleich weit entfernt, wähle diejenige mit dem kleineren Index.)

2. Bestimme für jede Zelle die neuen Repräsentationsvektoren als Schwerpunkte der zugeordneten Eingabe-Vektoren.

Es läßt sich leicht zeigen, daß eine solche Lloyd-Iteration die Qualität des Code-Buchs nur verbessern oder unverändert lassen kann [21].

Zum Trainieren des Code-Buchs wurde in dieser Arbeit ein lineares Maß D verwendet, da aus physikalischen Gründen auch bei der Abbildung eines Eingabe-Vektors auf das entsprechende Code-Wort ein lineares Maß zur Anwendung kam (siehe Kap. 4.3.1). Zur Erzeugung des Code-Buchs wurde also der lineare Abstand zweier Punkte im k-dimensionalen Raum verwendet:

$$D = \sum_{i=1}^{k} \mid x_i - y_i \mid \tag{3.8}$$

Diese Iteration wird nun eingebettet in den sogenannten *Verallgemeinerten Lloyd-Algorithmus*:

1. Starte mit einem gegebenen Code-Buch.
2. Erzeuge mittels Lloyd-Iteration ein neues Code-Buch.
3. Vergleiche die Qualität des alten und neuen Code-Buches. Wenn sich der mittlere Abstand für die beiden Code-Bücher nur um einen hinreichend kleinen Betrag verändert hat, ist der Algorithmus beendet. Andernfalls gehe zu 2.

Woher stammt nun das erste Code-Buch?

Da die Qualität des aus dem Lloyd-Algorithmus resultierenden Code-Buchs nur unwesentlich von der Wahl des ersten Code-Buches abhängt (siehe z.B. [21]), können zum Beispiel N zufällige Trainingsvektoren verwendet werden. Dies wird als "Random-Coding" bezeichnet. Diese N zufällig ausgewählten Vektoren brauchen auch nicht unbedingt die Verteilung der Daten-Vektoren wiederzugeben, die Trainings-Vektoren zur Code-Buch-Optimierung hingegen *müssen* die Struktur der Daten-Vektoren widerspiegeln.

Um ein ausreichend trainiertes Code-Buch zu erhalten, sollte die Anzahl der Trainingsvektoren die Anzahl der Code-Bucheinträge N je nach Art der Daten um etwa einen Faktor 30 - 50 übertreffen.

Da jede Lloyd-Iteration den mittleren Abstand der Trainingsvektoren von den Code-Wörtern nur verbessern kann, muß die Frage nach der Konvergenz des Algorithmus gestellt werden. In der Tat ist es so, daß für einen endlichen Satz von Trainingsvektoren der Verallgemeinerte Lloyd-Algorithmus immer nach einer endlichen Anzahl von Durchgängen konvergiert. Dies läßt sich wie folgt einsehen: Es gibt nur eine endliche Zahl von Möglichkeiten, die Menge der Trainingsvektoren in N Zellen einzuteilen. Jede Aufteilung hat einen festen Wert für die mittlere Entfernung zwischen Code-Wörtern und Trainingsvektoren. Da dieser Abstand monoton mit jeder Iteration sinkt, konvergiert der Algorithmus nach endlich vielen Schritten. Leider ist es keineswegs garantiert, daß das so gefundene Code-Buch das global optimale für die Trainingsvektoren ist. Um diesem Umstand Rechnung zu tragen, wurde bei der Implementation die Technik der Stochastischen Relaxation verwendet, um die Kostenfunktion aus lokalen Minima herauszuführen. Nach jeder Berechnung der Schwerpunkte der in der Iteration m gefundenen Zellen wird zu jeder Komponente eines jeden Code-Vektors ein Rauschen T_m addiert:

$$T_m = \sigma_x^2 \, (1 - \frac{m}{I})^3 \tag{3.9}$$

σ_x^2 steht für mittlere Varianz des Codewortes, und I bezeichnet die Gesamtanzahl der Iterationen. T_m wird auch als Temperatur bezeichnet und die Menge $\{T_m\}$ als Kühlplan ("Cooling Schedule"). Das Code-Buch wird sozusagen mit jeder Iteration weiter eingefroren. Weitere Details und alternative Möglichkeiten zur Überwindung lokaler Minima der Kostenfunktion finden sich in [21].

3.5 MPEG

Videodaten kennzeichnet i.a. ein hoher Grad an Redundanz: Zum einen enthalten die Einzelbilder häufig selbst redundante Informationen, zum anderen gibt es starke Korrelationen zwischen ihnen, denn die Unterschiede zwischen aufeinander folgenden Bildern sind meist nur gering. Bewegtbilder bieten also einen Spielraum zur Kompression, der genutzt werden kann, um die Bitrate für Übertragung und Speicherung so weit wie möglich zu reduzieren. Da das Interesse an effizienten Kompressionsverfahren für audio-visuelle Informationen hoch ist, gibt es auf diesem Sektor intensive Entwicklung und leistungsfähige Techniken und Werkzeuge. Aufgrund der hohen Kompressionsraten im Vergleich zu anderen Codierungen nimmt MPEG hierbei eine wichtige Stellung ein.

Das Akronym MPEG (Moving Pictures Experts Group) bezeichnet einen Satz von Standards für die Komprimierung von Audio- und Videodaten, der vom Joint ISO/IEC (International Standardization Organisation/International Electrotechnical Commission) Technical Committee on Information Theory entwickelt wurde. Er unterteilt sich in mehrere Einzelnormen.

MPEG-1 ist der ursprüngliche Standard für die Speicherung und das Abrufen von Video- und Audioinformationen, der für die CD-ROM-Technologie entwickelt wurde. Er definiert mittlere Bandbreiten von bis zu 1.5 Mega-Bit Pro Sekunde (Mbps), zwei Audiokanäle sowie zeilensprungfreies (non-interlaced) Video.

MPEG-2 ist eine Erweiterung von MPEG-1 für den Fernsehrundfunk und HDTV (High Definition Television). Er unterstützt höhere Bandbreiten bis 100 Mbps, 5 Audiokanäle sowie Interlaced Video.[3]

[3] MPEG-3 war für HDTV vorgesehen, was von MPEG-2 bereits mit abgedeckt wurde, so daß der MPEG-3 Standard verworfen wurde.

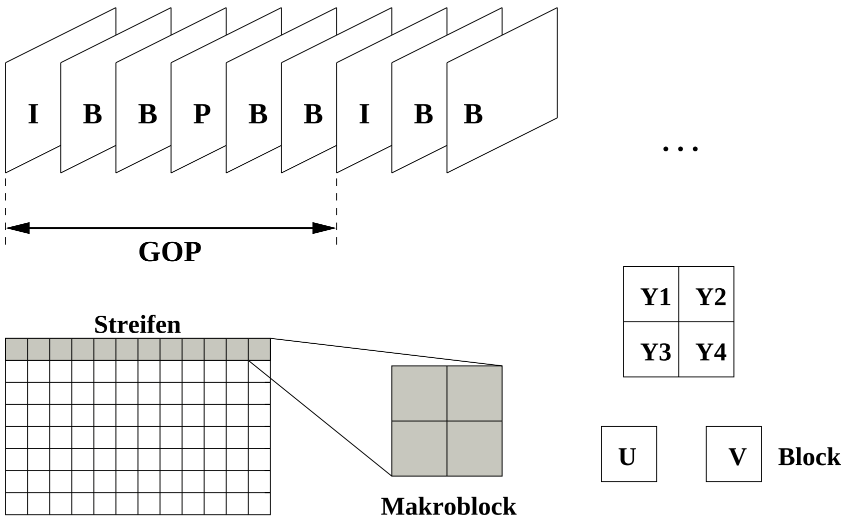


Abbildung 3.4: Zerlegung einer MPEG-Sequenz. Die größte Einheit sind die Bildgruppen (Group of pictures, GOP). Sie bezeichnen den Abstand zwischen aufeinanderfolgenden I-Bildern. Jedes Einzelbild wird in Streifen, jeder Streifen in Makro-Blöcke geteilt. Für die Luminanz-Komponenten bestehen diese Blöcke aus 16x16, für die Chrominanz-Komponenten aus 8x8 Pixeln.

Der MPEG-4 Standard ist ausgelegt für Video-Telefone und Multimedia-Anwendungen. Die Bandbreite liegt hier weit niedriger (bis 64 kbps).

MPEG-7 ist ein Standard in der Entwicklungsphase. Er beinhaltet zusätzlich zu den Multimediadaten, Informationen über den Inhalt, mit denen es möglich sein wird, mittels Suchmaschinen gezielt nach Videodaten zu einem Thema zu suchen.

Das Verfahren

Die MPEG-Codierung läßt sich in fünf Schritte teilen (Die Codierung wird hier vereinfacht dargestellt. Für eine Vertiefung sei auf [18], [31] verwiesen.).

- Bildaufbereitung: Das Eingabebild wird in eine Luminanz (Y)- und zwei Chrominanzanteile (U,V) zerlegt. Da das menschliche Sehsystem sensitiver auf Helligkeits- als auf Farbunterschiede ist, werden die Anteile in einem Y:U:V-Verhältnis (meist 4:1:1 oder 4:2:2) gespeichert und so die Auflösung der Chrominanzanteile reduziert.

- Zerlegung des Bildes in Streifen und Makroblöcke: Das Einzelbild wird in 16x16 (Chrominanz) bzw. 8x8 Pixel (Luminanz) große Blöcke zerlegt (siehe Abb. 3.4).

- Bewegungschätzung: Innerhalb eines Suchfensters wird ein Luminanz-Makroblock mit Makroblöcken vorhergehender oder nachfolgender Bildern verglichen, um Bewegungen innerhalb eines Bildes auszumachen. Ist ein passender Block gefunden, so wird ein Bewegungsvektor und die Unterschiede zwischen den Bildern gespeichert. Da zwischen

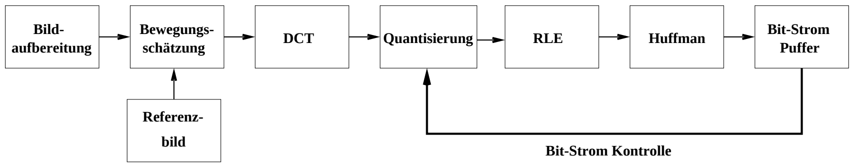


Abbildung 3.5: Schema des MPEG-Encoder (Erläuterung im Text, abgewandelt übernommen aus [18]).

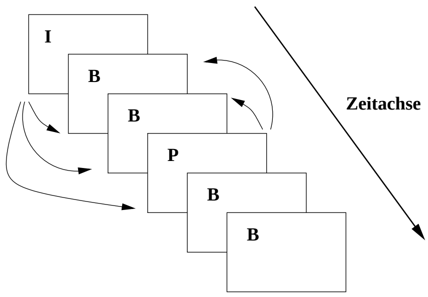


Abbildung 3.6: Abhängigkeiten zwischen den Einzelbildern einer MPEG-Sequenz.

aufeinanderfolgenden Bildern meist nur geringe Unterschiede auftreten, ist dieses Verfahren sehr effizient in seinem Kompressionsverhalten.

- Diskrete Cosinus-Transformation (DCT):[4] Luminanz- und Chrominanz-Blöcke werden in den Frequenzraum transformiert.
- Quantisierung: Die Koeffizienten der DCT werden gemäß der Eigenschaften des menschlichen Sehsystems quantisiert. Dieses Verfahren ist mit dem beim JPEG-Formate verwendeten identisch (siehe z.B. [18]). Da hochfrequente Anteile weniger gut wahrgenommen werden als niederfrequente, werden durch die Quantisierung die hochfrequenten Anteile der DCT unterdrückt und auf Null abgebildet. Diese lassen sich dann im nächsten Schritt weiter komprimieren.
- RLE- und Huffman-Codierung der quantisierten Koeffizienten.

Da der Encoder darauf ausgelegt ist, konstante Bitraten zu erzeugen, wird die Anzahl der erzeugten Bits rückgekoppelt und damit die zur Quantisierung der DCT-Koeffizienten benutzte Matrix angepaßt.

Abb. 3.5 faßt das Verfahren zusammen.

I-, P- und B-Bilder

MPEG unterscheidet im wesentlichen drei unterschiedliche Bildtypen.

[4] Eine Einführung in die Diskrete Cosinus-Transformation und ihre Anwendung in der Bildverarbeitung kann in [36] gefunden werden.

I-Bilder (Intra Frames) sind vollständige Bilder, sie benötigen keine weiteren Information für die Decodierung. Sie bilden die Referenz nachfolgender *P-Bilder* (Predictive Frames). Die Makroblöcke der P-Bilder bestehen aus den Vektoren der Bewegungschätzung oder, falls die Schätzung kein Ergebnis gebracht hat, den Originalblöcken. Vorhergehende und nachfolgendende I- und P-Bilder werden von den *B-Bildern* (Bidirectional Frames) als Referenz gebraucht und durch deren Interpolation wiederhergestellt. B-Bilder selbst werden nie als Referenz gebraucht. Mehr I-Bilder bedeuten i.a. höhere Qualität, mehr B-Bilder bessere Kompression. Abb. 3.6 zeigt die Referenzen zwischen den verschiedenen Bildtypen.

Kapitel 4

Untersuchung der Kompressionsverfahren

"Why," said the Dodo, "the best way to explain it is to do it."
Lewis Carroll (1832-1898)

4.1 Das Datenformat

Der Large Hadron Collider am CERN wird voraussichtlich im Jahr 2005 in Betrieb gehen. Aus diesem Grunde und da noch keine Daten von Detektor-Prototypen zur Verfügung stehen, werden zur Untersuchung der verschiedenen Kompressionsverfahren ausschließlich simulierte Daten verwendet, die mit Hilfe des GEANT 3-basierten Simulationsmoduls im Analyse-Rahmenprogramm *aliroot* [23] erzeugt werden. Aus diesen simulierten Teilchenspuren eines 12 Kammern großen Sub-Events mit einer Multipizität von $\mathrm{dN}_{charged}/\mathrm{d}\eta$=8000 werden anschließend die digitalisierten 10 Bit Datenwerte der TPC rekonstruiert.[1] Wie in Kap. 3.1 bereits erwähnt, wird aus physikalischen Gründen und zur Datenreduktion ein 45 Grad Konus aus den Daten herausgeschnitten.

Da die ADCs eine *absolute* Genauigkeit von etwa einem Bit haben, ist die relative Genauigkeit für große ADC Werte sehr viel besser als für kleine Werte. Bei der verwendeten nichtlinearen Kompression von 10 auf 8 Bit stellt sich nun neben der Reduzierung des Datenvolumens eine *konstante relative* Genauigkeit der ADC Werte über den gesamten dynamischen Bereich ein. Abb. 4.1 zeigt ein Diagramm der zur Transformation verwendeten Tabelle.[2] Nach dieser Transformation werden die Null-Intervalle lauflängencodiert.

[1] Diese Binärdaten liegen dann in der Datei nach Kammern, Pad-Reihen, Pads und Zeit-Bins geordnet vor.

[2] Für die Konvertierung von 10 Bit nach 8 Bit wurden *aliroot*-Klassen von Marian Ivanov, Gesellschaft für Schwerionen-Forschung (GSI) in Darmstadt, verwendet.

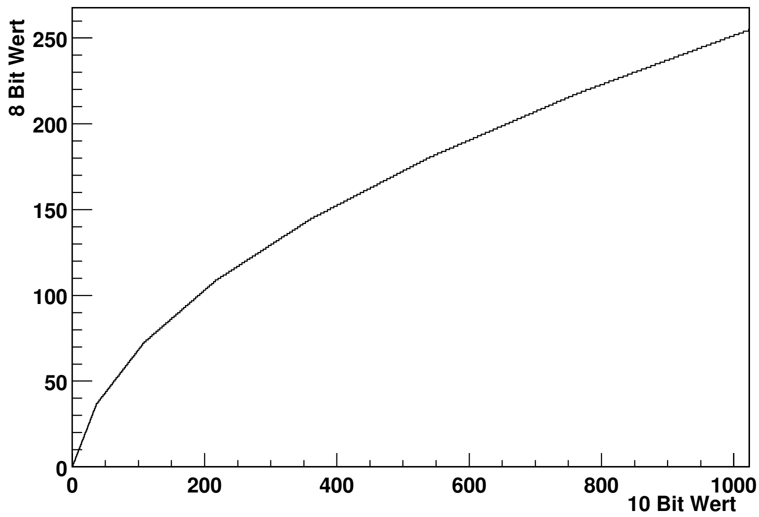


Abbildung 4.1: Diagramm der Tabelle für die Transformation von 10 nach 8 Bit.

Als Ausgangspunkt für die Kompression wird das Format noch leicht verändert: eine Schwelle bei einem ADC Wert von 10 wird angelegt und nur Werte größer oder gleich dieser Schwelle werden zusammen mit einer Vor- und Nachgeschichte von jeweils einem ADC Wert in Zeit-Richtung abgespeichert. Ergänzend werden die Position eines solchen eindimensionalen Pulses innerhalb der 445 Zeitschritte eines Pads sowie seine Länge mit 9 bzw. 4 Bits gespeichert. Abb. 4.2 illustriert das Vorgehen. So werden also keine lauflängencodierten Nullintervalle mehr benötigt. Da die Anzahl der Zeitschritte eines Pads 445 beträgt, decken 9 Bits den vollen Bereich der möglichen Positionen ab. Die Breite von 4 Bits für die Speicherung der Länge eines Pulses ist ein Kompromiß zwischen Platzbedarf und der Längenverteilung der Pulse (s.a. Abb. 4.6). Liegen mehr als 15 ADC Werte hintereinander oberhalb der Schwelle, so werden zwei oder entsprechend mehr Pulse angelegt.

Der verwendete Datenstrom hat somit die in Abb. 4.3 gezeigte Form.

Dieses Datenformat ist ein realistischer Ausgangspunkt zur Bewertung der verschiedenen Kompressionsverfahren, da es nur noch aus Headerinformationen und den ADC Werten besteht. Das Verfahren reduziert die Größe des Sub-Events bereits auf etwa 88 %. Diese wie auch alle im folgenden angegebenen Kompressionsraten beziehen sich auf die nullunterdrückte lauflängencodierte Roh-Event-Größe.

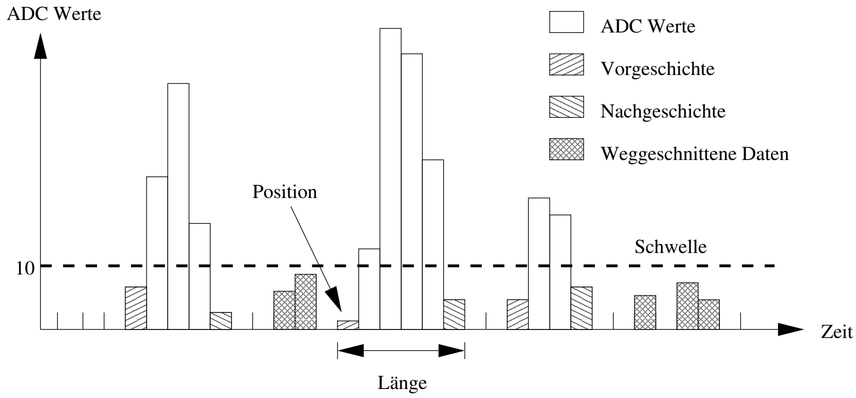


Abbildung 4.2: Erzeugung des verwendeten Datenformats (Erläuterung im Text).

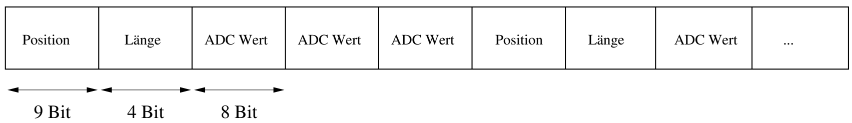


Abbildung 4.3: Bitstrom des verwendeten Datenformats.

4.2 Die Huffman-Codierung

4.2.1 ADC Werte

Abb. 4.4 zeigt die Häufigkeitsverteilung der 8 Bit großen ADC Werte in der Referenzdatei in logarithmischer Auftragung.

Neben dem globalen exponentiellen Abfall, sind das von der Front-End-Elektronik abgezogene Pedestal der ADC Werte 1 und 2 sowie die als 255 codierten Werte der ADCs in der Sättigung gut zu erkennen. Die Stufen in der Verteilung haben zwei unterschiedliche Ursachen: die erste Stufe bei einem ADC Wert von 10 ist eine Folge der oben schon diskutierten zusätzlich angelegten Schwelle bei einem ADC Wert von 10. Alle folgenden Stufen haben ihren Ursprung in der Konvertierung von 10 nach 8 Bit.

Diese exponentielle Verteilung legt es nahe, für die Kompression einen Entropie-Codierer, wie etwa die Huffman-Codierung, zu testen. Für die Effizienz dieses Algorithmus' ist die hier vorliegende stark unterschiedliche Auftrittswahrscheinlichkeit der einzelnen Zeichen von entscheidender Bedeutung.

Neben den Huffman-Bäumen für die ADC Werte selbst, wurden auch solche für die Headerinformationen, d.h. die Position innerhalb der 445 Zeitschritte eines Pads, sowie für die Pulslänge angelegt und diese beiden Werte so ebenfalls mitcodiert. Abb. 4.5 und Abb. 4.6 zeigen die Häufigkeitsverteilungen der Positionen und der Pulslänge jeweils in logarithmischer Auftragung.

Der Anstieg in der Verteilung der Positionen findet seine Erklärung in dem herausge-

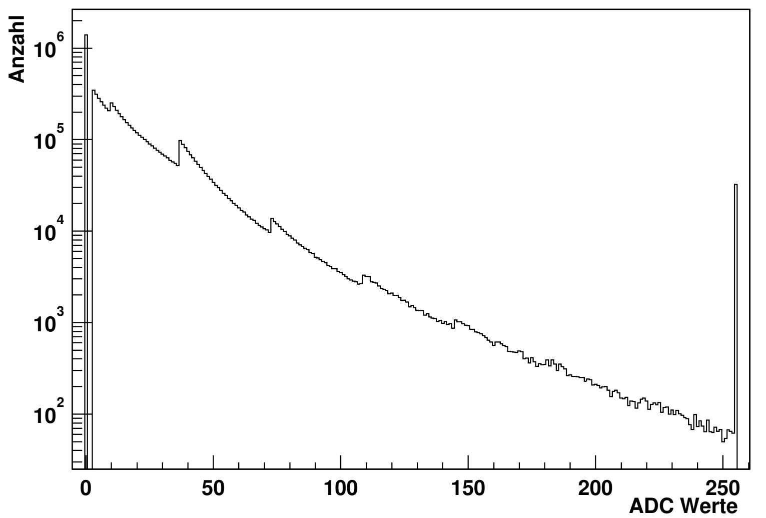


Abbildung 4.4: Verteilung der ADC Werte in den Referenz-Daten.

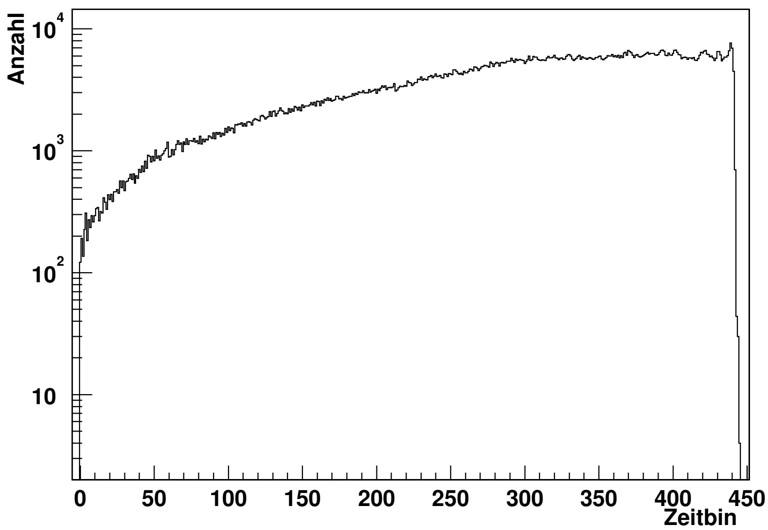


Abbildung 4.5: Verteilung der Positionen innerhalb der 445 Zeit-Bins eines Pads.

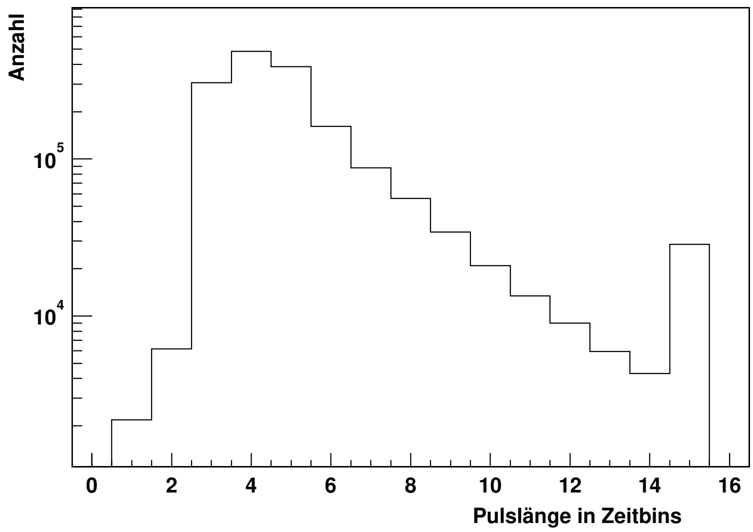


Abbildung 4.6: Verteilung der Pulslänge (inklusive Vor- und Nachgeschichte).

schnittenen 45 Grad Konus: Für die inneren Pad-Reihen werden abhängig von ihrem Abstand a von der Strahlröhre die ersten n Zeit-Bins nicht ausgelesen. Dieses n nimmt mit wachsendem a ab. Infolgedessen sind für die innersten Pad-Reihen lediglich sehr hohe Zeit-Bins relevant, während für die weiter außen gelegenen zusätzlich auch kleinere ausgelesen werden.

Die Verteilung der Pulslängen in Abb. 4.6 hat ihr Maximum bei einem Wert von 4 Zeit-Bins. Wegen der hohen Auslastung der TPC ($\sim$ 8000 geladene Spuren pro Rapiditätseinheit) werden die Pulsformer schon in der Simulation auf eine sehr kurze Shaping-Zeit eingestellt, um so gut wie möglich ein Überlappen der Cluster zu verhindern. Aus diesem Grunde hat die Pulslängenverteilung ihr Maximum inklusive Vor- und Nachgeschichte bei einer Länge von 4 Zeit-Bins.

Man sieht ebenfalls einen starken Abfall zu großen Pulslängen hin. Der Peak bei einer Pulslänge von 15 Zeit-Bins erklärt sich mit der Wahl von 4 Bits für die Codierung der Länge und somit einer *maximalen* Länge von 15 Zeit-Bins. Pulslängen von 1 bzw. 2 Zeit-Bins entstehen an den Rändern von Pads oder Pad-Reihen.

Die Codierung aller genannten Werte führt zu einer Reduktion des Datenvolumens auf 65%.

Die UNIX Programme *pack* (statische Huffman-Codierung) und *compact* (adaptive Huffman-Codierung) können das Datenvolumen hingegen nur auf etwa 80% der ursprünglichen Größe reduzieren.

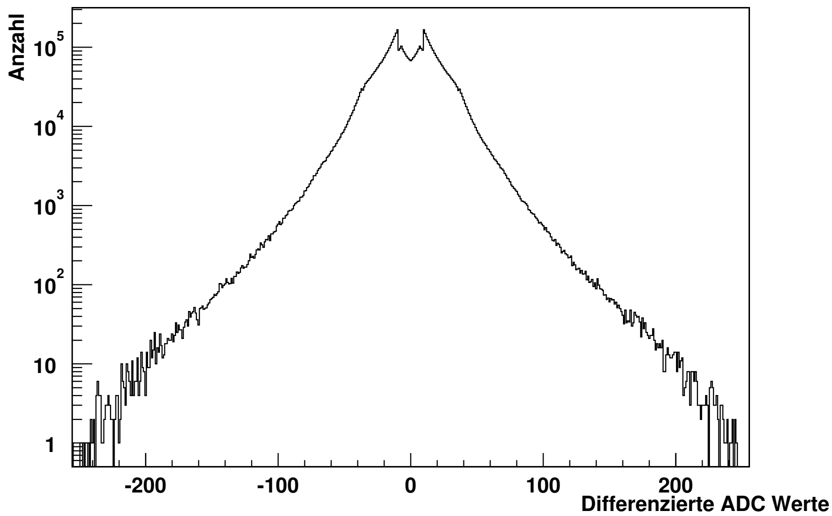


Abbildung 4.7: Verteilung der differenzierten ADC Werte.

4.2.2 Differenzierte Daten

Aufgrund der begrenzten Bandbreite der Elektronik sollte der dynamische Bereich des differenzierten Signals kleiner sein als der des ursprünglichen und somit eine stärkere Korrelation zwischen aufeinanderfolgenden Datenpunkten eine bessere Kompression gewährleisten.

Im Hinblick darauf wird als nächstes die Verteilung der differenzierten ADC Werte untersucht. Abb. 4.7 zeigt die Verteilung der Differenzen aufeinanderfolgender ADC Werte in dem Referenz-Event in logarithmischer Auftragung.

Wie bei den ADC Werten selbst, ist auch hier eine exponentielle Verteilung der Häufigkeiten zu sehen. Die beiden Einschnitte in der Verteilung bei den ADC Werten ± 10 sind wiederum eine Folge der zusätzlichen angelegten Schwelle. Die leichte Änderung der Steigung bei einem ADC Wert von 37 ist eine Folge der Konvertierung von 10 nach 8 Bit.

Mittels der Huffman-Codierung, die - genau wie bei den ADC Werten - auch auf die Headerinformationen angewendet wird, kann das Datenvolumen in dieser Form auf 73 % reduziert werden.

Die Codierung der ADC Werte selbst erreicht also eine etwas bessere Kompression.

4.3 Vektorquantisierung

4.3.1 Vorbemerkungen

Die Fähigkeit eines Vektorquantisierers, Eingabe-Vektoren mit sinnvoller Rate auf Elemente des Code-Buchs abzubilden, hängt von mehreren Faktoren ab.

Zum ersten spielt die Qualität des Code-Buchs eine Rolle. Sind die Code-Vektoren nur schlecht an die Daten angepaßt, so wird auch die Rate, mit der Eingabe-Vektoren erkannt werden, relativ niedrig sein. Deshalb wird das Code-Buch mit Trainingsvektoren optimiert.

Dann sind bei fester Code-Buch-Größe die Dimension des Eingaberaumes und der dynamische Bereich der Komponenten (hier: der ADC Werte) wichtige Punkte, da mit zunehmender Dimension k und zunehmender Größe des Intervalls, in dem die Werte der Komponenten liegen, die Kombinatorik für die Eingabe-Vektoren stark anwächst. So sind es für eine Vektordimension $k = 3$ und einem Bereich der ADC Werte bis 255 (8 Bit Daten) knapp 17 Mio. mögliche Eingabe-Vektoren, während sich dieser Wert für $k = 4$ und ADC Werte bis 1024 (10 Bit Daten) bereits auf über 10^{12} Möglichkeiten für die Vektoren beläuft.

Als letzter Punkt sei die Möglichkeit der verlustbehafteten Kompression angeführt. Ist der Eingabe-Vektor kein Element des Code-Buchs, so kann er dennoch mit einem Code-Wort identifiziert werden, wenn die Abweichungen in einem gewissen Toleranzbereich liegen. Dieser Toleranzbereich hängt von den jeweiligen Gegebenheiten ab, muß für die spezielle Anwendung angepaßt werden und in die Abbildungskriterien eingehen. Im vorliegenden Fall ist für einen Puls neben seiner Position vor allem seine Ladung interessant, die gerade durch das Integral der einzelnen ADC Werte eines Pulses angenähert wird. Die Ladungsmessung liefert den Wert für die spezifische Ionisation dE/dx, die zur Identifizierung der erzeugten Teilchen dient. Aus diesem Grunde wird als ein Abbildungskriterium f die relative Ladungsdifferenz zwischen Code-Wort und Daten-Vektor gewählt. Bezeichnet c die angenäherte Ladung des Daten-Vektors und cv die Ladung des Code-Vektors, so erhält man für f:

$$f = \frac{\mid c - cv \mid}{c} * 100 \tag{4.1}$$

Als weitere Kriterien dienen die relative Abweichung einzelner Komponenten des Daten- bzw. Code-Vektors, um zu verhindern, daß bei gleicher Ladung die Form eines Pulses (und damit auch z.B. auch sein Schwerpunkt, der für die Ortsbestimmung herangezogen wird) übermäßig stark verändert wird. Eine Abweichung von einem ADC Wert wurde als Rauschen betrachtet und immer zugelassen.

Die Wahl des Toleranzbereiches, d.h. die maximale Abweichung zwischen Daten- und Code-Vektor, hängt von dessen Einfluß auf die in den Daten enthaltene Physik ab. Deshalb werden zur Bestimmung des akzeptablen Toleranzbereiches verschiedene physikalische Größen untersucht, siehe Kapitel 5.

4.3.2 Vorgehen

Um einen geeigneten Satz von Parametern (Vektordimension, Code-Buch-Größe, maximal tolerierte Abweichung) zu finden, werden diese Parameter in verschiedenen Kombinationen untersucht. Tabelle 4.1 zeigt einen Überblick über die verschiedenen getesteten Parameter.

Vektordimension \ Code-Buch-Größe	32	64	128	256	512	1024
2	x	x	x	x		
3		x	x	x	x	
4			x	x	x	x

Tabelle 4.1: Tabelle der getesteten Parameter-Kombinationen.

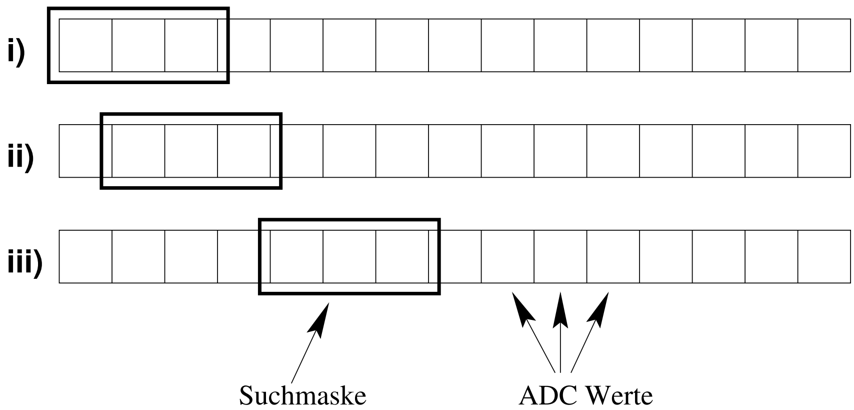


Abbildung 4.8: Verwendung einer Maske bei der Suche nach passenden Code-Vektoren:
i) erster Ansatz der Maske.
ii) beim ersten Versuch konnte kein passender Eintrag im Code-Buch gefunden werden, der erste ADC Wert wird gespeichert und die Maske eine Position nach rechts verschoben.
iii) beim zweiten Versuch konnte ein passender Eintrag gefunden werden, der Eintrag im Code-Buch wird gespeichert und die Maske bewegt sich um die Vektordimension nach rechts über die ADC Werte.

Die Wahl des Bereiches der Vektordimension ergibt sich aus Abb. 4.6: Pulse der Länge 4 treten am häufigsten auf. Zu höheren Dimensionen hin nimmt die Anzahl der Pulse, die für die Codierung in Frage kommen, stark ab. Da für eine Vektordimension k alle Pulse mit einer Länge kleiner k nicht codiert werden können, wird wegen des zu geringen Anteils der erkannten Eingabe-Vektoren auf eine tiefergehende Untersuchung der Vektordimensionen 5 oder höher verzichtet. Für die beiden nächst kleineren Dimensionen, 2 und 3, werden allerdings verschiedene Parametersätze untersucht. Die tolerierte Maximalabweichung der Summe der Komponenten eines Pulses wird jeweils in einem Bereich von 0 bis 60% variiert.

Bei der Suche nach einem geeigneten Code-Wort wird eine Suchmaske mit der Größe der jeweiligen Vektordimension über den eventuell längeren Puls geschoben, um möglichst häufig die Möglichkeit zu haben, einen passenden Code-Vektor zu finden. In Abb. 4.8 ist dieses Vorgehen verdeutlicht. Kann kein passender Code-Vektor gefunden werden, wird der jeweilige ADC Wert unverändert abgespeichert. Ob im codierten Datenstrom ein ADC Wert im Original gespeichert oder mehrere zu einem Vektor zusammengefaßt werden, wird mit einem zusätzlichen Bit codiert.

Damit es nicht nötig ist, bei der Suche nach einem passenden Code-Vektor immer das

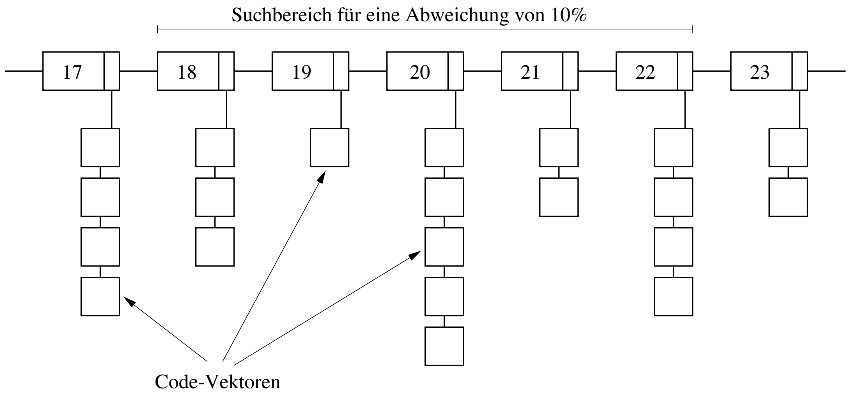


Abbildung 4.9: Ausschnitt aus der Datenstruktur zur Suche nach einem geeigneten Code-Vektor. Die Liste ist sortiert nach dem jeweils ersten Wert der Code-Vektoren. Ebenfalls eingezeichnet ist der Suchbereich bei einer maximalen Abweichung von 10% der Code-Vektor-Komponenten von den Komponenten des Daten-Vektors (in diesem Beispiel habe der Daten-Vektor als erste Komponente einen ADC Wert von 20).

gesamte Code-Buch durchsuchen zu müssen, werden die Code-Vektoren in einer Liste verwaltet, sortiert nach dem jeweils ersten ADC-Wert. Jeder Eintrag dieser Liste verweist auf eine zweite Liste, welche die Nummer der Code-Wörter enthält. So kann auch der zugelassene Bereich der Suche sukzessive vergrößert werden. Abb. 4.9 zeigt einen Ausschnitt aus dieser Liste.

Der Aufwand für die Kompression der Daten mittels des Vektorquantisierers ist erheblich höher als dies z.B. für die Huffman-Codierung der Fall ist. So wird jeder Puls mit der oben beschriebenen Suchmaske in Vektoren zerlegt, für die dann, abhängig vom jeweiligen Toleranzbereich, die Liste nach einem passenden Code-Vektor durchsucht wird. Das bedeutet für jeden Eintrag der Liste im Toleranzbereich den Vergleich bezüglich der relativen Abweichung der Summe der einzelnen ADC Werte sowie der relativen Abweichung der einzelnen Komponenten des Vektors mit dem Daten-Vektor. Mit zunehmender Vektordimension und größer werdenden Code-Buch steigt dieser Aufwand erheblich an, da die Zahl der Vergleiche stark zunimmt. Kann kein passender Code-Vektor gefunden werden, so wird die Suchmaske, wie oben beschrieben, verschoben und mit einem neuen Daten-Vektor, der teilweise dieselben Daten wie der vorige enthält, die Prozedur wiederholt. Dieser erhebliche Aufwand kann sich nur durch wesentlich bessere Kompressionsraten rechtfertigen.

Um einen Eindruck von der Struktur der Daten und des angepaßten Code-Buchs zu erhalten, zeigt Abb. 4.10 als Beispiel die Verteilung von 10.000 der 4-dimensionalen Daten- und der erzeugten Code-Buch-Vektoren in 3-dimensionalen Projektionen.

Die Koordinaten x, y, z und t bezeichnen nacheinander die Zeitbins der ADC Werte des Pulses. Man erkennt deutlich eine erhöhte Dichte bei kleinen ADC Werten. Die Dich-

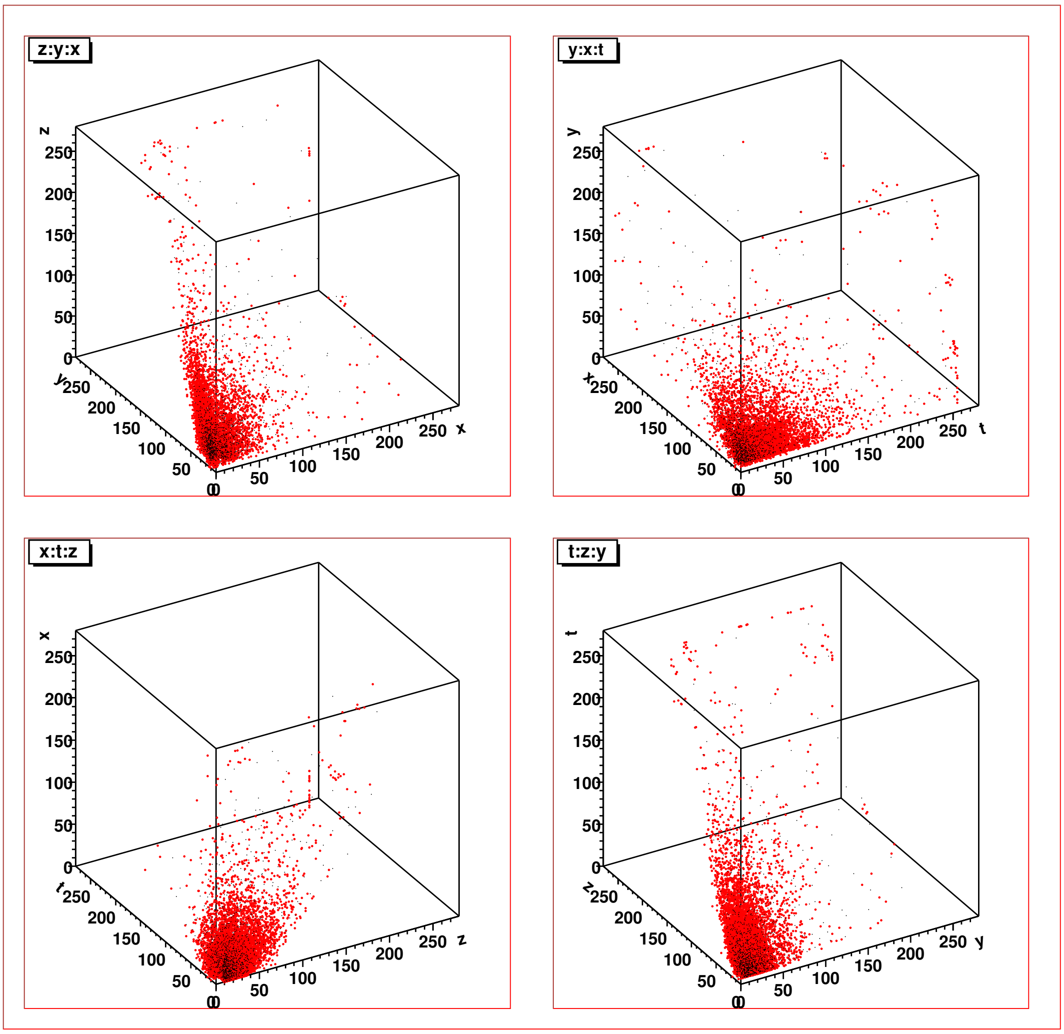


Abbildung 4.10: Projektionen vierdimensionaler Daten-Vektoren (rot) und eines angepaßten Code-Buchs (schwarz) in den dreidimensionalen Raum.

teverteilung der Code-Wörter des trainierten Code-Buchs entspricht - wie erwartet - der Dichteverteilung der Daten.

Richtig deutlich wird die Korrelation zwischen den einzelnen Komponenten eines Pulses aber erst in der Projektion auf 2 Dimensionen. Abb. 4.11 zeigt die Seiten der Würfel aus Abb. 4.10.

Deutlich zu erkennen sind in den ersten drei Bildern die angelegte Schwelle bei einem ADC Wert von 10 als kleines freies Quadrat bei Null. Zwei aufeinanderfolgende Datenwerte, wie eben die Paare x/y, y/z oder z/t, dürfen nicht beide unterhalb der Schwelle liegen, wenn sie noch als *ein* Puls gelten sollen. Zusätzlich erkennt man den Zusammenhang zweier direkt aufeinander folgender Datenpunkte: So sind z.B. die Werte des dritten und vierten ADC Wertes (z und t) eher gleich groß, da sie sich z.B. in der Mitte eines längeren Pulses befinden, und liegen daher näher bei der Winkelhalbierenden, während der erste und der letzte ADC Wert (x und t) eher bei einer der Achsen liegen. Dies hat seine Ursache in der Tatsache, daß die meisten Pulse eine Länge von 4 Zeit-Bins haben, so daß die Werte von x und t meist schon wieder unterhalb der Schwelle liegen, da sie als Vor- bzw. Nachgeschichte zweier ADC Werte, die oberhalb der Schwelle liegen, abgespeichert wurden. Feinere Strukturen sind an den eingetragenen Vektoren des Code-Buchs zu erkennen: In der Auftragung x:y ist die Häufung kleiner x-Werte zu sehen, die den Anfang des Pulses bilden. Bei der Auftragung z:t gilt das gleiche für die Werte von t, die das Ende des Pulses darstellen. Die Verteilung y:z ist dagegen symmetrisch, da hier die beiden mittleren ADC Werte des Pulses aufgetragen sind. Auch die Verteilung x:t ist symmetrisch, da hier die beiden End-Werte des Pulses eingezeichnet sind. In den Diagrammen x:z und y:t ist wiederum deutlich die Schwelle bei einem ADC Wert von 10 zu sehen, über der die Werte von z und y im allgemeinen liegen.

Ferner sind die Werte der ADCs in der Sättigung bei 255 zu sehen, sowie das von der Front-End-Elektronik abgezogene Pedestal der Werte 1 und 2 bei der Auftragung x:z.

4.3.3 Erkennungs- und Kompressionsraten

Abhängig von Code-Buch-Größe und Vektor-Dimension wird die Rate untersucht, mit der der Vektorquantisierer ein geeignetes Code-Wort finden kann, und inwieweit das Datenvolumen dadurch zu reduzieren ist.[3]

Die Erkennungsrate wird definiert als der Anteil der erfolgreichen Versuche, ein geeignetes Code-Wort zu finden, an der Gesamtzahl der Versuche. Die Kompressionsrate ist definiert als der Bruchteil des komprimierten Datenvolumens an der Größe der nullunterdrückten lauflängencodierten Originaldaten in % und schließt die Huffman-Codierung für die Headerinformationen und die benutzten Code-Buch-Einträge mit ein.

Um die Abhängigkeit der Erkennungsraten von der Größe des Code-Buches zu verdeutlichen, zeigt Abb. 4.12 die Raten für eine feste Vektordimension und verschiedene Code-

[3]Für eine vollständige Liste der Kurven für die einzelnen Parameter sei auf Anhang A verwiesen.

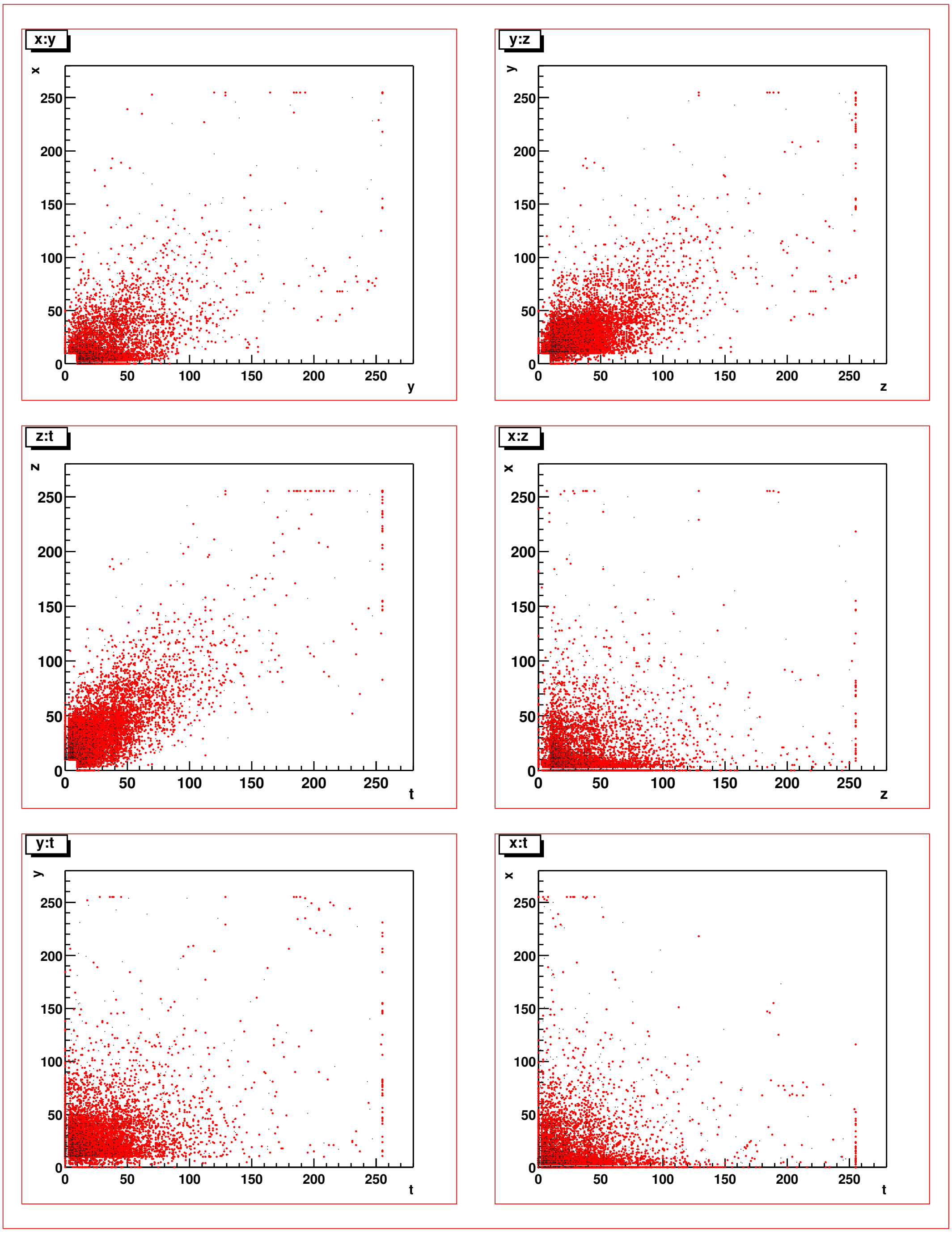


Abbildung 4.11: 2D-Projektionen der 4D-Daten-Vektoren und des angepaßten Code-Buchs aus Abb. 4.10.

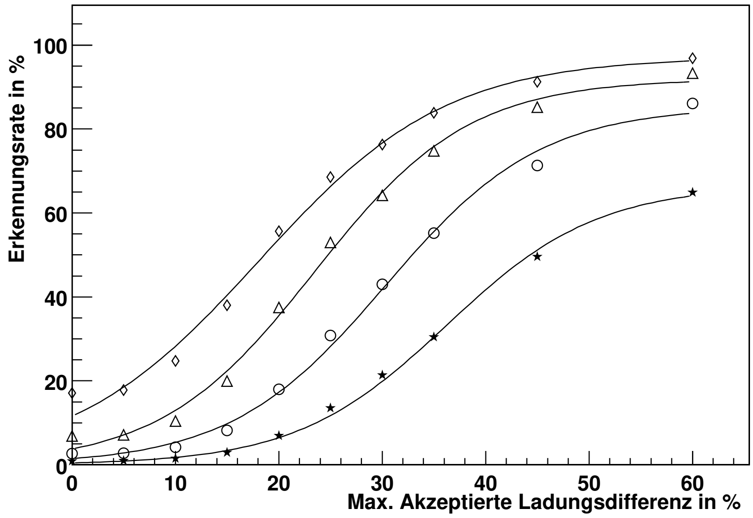


Abbildung 4.12: Erkennungsraten für eine Vektordimension von 3 und Code-Buch-Größen von 64 ($\star$), 128 ($\circ$), 256 ($\triangle$) und 512 ($\diamond$) Elementen (von unten nach oben). Angepaßt an die Werte sind Kurven der Form (4.2).

Buch-Größen. Als Beispiel wurde eine Vektordimension von 3 gewählt.

An die Datensätze wurden mittels einer χ^2-Anpassung sogenannte "Epidemie-Kurven"[4] mit fünf freien Parametern angepaßt:

$$g_{\alpha,\beta,\gamma,\delta,\epsilon}(x) = \epsilon + \frac{\gamma}{1 + \alpha e^{-\beta(x+\delta)}} \tag{4.2}$$

Wie man deutlich sieht, werden die Erkennungsraten (wie auch die Kompressionsraten, siehe weiter unten) durch dieses Modell relativ gut beschrieben.

Während bei geringem tolerierten Fehler die Erkennungsrate noch sehr niedrig ist, steigt sie mit zunehmender maximaler Abweichung exponentiell an bis sie in Sättigung geht und zu jedem Daten-Vektor ein passendes Code-Wort gefunden werden kann. Für ein Code-Buch mit mehr Einträgen liegt die Rate der erkannten Vektoren immer über derjenigen für kleinere Code-Bücher, wie es zu erwarten ist, da ja viel mehr Code-Wörter als Optionen vorliegen. Außerdem steigt mit zunehmenden maximal tolerierten Fehlern die Rate für größere Code-Bücher steiler an.

Die zugehörigen Kompressionsraten zeigt Abb. 4.13. Die höhere Erkennungsrate gibt

[4]"Epidemie-Kurven" (Epidemic Curves) werden normalerweise zur Beschreibung der Ausbreitung von Ansteckungskrankheiten benutzt. Dabei ist die Abszisse die Zeitachse, während auf der Ordinate die Häufigkeit der Krankheitsfälle aufgetragen ist. Auch eine Kurve, die in Abhängigkeit von der Zeit die Zahl der infizierten Personen beschreibt, wird als "Epidemic Curve" bezeichnet. Die einzelnen Parameter haben dabei folgende Bedeutung: α ist die Zahl derjenigen Menschen, die zu Beginn der Epidemie nicht infiziert sind, β beschreibt den Anteckungsgrad der Krankheit, γ ist die Zahl der maximal anzusteckenden Personen, δ ist eine Verschiebung auf der Zeitachse (Abszisse) und ϵ ein Plateau der zu Beginn bereits infizierten.

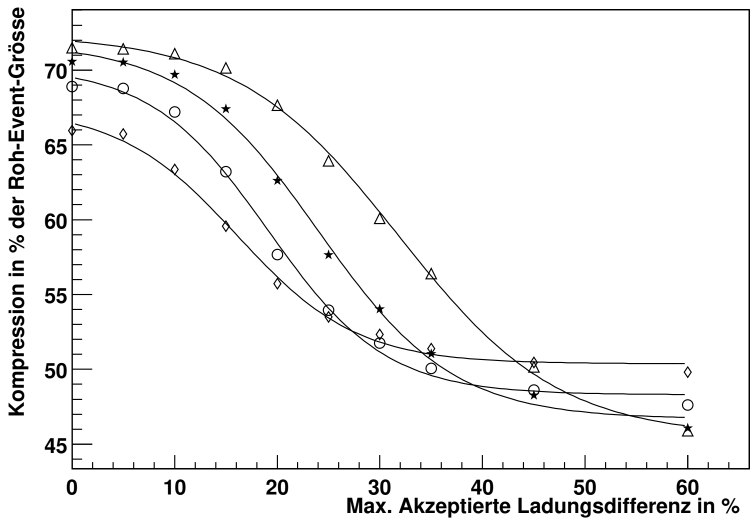


Abbildung 4.13: Kompressionsraten für eine Vektordimension von 4 und Code-Buch-Größen 128 ($\triangle$), 256 ($\star$), 512 ($\circ$) und 1024 ($\diamond$) Elementen.

bei kleinen Fehlern den Ausschlag, daß das Code-Buch mit den meisten Einträgen die beste Kompression erzielt. Dieser Vorteil hält sich bis zu maximal erlaubten Fehlern von etwa 30 %. Ab diesem Punkt wirkt sich aus, daß zur Benennung eines Eintrags in den kleineren Code-Büchern weniger Bits vonnöten sind und die Daten aus diesem Grunde in diesem Bereich besser komprimiert werden können. Um das zu verdeutlichen zeigt Abb. 4.14 die Kompressionsraten aus Abb. 4.13 nochmals im Bereich einer maximal tolerierten Abweichung von 30 bis 60 %. Hier sind deutlich die Vorteile in der Kompression bei den kleineren Code-Büchern aufgrund der geringeren zur Codierung notwendigen Bitzahl im Sättigungsbereich zu erkennen.

Ebenso kann für eine feste Code-Buch-Größe die Vektordimension variiert und der Einfluß auf Erkennungs- und Kompressionsraten untersucht werden. Abb. 4.15 zeigt die Erkennungsraten für eine Code-Buch-Größe von 256 Einträgen und die Vektordimensionen 2, 3 und 4. Aufgrund der geringen Kombinatorik bei einer Vektordimension von nur zwei Zeit-Bins liegt die Erkennungsrate hier bereits bei kleinen zugelassenen Abweichungen über 60% und geht relativ früh in Sättigung. Für die größeren Vektordimensionen liegen die Raten deutlich niedriger und hohe Raten können hier erst mit großen Maximalabweichungen erzielt werden. Die zugehörigen Werte für die Kompression zeigt Abb. 4.16. Wiederum geben die höheren Erkennungsraten im unteren Bereich der Fehlerskala den Ausschlag für die besseren Kompressionsergebnisse bei kleinerer Vektordimension, während zu hohen Werten hin sich die bessere Kompression durch das Abbilden von mehr ADC Werten auf ein Code-Wort in einer stärkeren Datenreduktion bemerkbar macht.

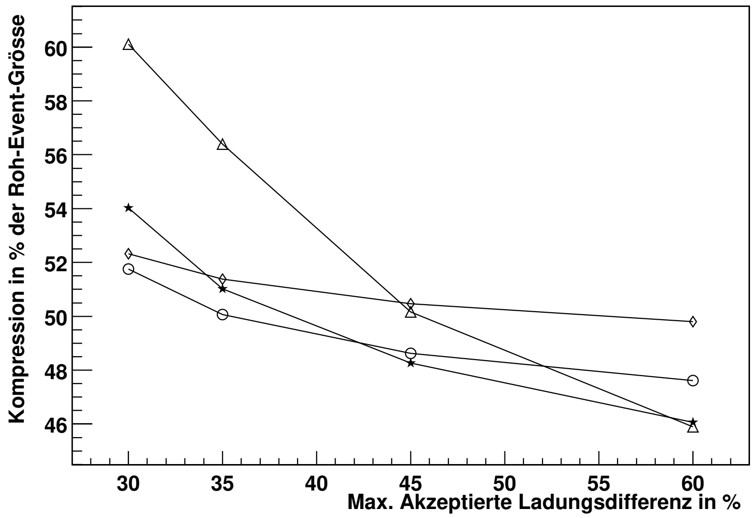


Abbildung 4.14: Ausschnitt aus Abb. 4.13. Code-Buch-Größen: 128 (△), 256 (⋆), 512 (○) und 1024 (◊) Elemente bei einer Vektordimension von 4.

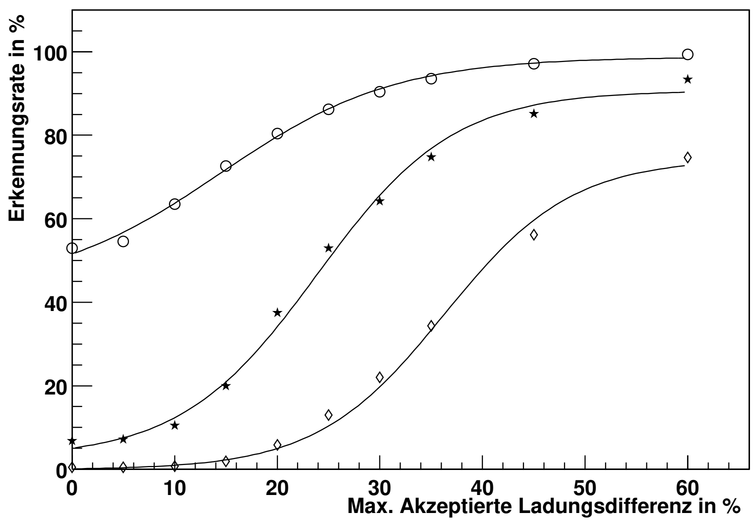


Abbildung 4.15: Erkennungsraten für eine feste Code-Buch-Größe von 256 Einträgen mit Vektordimensionen von 2 (○), 3 (⋆) und 4 (◇).

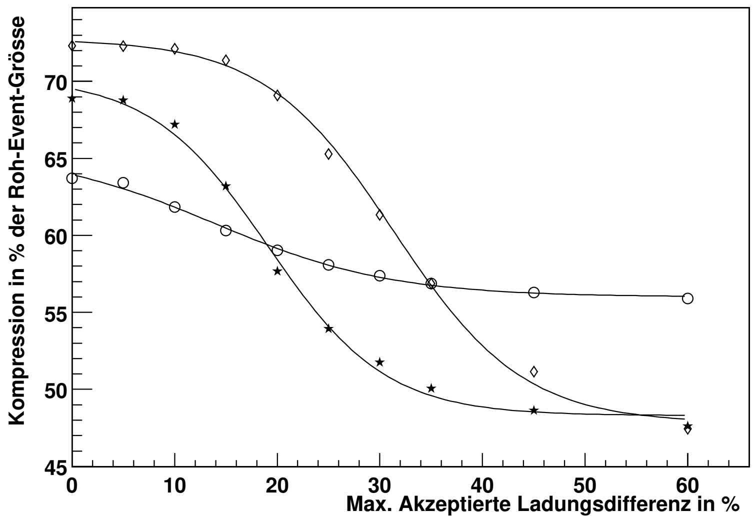


Abbildung 4.16: Kompressionsraten für eine feste Code-Buch-Größe von 256 Einträgen mit Vektordimensionen von 2 ($\circ$), 3 ($\star$) und 4 ($\diamond$).

Die tatsächlichen Fehler in Abhängigkeit von der Größe des Code-Buchs zeigt Abb. 4.17. Die Tatsache, daß dieser bei einem maximalen Fehler von 0% bei etwa 3% liegt, erklärt sich damit, daß eine Abweichung von einem ADC Wert pro Datenpunkt als Rauschen immer zugelassen wurde. Bei wesentlich höheren zugelassenen Fehlern sieht man jedoch, daß der mittlere tatsächliche Fehler deutlich unter dem Maximalwert liegt. Der mittlere Fehler nimmt mit kleiner werdenden Code-Büchern zu. Dies erklärt sich aus der höheren Dichte der Code-Wörter im Eingaberaum bei größeren Code-Büchern und somit geringeren Abständen zwischen Eingabe-Vektor und zugeordnetem Code-Vektor.

Betrachtet man die Abhängigkeit des mittleren Fehlers von der Vektordimension in Abb. 4.18, so findet man innerhalb ihrer jeweiligen Fehler eine Übereinstimmung bei niedrigen akzeptierten Fehlern. Zu höheren Werten hin ist die mittlere Abweichung und ihre Streuung bei kleineren Vektoren kleiner, da sie bei gleicher Code-Buch-Größe wesentlich genauer dargestellt werden können.

Abb. 4.19 zeigt die Verteilung der Absolutfehler für ein 1024 Elemente großes Code-Buch, einer Vektordimension von 4 Zeit-Bins und einem maximal akzeptierten Fehler von 20%. Die leichte Verschiebung des Schwerpunktes zu positiven Werten erklärt sich aus der Verwendung der relativen Abweichung der "Ladungen" von Daten- und Code-Vektor, siehe auch Formel (4.1) von Seite 40: Da auf die Ladung des *Daten*-Vektors normiert wird, ist der zulässige Bereich für Abweichungen für kleinere Code-Wörter größer und somit wird das Integral des Datenvektors eher auf (von der Ladung her gesehen) kleinere Code-Vektoren abgebildet. Hat der Daten-Vektor z.B. die Ladung 10 und ein Code-Vektor die Ladung 11,

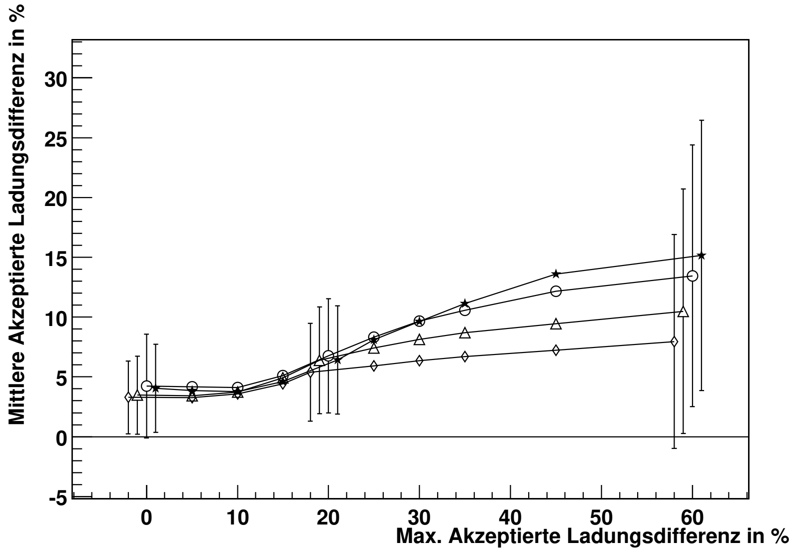


Abbildung 4.17: Mittlere prozentuale Ladungsabweichung für verschiedene Code-Buch-Größen (64 ($\star$), 128 ($\circ$), 256 ($\triangle$) und 512 ($\diamond$)) bei einer Vektordimension von 3. (Um das Diagramm übersichtlich zu halten, wurden nicht für jeden Datenpunkt die Streuung eingetragen, sondern nur für die Datenpunkte bei 0, 20 und 60%. An diesen Stellen wurden die Werte auf der Abszisse auch leicht gegeneinander versetzt, um die einzelnen Fehler sichtbar zu machen.)

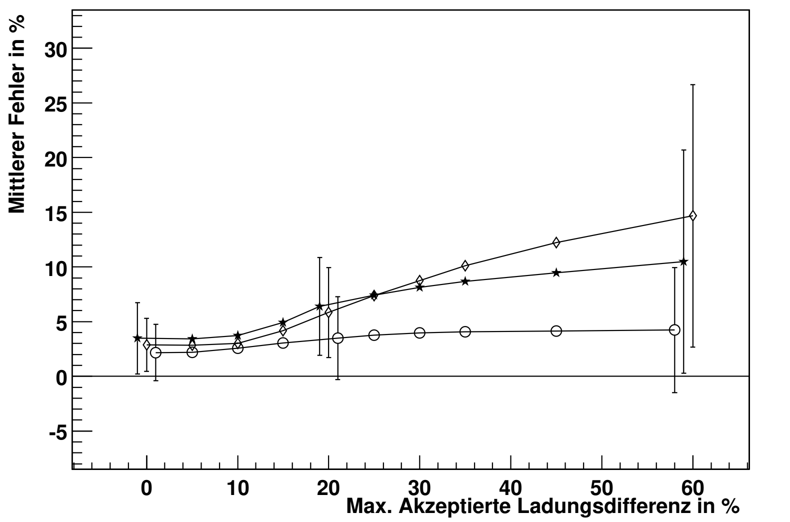


Abbildung 4.18: Tatsächlich akzeptierte prozentuale Ladungsabweichung für verschiedene Vektordimensionen bei einer festen Code-Buch-Größe von 256 Einträgen (Vektordimensionen 2 ($\circ$), 3 ($\star$) und 4 ($\diamond$)).

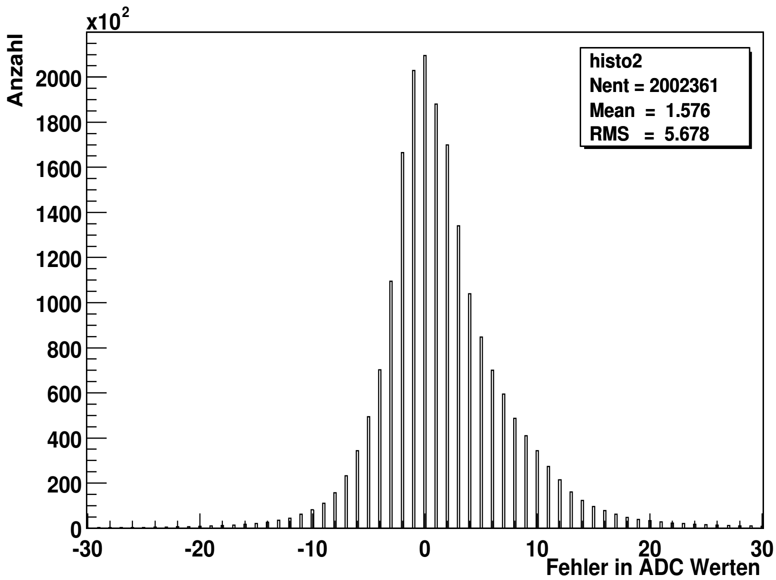


Abbildung 4.19: Verteilung der Absolutfehler für ein Codebuch der Größe 1024, einer Vektordimension von 4 und einem maximal akzeptierten Fehler von 20%.

so beträgt die relative Differenz 10%. Ist die Situation umgekehrt, hat also der Daten-Vektor die Ladung 11 und der Code-Vektor die Ladung 10, beträgt die relative Differenz nur 9%. Die Differenz zwischen Daten- und Code-Vektor ist daher öfter positiv als negativ und die Verteilung zu positiven Werten hin verschoben. Die Normierung auf den *Code*-Vektor würde die Verteilung zu negativen Werten verschieben.

Eine Zusammenstellung der Kompressionsergebnisse verschiedener bereits erwähnter Verfahren, auf die in diesem Kapitel nicht näher eingegangen wird, findet sich in Anhang B.

Kapitel 5

Einfluß auf physikalische Größen

Si quid agis, prudenter agas et respice finem.
Aesop (Anonymus Neveleti), II. 22.

Der entscheidende Punkt bei der *verlustbehafteten* Kompression mittels eines Vektorquantisierers oder anderer Verfahren ist freilich, inwieweit die in den Daten enthaltene "Physik" verändert wird. Zur Beantwortung dieser Frage sollen die Daten für ausgewählte Parametersätze komprimiert und dekomprimiert werden, um sie dann zu analysieren. Die Ergebnisse werden anschließend mit den analysierten Referenzdaten verglichen. Abb. 5.1 zeigt das Vorgehen schematisch.

5.1 Vektorquantisierung

Wie in der Einführung bereits erwähnt, ist die TPC eines der Hauptwerkzeuge zur Spurerkennung und muß im Zusammenwirken mit den anderen Detektoren die Teilchenidentifikation, die über die spezifische Ionisation dE/dx erfolgt, sowie die Impulsmessung mit hinreichend hoher Genauigkeit gewährleisten. Daher ist es besonders wichtig zu untersuchen, welchen Einfluß die verlustbehaftete Kompression auf diese Größen hat.

So werden neben der Zahl der Cluster pro gefundener Spur die Impulsverteilung, die Spurerkennungseffizienz (Tracking Efficiency), die relative Auflösung des Transversalimpulses p_t aller erzeugten Teilchen, die relative Breite der spezifischen Ionisation dE/dx für Pionen im Impulsbereich zwischen 0.4 und 0.5 GeV/c sowie die Ortsauflösung untersucht. Zur Bestimmung dieser Größen wird die im TPC-Package von *aliroot* enthaltene Analyse-Software verwendet.

Für die weitere Untersuchung werden die zwei Parametersätze ausgewählt, welche die vielversprechendsten Kompressionsraten erreichen: die Code-Buch-Größen 64 und 128 bei einer Vektordimension von 2. Abb. 5.2 zeigt die entsprechenden Kompressionsraten. Wäh-

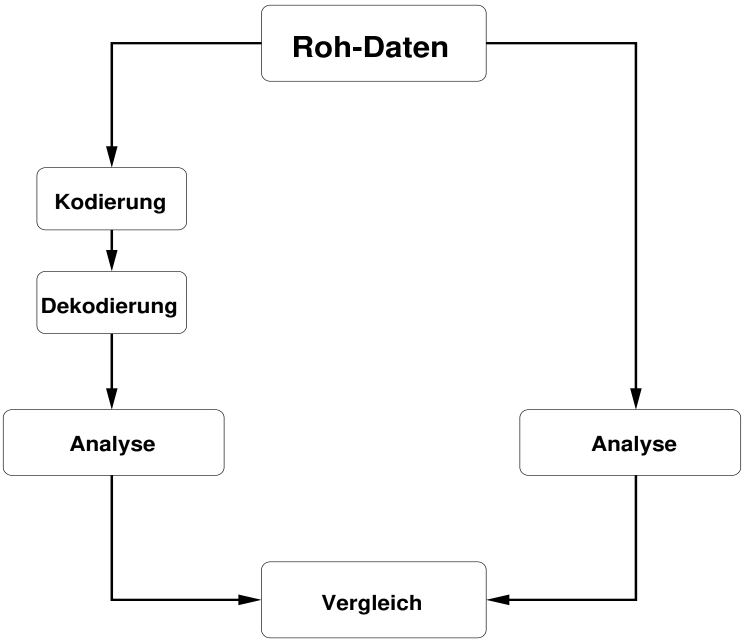


Abbildung 5.1: Schema des Vorgehens zur Untersuchung des Einflusses der verlustbehafteten Kompression auf physikalische Größen.

rend das größere Code-Buch bei maximal akzeptierten Fehlern bis 20% die beste Kompression aller Parametersätze in der Untersuchung erzielt, so gilt dieses für das Code-Buch mit 64 Einträgen im Fehlerbereich von 20 bis 60%.

Vor der Diskussion des Einflusses der angelegten Schwelle und der verlustbehafteten Kompression auf die im einzelnen untersuchten physikalischen Größen soll hier kurz erläutert werden, wie die Analyse-Software vorgeht, um Spuren zu finden.[1]

Bevor die Rekonstruktion von Teilchenspuren beginnt, wird in der Pad-Reihen-Zeit-Ebene nach zweidimensionalen Clustern gesucht, aus denen dann Raumpunkte rekonstruiert werden. Diese Cluster sind beieinander liegende Punkte in der Pad-Reihen-Zeit-Ebene, deren ADC Werte über der Nullunterdrückungsschwelle liegen. Hat ein so gefundener Cluster nur ein lokales Maximum, so wird davon ausgegangen, daß er von nur einer Spur erzeugt wurde. Der Raumpunkt errechnet sich als der Schwerpunkt der ADC Werte. Bei mehreren Maxima wird der Cluster am nächsten Sattelpunkt zerlegt und man erhält mehrere Raumpunkte.[2] Diese Raumpunkte werden interpretiert als der Durchstoßpunkt einer Spur durch die Ebenen der Zentren der Pad-Reihen.

Die Suche nach Teilchenspuren beginnt in den äußersten Pad-Reihen, da hier die Clusterdichte am geringsten ist. Es wird nach Paaren von Raumkoordinaten gesucht, die auf

[1] Die Darstellung hier ist vereinfacht, für eine detailliertere Beschreibung sei auf [4] verwiesen.

[2] Die maximale Anzahl der so aus einem Cluster erhaltenen Raumpunkte beträgt in der verwendeten Version der Analyse-Software 30 Stück. Alle Cluster, die mehr Maxima enthalten, werden verworfen.

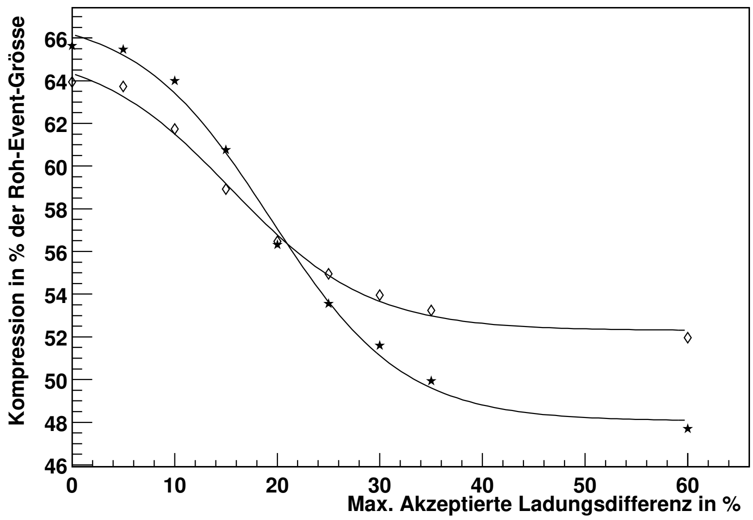


Abbildung 5.2: Kompressionsraten für die beiden ausgewählten Parametersätze der Code-Buch-Größen 64 ($\star$) und 128 ($\diamond$) bei einer Vektordimension von 2.

den Wechselwirkungspunkt zeigen. Sind solche Punkte gefunden, werden aus ihnen als erste Approximation der Teilchenspur die Parameter einer Helix berechnet. Diese Parameter sind die Eingabewert für einen Kalman-Filter, eine in der Hochenergiephysik weit verbreitete wichtige Methode für statistische Schätzungen und Vorhersagen. Dieser Filter bietet die Möglichkeit der simultanen Spurerkennung und -anpassung. Auf diese Weise werden also erste Spurkandidaten gefunden, deren Rekonstruktion dann durch die gesamte TPC fortgesetzt wird: Innerhalb von einem gewissen Toleranzbereich um die berechnete Helix herum wird nach weiteren Clustern gesucht, die dieser Spur zugeordnet werden können und dann aus dem Ereignis entfernt werden. Ist man bei der Rekonstruktion am inneren Radius der TPC angelangt, wird überprüft, ob die rekonstruierte Spur Raumpunkte in mindestens 40% aller Pad-Reihen hat. Ist dies der Fall, so gilt die Spur als "gefunden". Andernfalls wird die Spur verworfen und die zugeordneten Cluster verbleiben in dem Ereignis, um eventuell anderen Kandidaten zugeordnet zu werden.

5.1.1 Clusterzahl und Impulsverteilung

Als erstes wird die Auswirkung der verlustbehafteten Komprimierung auf die Zahl der Cluster pro gefundener Spur und die Impulsverteilung dieser Spuren untersucht.

Die Abbildung 5.3 zeigt die Verteilungen der Cluster pro "guter" Spur, d.h. einer Spur, die mindestens 40% der Pad-Reihen kreuzt, in den Referenzdaten, in den komprimierten und anschließend dekomprimierten Daten sowie zur Verdeutlichung die Differenz dieser beiden

Verteilungen. Die Code-Buch-Größe in diesem Beispiel ist 64, die Vektordimension 2 und die maximal erlaubte Abweichung 60%.

Das abrupte Ansteigen der Verteilungen bei etwa 70 Cluster pro Spur resultiert aus der ausschließlichen Verwendung "guter" Spuren: die Gesamtzahl der Pad-Reihen für innere und äußere Kammern zusammen beträgt 174, wovon 40% etwa 69 sind. Die Verteilung wird durch den Vektorquantisierer selbst bei diesem großen maximal zugelassenem Fehler nur unwesentlich verändert, und so zeigt die Differenz ein rauschähnliches Verhalten.

Die Impulsverteilungen in Abb. 5.4 der gefundenen guten Spuren zeigen ebenfalls keinen systematischen Einfluß der Kompression.

5.1.2 Spurerkennungseffizienz

Die Spurerkennungseffizienz wird bestimmt als der prozentuale Anteil der "gefundenen guten" Spuren, d.h. der Spuren, die mindestens 40% der Pad-Reihen kreuzen und bei denen weniger als 10% der ihnen zugeordneten Cluster nicht zu der jeweiligen Spur gehören, an der Gesamtzahl der "guten" Spuren in den unveränderten Daten. Die "Falschen Spuren" (Fake Tracks) hingegen sind Spuren, die zwar eine ausreichende Zahl an Clustern haben, denen jedoch mehr als 10% davon fälschlicherweise zugeordnet wurden.

Die Abbildungen 5.5 und 5.6 zeigen, wie sich die Spurerkennungseffizienz und der Anteil der Falschen Spuren unter dem Einfluß der Kompression verhält. Die Spurerkennungseffizienzen in den komprimierten Daten und in den Referenz-Daten stimmen innerhalb von 0.5% überein. Die Abweichung von der Effizienz in den Originaldaten (mit 45-Grad-Konus und ohne angelegte Schwelle) beträgt etwa 1%. Der Rückgang der Spurerkennungseffizienz ist also weniger auf den Verlust von Information durch die Kompression als vielmehr durch das Datenformat zurückzuführen und somit die gewählte Schwelle zurückzuführen. Die Spurerkennungseffizienz wird also durch die verlustbehaftete Kompression nur unwesentlich beeinflußt.

Ebenfalls um weniger als 0.5% schwankt die Zahl der "Falschen Spuren" um den Wert in den Originaldaten. So hat auch hier der Vektorquantisierer einen zu vernachlässigenden Einfluß.

5.1.3 Transversalimpulsauflösung

Die Bestimmung der relativen Abweichung der Impulse geschieht mittels

$$\sigma_{p_t} = \frac{p_t - p_{t,MC}}{p_{t,MC}}$$

$p_{t,MC}$ bezeichnet hierin den Transversalimpuls der Spur in den Monte-Carlo-Daten, während p_t den Transversalimpuls der gefundenen Spur kennzeichnet. Die Verteilung dieser

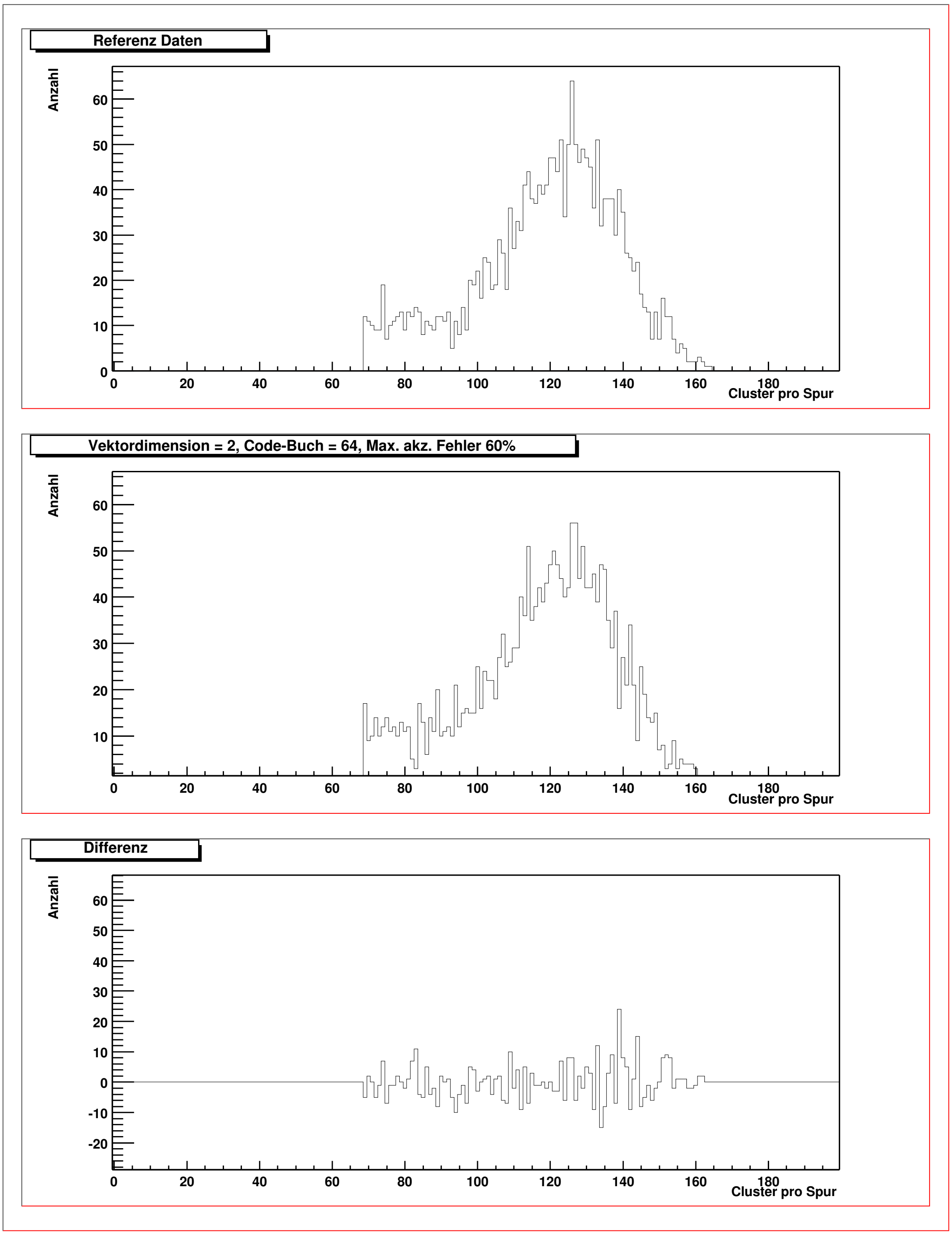


Abbildung 5.3: Verteilung der Cluster pro Spur in den Referenzdaten, bei einem Code-Buch mit 64 Einträgen und einer Vektordimension von 2 sowie deren Differenz.

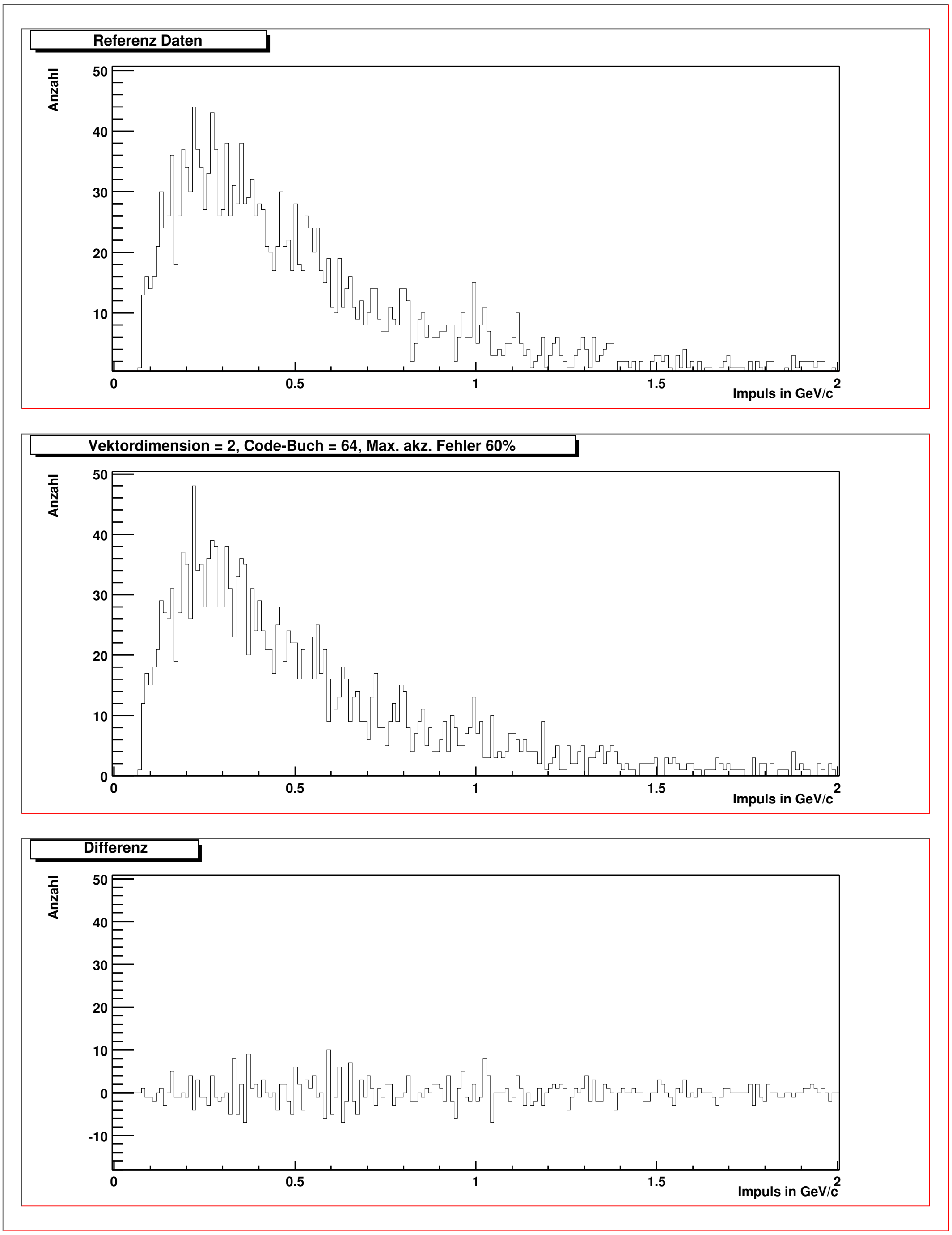


Abbildung 5.4: Impulsverteilung in den Referenzdaten bei einem Code-Buch mit 64 Einträgen und einer Vektordimension von 2 sowie deren Differenz.

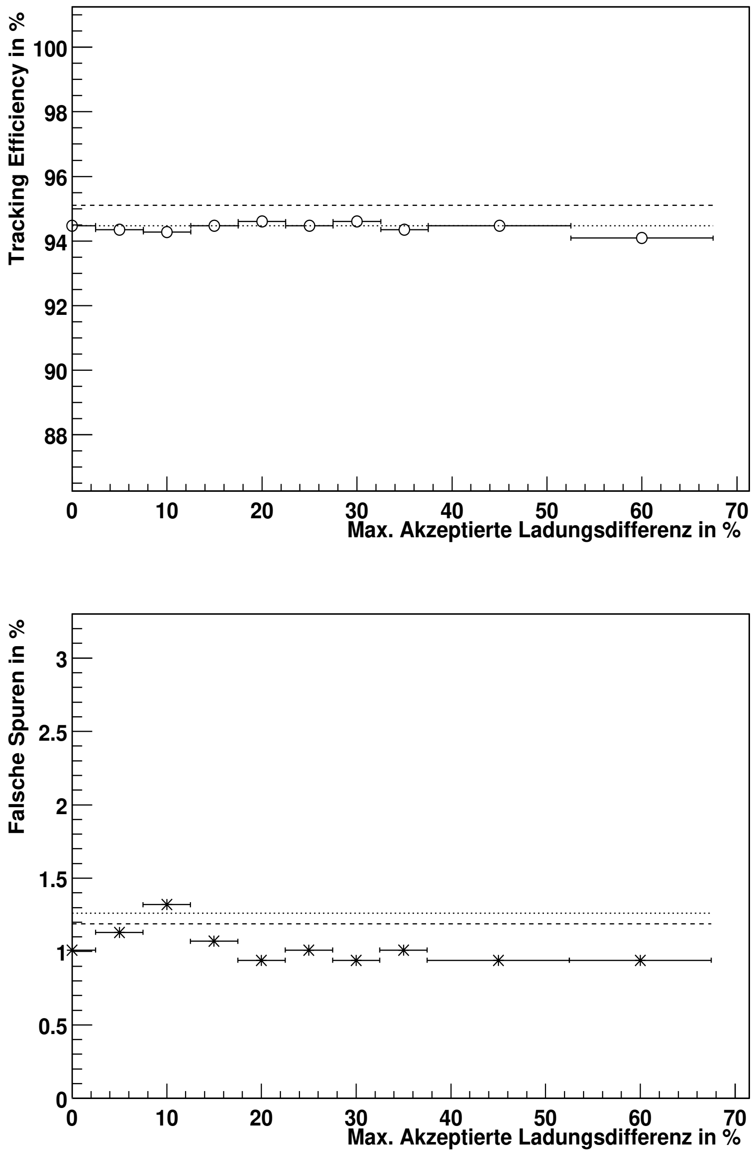


Abbildung 5.5: Spurerkennungseffizienz und Falsche Spuren für ein Code-Buch der Größe 64 und der Vektordimension 2. Ebenfalls eingezeichnet ist die Tracking Efficiency und die Zahl der Falschen Spuren für die Referenzdaten (···) sowie für die Daten ohne Anlegung einer Schwelle und inklusive des 45-Grad-Konus (- - -).

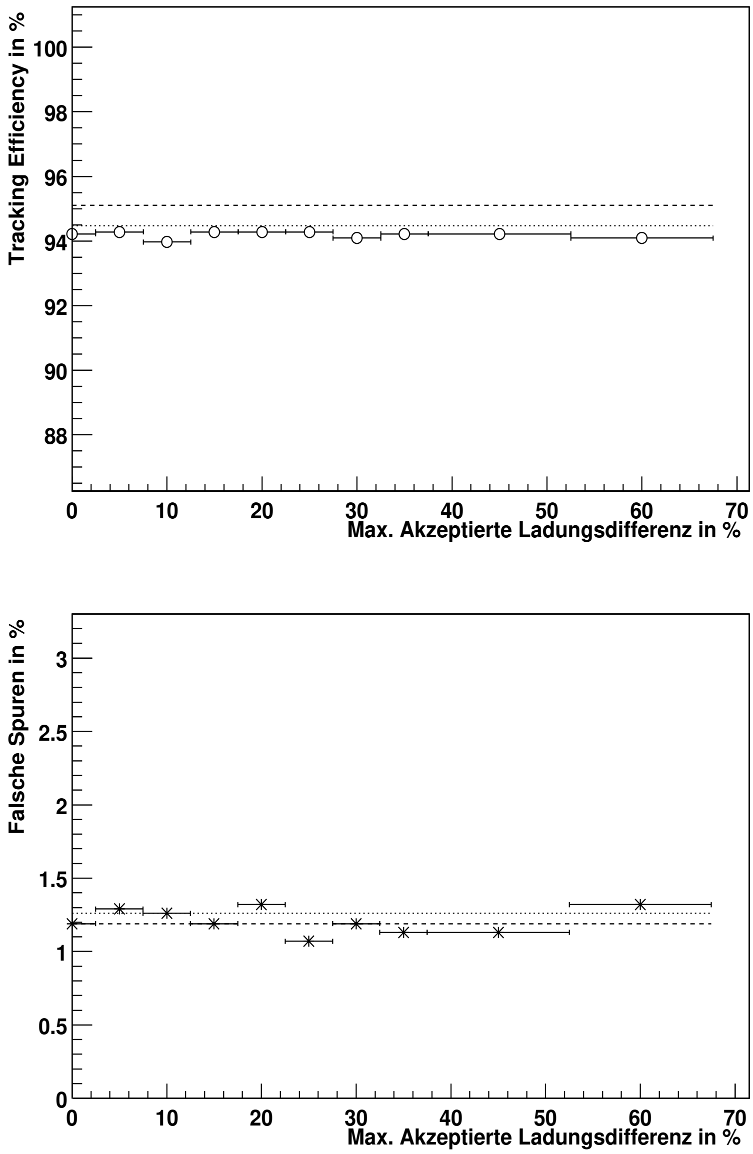


Abbildung 5.6: Spurerkennungseffizienz und Falsche Spuren für ein Code-Buch der Größe 128 und der Vektordimension 2. Ebenfalls eingezeichnet ist die Tracking Efficiency und die Zahl der Falschen Spuren für die Referenzdaten (···) sowie für die Daten ohne Anlegung einer Schwelle und inklusive des 45-Grad-Konus (- - -).

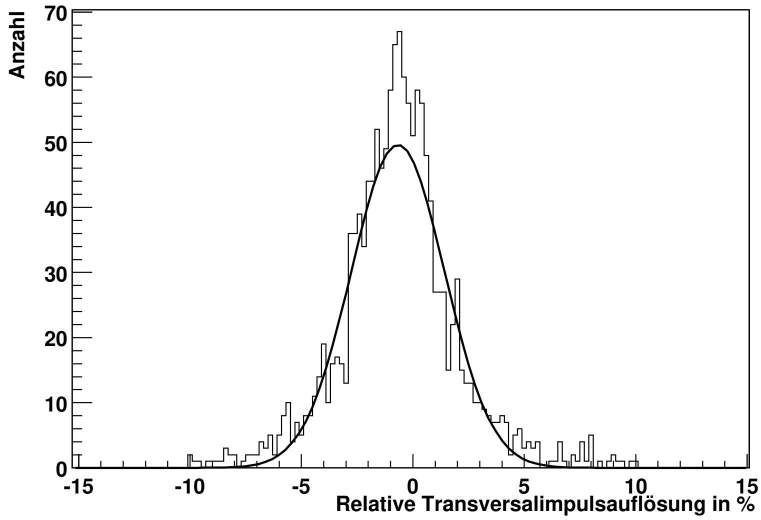


Abbildung 5.7: Verteilung der relativen Transversalimpulsauflösungen in den Originaldaten, aufintegriert über η sowie über alle Impulse.

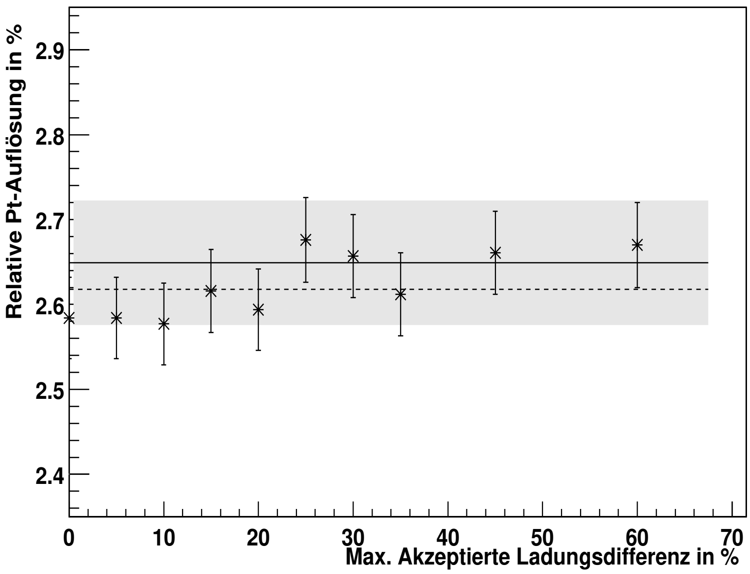
Abweichungen in den Originaldaten ist in Abb. 5.7 zu sehen. Offensichtlich hat diese Verteilung kein rein gaußförmiges Profil, so daß zur Bestimmung der Breite keine Anpassung einer Normalverteilung, sondern die mittlere quadratische Abweichung (RMS) herangezogen wird. Alle Werte über 10% werden abgeschnitten, um einen robusten Wert für die Breite der Verteilung zu erhalten. Durch diesen Schnitt wird verhindert, daß einzelne Datenpunkte, die weit vom Mittelwert abweichen, die ermittelte mittlere quadratische Abweichung dominieren.

Um die Transversalimpulsaufösung in Abhängigkeit von dem Transversalimpuls der Spur zu bestimmen, wie es Gleichung 2.2 von Seite 19 nahelegt, reicht die Statistik in dem verwendeten Ereignis nicht aus. Aus diesem Grunde wird über alle Spuren aufsummiert.

Die Breite der relativen Transversalimpulsauflösung zeigt die Abb. 5.8 in Abhängigkeit von der maximal akzeptierten Abweichung.

Der ermittelte Wert für die Originaldaten von etwa 2.65% liegt über dem Wert für σ_{p_t}/p_t aus [4] von 2.1%. Wie bereits erwähnt, hängt der ermittelte Wert allerdings empfindlich von der Art der Bestimmung der Breite ab. So konnte mit der Anpassung einer Gaußkurve an die Daten ebenfalls eine Breite von 2.1% ermittelt werden. Wie Abb. 5.7 zeigt, ist die Übereinstimmung zwischen Modell und Daten jedoch unbefriedigend.

Über den gesamten Bereich hinweg ist diese Auflösung mit der in den Originaldaten verträglich. Die Tendenz zu einer etwas besseren Auflösung in den bearbeiteten Daten hat seine Ursache in der Vorgehensweise der Analyse-Software bei der Spursuche: Durch das Verwerfen von Clustern, die mehr als 30 Maxima haben, gehen einige Koordinaten-Punkte



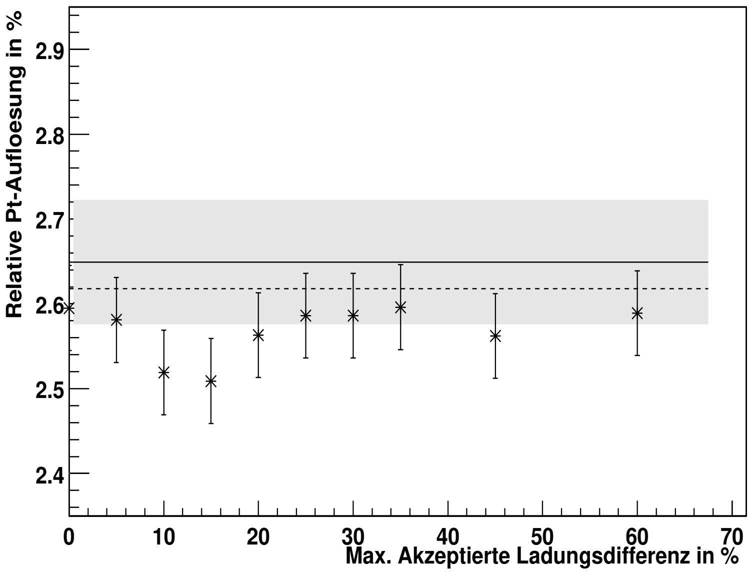


Abbildung 5.8: Breite der relativen p_t - Auflösung in % bei einer Code-Buch-Größe von 64 (oben) bzw. 128 (unten) Einträgen und einer Vektordimension von 2 gemittelt über η sowie über alle Impulse. Ebenfalls eingezeichnet ist der Wert der Originaldaten (——) mit seinem Fehler (grauer Balken) sowie der Wert bei den Referenzdaten mit einer angelegten Schwelle bei einem ADC Wert von 10 (- - -).

mancher Spuren verloren. Durch das Datenformat werden die großen Cluster in mehrere kleinere zerlegt, so daß die Anzahl der Maxima pro Cluster sinkt. Dadurch bleiben diese Raumpunkte erhalten. Große Cluster werden vor allem bei kleineren Radien auftreten und so werden insbesondere hier neue Koordinaten dazukommen. Dies kann eine Spur verlängern und so zu einer besseren Auflösung beitragen.

Insgesamt ist also kein entscheidender Einfluß des Datenformats oder der Kompression auf die Impulsauflösung festzustellen.

5.1.4 Die spezifische Ionisation

Eine wohl sehr empfindliche Größe sollte die durch die ADC Werte angenäherte Ladung eines Pulses sein, da für sie explizit ein gewisser Toleranzbereich eingeräumt wird. Somit wird eine Größe, die am ehesten nachteilig durch die Kompression beeinflußt wird, die spezifische Ionisation dE/dx sein. Die Abb. 5.9 zeigt die relative Breite der spezifischen Ionisation für die ausgewählten Parameter.

Die ermittelten Werte sind innerhalb ihrer Fehler mit den Werten der Referenz- und der Originaldaten verträglich und zeigen keinen signifikanten Einfluß der Vektorquantisierung. Die Ergebnisse der mit dem Vektorquantisierer bearbeiteten Daten liegen ab maximal tolerierten Fehlern von etwa 20% konstant über demjenigen der Referenzdaten. Dies hat seine Ursache in der bevorzugten Wahl von Code-Vektoren mit kleinerer Ladung, wie in Abschnitt 4.3.3 beschrieben.

Die Auflösung bei der spezifischen Ionisation von $\sim$11% liegt nahe bei der in [4] angegebenen von 10%.

5.1.5 Ortsauflösung

5.1.5.1 Hohe Multiplizität

Zur Bestimmung der Ortsauflösung werden die transversalen und longitudinalen Residuen[3] herangezogen, also die Abstände der Raumpunkte in einer Pad-Reihe von dem aus den Spurparametern berechneten Durchstoßpunkt durch die entsprechende Ebene in Pad- und Drift-Richtung. Das Vorgehen erläutert Abb. 5.10.

Beim hier verwendeten Datenformat werden alle ADC Werte unterhalb der Schwelle von 10 weggeschnitten, nachdem eine Vor- und Nachgeschichte von jeweils einem ADC Wert in Zeitrichtung gespeichert wurde. Somit gehen in Pad-Richtung (azimuthal) eventuell die Ausläufer eines Clusters verloren und verfälschen seine Schwerpunkts- und somit seine Ortskoordinaten.

[3] Zur Berechnung der Residuen wurde Quelltext von Anders Vestbø, Universität Bergen/Norwegen, verwendet.

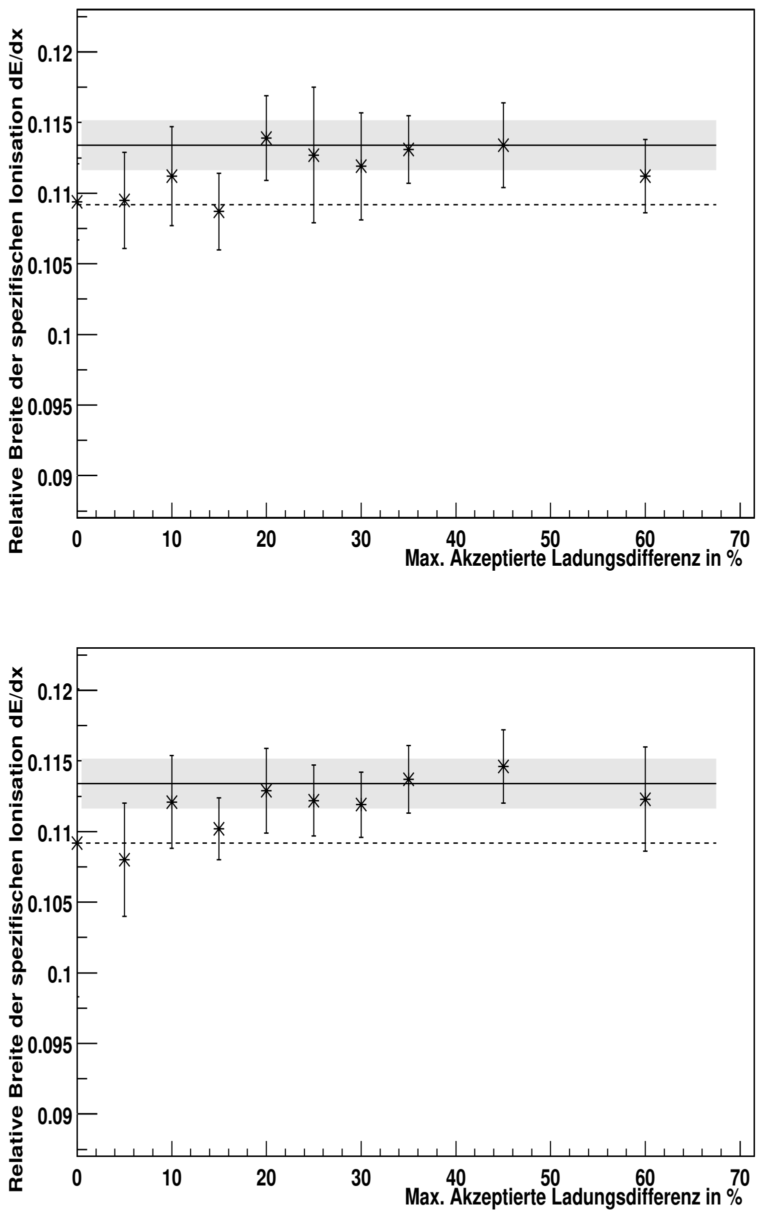


Abbildung 5.9: Relative Breite der spezifischen Ionisation dE/dx für eine Code-Buch-Größe von 64 Einträgen und einer Vektordimension von 2. Ebenfalls eingetragen ist der Wert für die Referenz-Daten (- - -) sowie der Wert aus den Originaldaten (——) mit seinem Fehler (grauer Bereich).

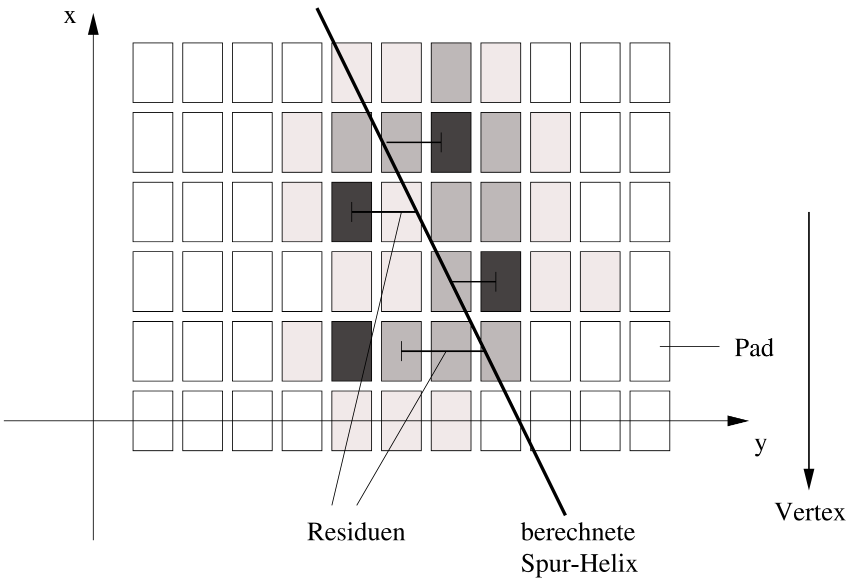


Abbildung 5.10: Zur Erläuterung der Berechnung der Residuen. In einer gegebenen Pad-Reihe wird der Helix-Durchstoßpunkt einer Spur durch das Zentrum der Pad-Reihe - Zeit - Ebene berechnet und mit den aus den Schwerpunkten bestimmten Raum-Koordinaten des zugeordneten Clusters in der gleichen Pad-Reihe in Pad-Richtung (transversal) und Drift-Richtung (longitudinal) verglichen. Die Abbildung zeigt das Vorgehen exemplarisch für die transversalen Residuen. Die unterschiedliche Färbung einzelner Pads deutet die auf ihnen deponierte Ladung an.

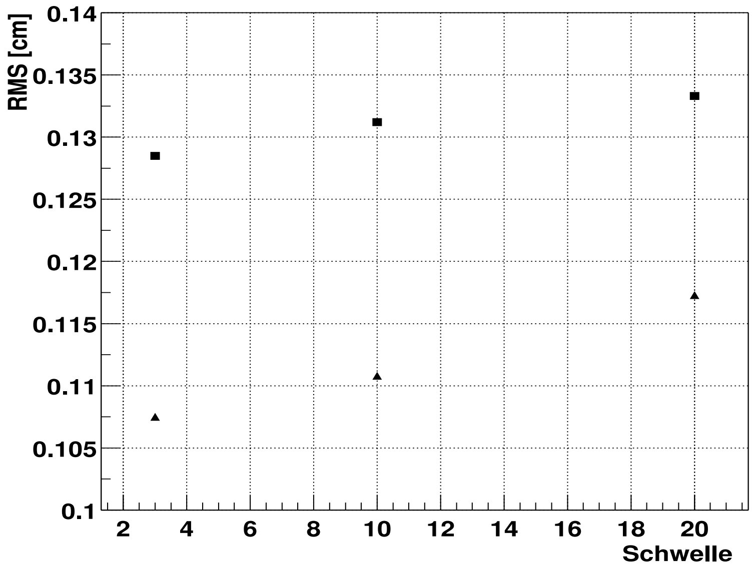


Abbildung 5.11: Einfluß der Schwelle auf die Ortsauflösung in transversaler Richtung bei einem Ereignis mit voller Multiplizität. Die Dreiecke sind die Werte für die äußeren, die Quadrate diejenigen für die inneren Kammern. Dieses Diagramm wurde freundlicherweise von [39] zur Verfügung gestellt.

Aufgrund der hohen Multiplizität allerdings überlappen insbesondere in den inneren Kammern viele der Cluster, so daß dieser Effekt hier bei der Bestimmung der Raumpunktkoordinaten weniger stark ein Problem darstellt als der Überlapp und die schwierige Trennung der Cluster selbst. Alle im folgenden gezeigten Ergebnisse hängen somit von den Eigenschaften der Analyse-Software und insbesondere dem Cluster-Finder ab. An dieser Stelle sei auch angemerkt, daß durch die Art der Bestimmung der Ortsauflösung aus den Residuen, z.B. über die mittlere quadratische Abweichung oder mittels der Anpassung einer Gaußkurve, das Ergebnis empfindlich beeinflußt werden kann.

Den Einfluß der Schwelle auf die Ortsauflösung zeigt Abb. 5.11. Hier wird die Auflösung aus der mittleren quadratischen Abweichung der Residuen bestimmt, wobei alle Werte kleiner -0.5 cm oder größer 0.5 cm abgeschnitten wurden, damit das Ergebnis nicht von wenigen Werte, die sehr weit vom Mittelwert entfernt liegen, dominiert wird. Ermittelt werden hier relative Verschlechterungen der Ortsauflösung durch eine Schwelle bei einem ADC Wert von 10 von etwa 2% (innen) und 3.5% (außen).

Die Abbildungen 5.12 und 5.13 zeigen die Verteilung der transversalen Residuen für die Originaldaten, getrennt nach inneren und äußeren Kammern für Spuren mit einem Impuls von mehr als 1 GeV/c und mehr als 100 Clustern. Spuren mit hohem Impuls unterliegen weniger dem Einfluß der Vielfach-Streuung, die die Ortsauflösung verschlechtert. Der Schnitt auf der Clusterzahl stellt sicher, daß nur lange Spuren zur Bestimmung der Ortsauflösung

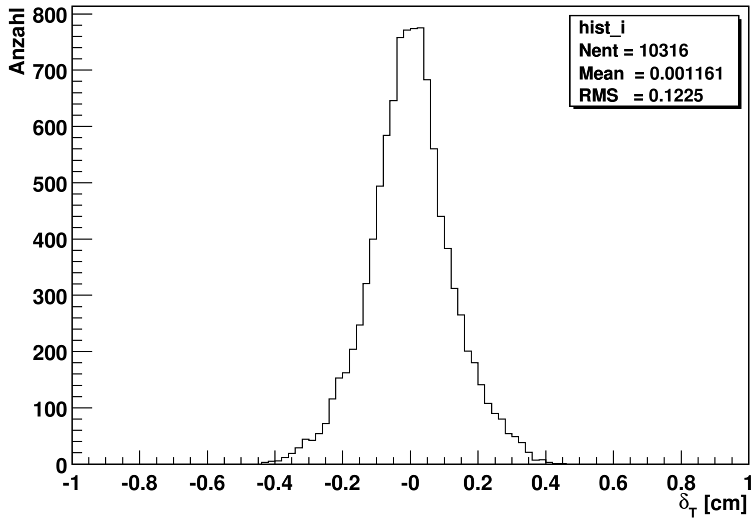


Abbildung 5.12: Verteilung der transversalen Residuen für die inneren Kammern. Die verwendeten Spuren haben Impulse $\geq$ 1 GeV/c und mehr als 100 ihnen zugeordnete Cluster.

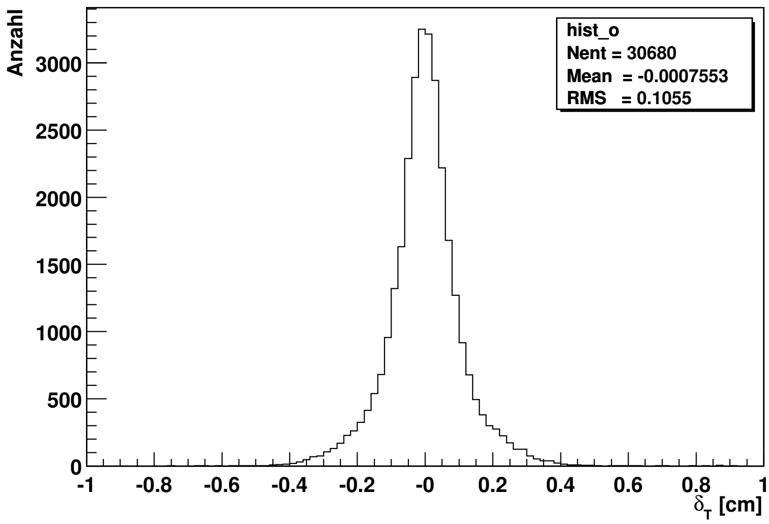


Abbildung 5.13: Verteilung der transversalen Residuen für die äußeren Kammern. Die verwendeten Spuren haben Impulse $\geq$ 1 GeV/c und mehr als 100 ihnen zugeordnete Cluster.

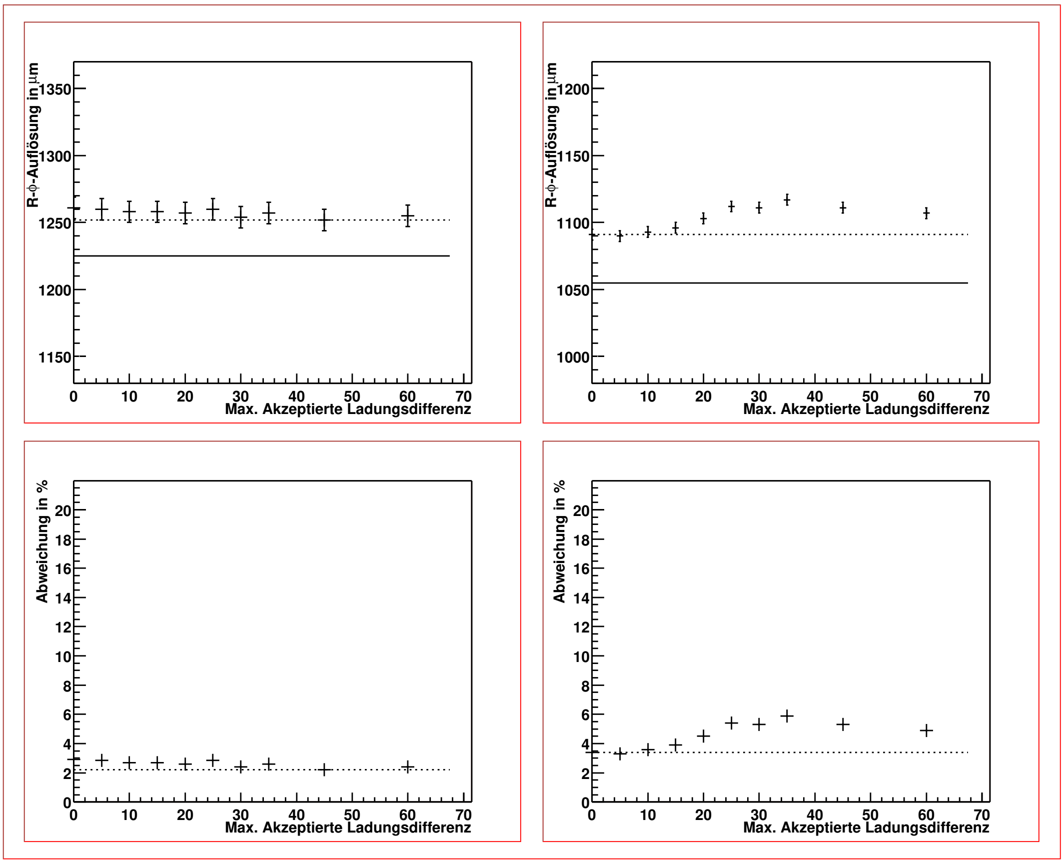


Abbildung 5.14: Abhängigkeit der Ortsauflösung von der maximal akzeptierten Ladungsdifferenz für die inneren (linke Spalte) und die äußeren Kammern (rechte Spalte) für ein Ereignis mit voller Multiplizität. Aufgetragen ist in den oberen Diagrammen die absolute Auflösung, in den unteren die relative Abweichung $\frac{\sigma_{Schwelle}-\sigma_{Original}}{\sigma_{Original}}$ in %. Die durchgezogene Linie kennzeichnet den Wert der Originaldaten, die gepunktete diejenige für die Daten mit der Schwelle.

herangezogen werden.

Da es in den äußeren Kammern zwei verschiedene Pad-Größen und somit unterschiedliche intrinsische Auflösungen gibt, ist die Verteilung nicht normal. Zur Breitenbestimmung wird deshalb wie schon bei der Impulsbestimmung die mittlere quadratische Abweichung herangezogen. Aus Konsistenzgründen wird die Breite auch für die inneren Kammern auf diese Weise bestimmt.

In der Abb. 5.14 ist die aus den transversalen Residuen bestimmte Ortsauflösung gezeigt.[4] Die Auflösung in den Originaldaten liegt bei 1225 μm für die inneren und 1055 μm für die äußeren Kammern.[5] Nach Anlegung der Schwelle und Bearbeitung durch den Vektorquantisierer verschlechtert sich die Auflösung um etwa 30 μm (innen) bzw. 40 μm (außen). Die relative Abweichung liegt somit zwischen 3% (innen) und 6% (außen). Die relative Verschlechterung allein durch die Anlegung einer Schwelle deckt sich mit den oben bereits erwähnten, siehe Abb. 5.11.

Wenn die Verschlechterung der Auflösung durch die Schwelle und die Kompression als unabhängig angenommen wird, addieren sich die Varianzen quadratisch mit der intrinsischen Auflösung zur Gesamtauflösung:

$$\sigma_{total}^2 = \sigma_{intrinsisch}^2 + \sigma_{Schwelle}^2 + \sigma_{Kompression}^2 \tag{5.1}$$

Aus der Messung der Residuen *ohne* die Anwendung der Kompression erhält man für $\sigma_{Schwelle}$ einen Wert von etwa 260 μm (innen) bzw. 280 μm (außen). Die Berechnung von $\sigma_{Kompression}$ ergibt dann mit diesem Wert für $\sigma_{Schwelle}$ bei einer maximal zulässigen Ladungsdifferenz von 20%, was einer Kompressionsrate von 55% entspricht, einen Wert von 160 μm für die äußeren Kammern. In den inneren Kammern stimmen die Auflösungen innerhalb der Fehlergrenzen überein.

Bei der Auflösung in Drift-Richtung sind die Effekte nicht ganz so stark. Hier liegen die relativen Abweichung innen und außen unter 2%. In diese Richtung werden bei der Anlegung der Schwelle ja eine Vor- und Nachgeschichte von einem Zeit-Bin berücksichtigt, und somit die Ausläufer eines Clusters nicht so rigoros weggeschnitten.

Die Ortsauflösung ist also eine Größe, die durch das Datenformat und den Vektorquantisierer sichtbar nachteilig beeinflußt wird, auch wenn dieser Effekt nur klein ist.

---

[4] Die Ortsauflösungen der Daten, die mit einem Code-Buch mit 64 Einträgen behandelt wurden, zeigen nichts grundsätzlich Neues und werden aus diesem Grunde weggelassen.

[5] In [4] sind Werte von 1150 μm und 800-850 μm angegeben, allerdings für minimal ionisierende Teilchen. Für gut getrennte Spuren, d.h. für eine geringe Multiplizität, werden Werte in Pad-Richtung von 870 μm (innen) bzw. 720 μm (außen) gemittelt über den Impuls der Spuren genannt. Diese Werte in etwa verdoppelt ergeben eine realistische Abschätzung für die Auflösung [15].

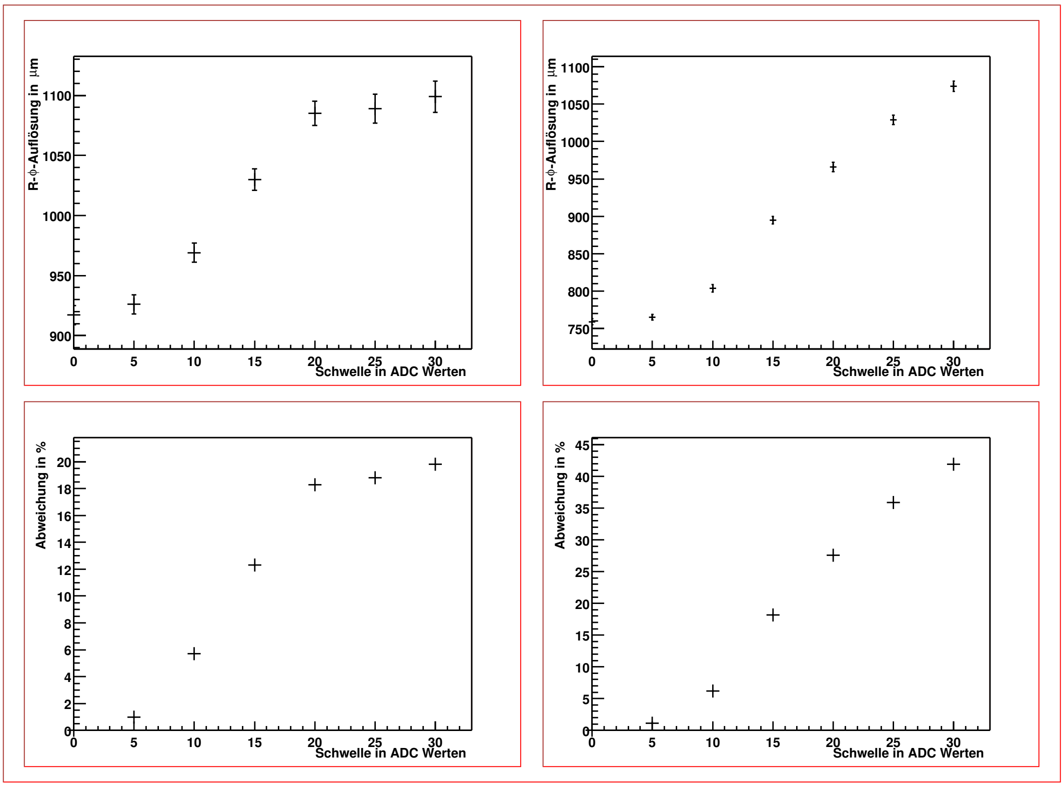


Abbildung 5.15: Abhängigkeit der Ortsauflösung von der Anlegung einer Schwelle in einem Ereignis mit geringer Multiplizität für die inneren (linke Spalte) und die äußeren Kammern (rechte Spalte). Aufgetragen ist in den oberen Diagrammen die absolute Auflösung, in den unteren die relative Abweichung $\frac{\sigma_{Schwelle}-\sigma_{Original}}{\sigma_{Original}}$ in %.

5.1.5.2 Geringe Multiplizität

Der geringe Einfluß der verlustbehafteten Kompression und auch der Schwelle auf die Ortsauflösung in den inneren Kammern werden auf die hohe Multiplizität und die damit verbundene hohe Clusterdichte zurückgeführt: Die verschiedenen Cluster überlappen und die Ausläufer eines Cluster laufen bereits in den nächsten hinein. Der Clusterfinder trennt solche Übergänge am Sattelpunkt ohne Entfaltung. Somit entfallen durch die Vorgehensweise kleinere ADC Werte an den Rändern eines Clusters für die Schwerpunkts- und somit die Raumpunktbestimmung.

Um dies zu verifizieren, werden verschiedene Schwellen auf ein Ereignis mit einer Multiplizität von $dN_{charged}/d\eta$=1000 (im Gegensatz zu $dN_{charged}/d\eta$=8000 bisher) angewendet. Immer wird bei der Anwendung einer Schwelle eine Vor- und Nachgeschichte von einem ADC Wert in Zeitrichtung abgespeichert.

Abb. 5.15 zeigt den Einfluß auf die Ortsauflösung in Abhängigkeit von der angelegten Schwelle.

Die hier wiederum mittels der mittleren quadratischen Abweichung zu 917 μm (innen) bzw. 757 μm (außen) bestimmten Auflösungen sind verträglich mit den in [4] angegebenen von 870 μm (innen) und 720 μm (außen).

Hier zeigt sich deutlich, wie empfindlich die ermittelten Werte von der verwendeten Technik abhängen: Nimmt man statt der mittleren quadratischen Abweichung eine Gauß-Anpassung vor, so erhält man Auflösungen von 836 μm (innen) und 621 μm (außen), also stark abweichende Werte.

Für eine Schwelle bei einem ADC Wert von 10, wie sie auch auf das Ereignis mit hoher Multiplizität angewendet wird, liegt die relative Verschlechterung der Ortsauflösung bei 6%. Dies entspricht dem Wert bei hoher Multiplizität in den äußeren Kammern, wo die Cluster besser voneinander getrennt sind. Für größere Schwellen nimmt der Fehler in der Ortsauflösung rasch zu.

Dies zeigt, daß die Verschlechterung in der Ortsauflösung von lediglich 3% in dem Event mit der hohen Multiplizität seine Ursache offensichtlich im Überlapp der Cluster durch die hohe Spurdichte findet.

5.1.5.3 Keine Schwelle

Wie aus Abb. 5.14 ersichtlich ist, hat die Kompression mittels des Vektorquantisierers zumindest bei kleineren zugelassenen Ladungsabweichungen auf den Daten mit einer Schwelle kaum einen Einfluß auf die Ortsauflösung: Die Auflösungen stimmen innerhalb ihrer Fehler mit denen des Referenzformats überein.

Um den Einfluß des Vektorquantisierers auf die Originaldaten zu untersuchen, wird die Kompression auf die Daten bei einer Schwelle bei einem ADC Wert von 3 untersucht. Diese Schwelle läßt die Daten unverändert, da bereits durch die Nullunterdrückung die ADC Werte 1 und 2 abgeschnitten werden. Es wird wie bisher ein Codebuch mit 128 Einträgen bei einer Vektordimension von 2 verwendet, das mit den "neuen" Daten trainiert wird.

Durch die niedrigere Schwelle werden mehr Cluster erhalten und so vergrößern mehr Zusatzinformation zu deren Position und Länge das Datenvolumen. Durch die Anwendung der Kompression kann das Datenvolumen bei einer maximal tolerierten Ladungsdifferenz von 20% lediglich auf 64% reduziert werden, im Vergleich zu 55% mit der Schwelle bei einem ADC Wert von 10.

Die Ortsauflösung in den inneren Kammern zeigt sich durch den Vektorquantisierer - wie schon bei den Daten mit Schwelle - bei maximal tolerierten Ladungsabweichungen von 0 und 20 % unbeeinflußt, außen verschlechtert sich die Auflösung in transversaler Richtung um 20 μm bzw. knapp 30 μm, was Relativwerten von knapp 2% und 2.5% entspricht. Somit ist der Einfluß in den äußeren Kammern durch den Vektorquantisierer etwas größer als bei den Daten, die mit Schwelle und Vektorquantisierer bearbeitet werden. Dies wird allerdings durch die Schwankung der Ortsauflösung mit verschiedenen zugelassenen Fehlern des Vektorquantisierers relativiert. So erhält man für einen maximal erlaubten Fehler von 30% lediglich eine Abweichung von weniger als 18 μm, was einem Relativwert von etwas über 1.5% entspricht.

In longitudinaler Richtung liegen die ermittelten Auflösungen der Originaldaten bei 2085 μm (innen) und 1730 μm (außen). Diese Werte sind gemittelt über η.[6] Durch die Kompression wird die Auflösung in den inneren Kammern nicht beeinflußt, außen verschlechtert sie sich um 1%.

Insgesamt bleibt der Einfluß des Vektorquantisierers auf die Ortsauflösung meßbar, aber klein.

[6] Zur genaueren Untersuchung muß auf den Winkel zwischen Strahlrohr und Spur geschnitten werden, da nur die Elektronenwolken von Spuren nahezu senkrecht zum Strahlrohr in etwa die gleiche Strecke zu den Endkappen zurücklegen müssen und die Verbreiterung durch Diffusionsprozesse vergleichbar ist. Bei Spuren mit kleineren Winkeln hingegen ändert sich die Auflösung in longitudinaler Richtung aufgrund der unterschiedlichen Driftlänge innerhalb derselben Spur. In transversaler Richtung ist wegen der parallelen Ausrichtung von E- und B-Feld die Diffusion stärker unterdrückt. Da der Einfluß der Vektorquantisierung ohne Schwelle auf die Auflösung in longitudinaler Richtung sich ansonsten nicht grundlegend von der in transversaler Richtung unterscheidet, wird hier auf eine tiefergehende Untersuchung verzichtet und es werden nur die gemittelten Werte angegeben.

5.2 Videokompression

Wie die Vektorquantisierung gezeigt hat, ist die in den Daten enthaltene Physik relativ unempfindlich gegen kleine rauschähnliche Veränderungen der ADC Werte. Diese Unempfindlichkeit und die Tatsache, daß die Daten in 8 Bit vorliegen und somit gut auf 256 Graustufen abzubilden sind, liefern die Argumente für den Test verlustbehafteter Bildkompressionsverfahren, die i.a. sehr hohe Kompressionsraten erzielen. Da jedes der Kathoden-Pads in 445 *Zeit*-Schritten ausgelesen wird, die Cluster sich über mehrere Pads und Zeitschritte erstrecken und somit die Einzelbilder stark korreliert sind, bietet sich die Transformation der Daten zu einer Bildfolge und die Verwendung eines Videokompressionsverfahrens - aufgrund der guten Kompressioneigenschaften wird hier MPEG gewählt - an, welches dann die Korrelation zur Kompression ausnutzt.

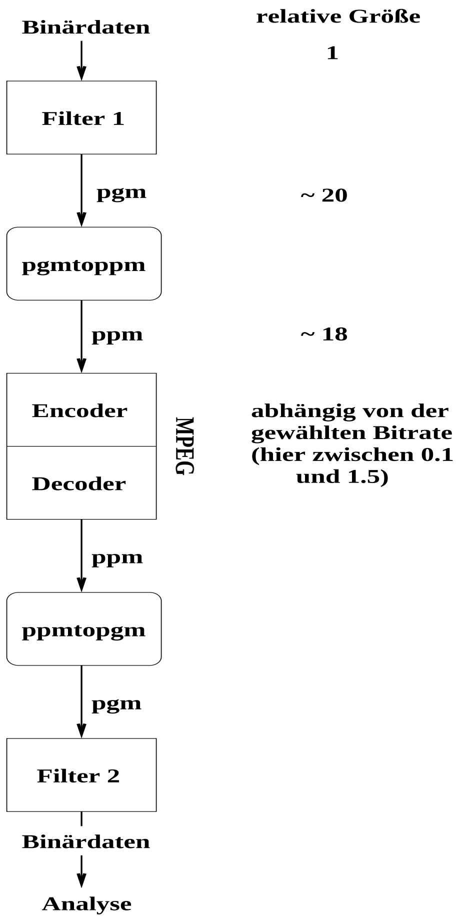


Abbildung 5.16: Datenfluß beim Test der MPEG-Codierung. Da die pgm-Dateien der Einfachheit halber im ASCII-Format (Magic Number P2) gespeichert werden, ist ihr Datenvolumen größer als das der ppm-Dateien, die nach der Konvertierung im Roh-Format (P6) vorliegen.

Da die Daten der TPC kammerweise vorliegen, werden sie zunächst in Bilder von 160x176 Pixel im pgm[7]-Format konvertiert, in denen dann jeweils eine Zeitscheibe einer inneren und der zugehörigen äußeren Kammern abgebildet ist.[8] Durch die gewählten Dimensionen läßt sich ein Bild leicht in Makroblöcke zerlegen. Zudem stimmt die Höhe in etwa mit der Anzahl der Pad-Reihen in einem Sektor, 55 innen und 119 außen, überein. Um die Daten dem Format anzupassen, wird zwischen den Sektoren sowie am oberen Bildrand jeweils eine zusätzliche Zeile eingefügt. Bereiche, in denen keine Pads sind und die etwa 37% des Bildes ausmachen, werden schwarz gefärbt. Diese Regionen sollten allerdings bei der Kompression, da sie in jedem Bild an der gleichen Position auftreten, gut komprimiert werden können.

Da der verwendete Encoder (mpeg2vidcodec, Version 1.2 von [32]) nur bestimmte Eingabe-

[7]**P**ortable **gray**ma**p**, ein Datei-Format von Jeff Poskanzer, siehe z.B. [16].

[8]Hier wurden die Originaldaten mit einer Schwelle von 3, also die Originaldaten, verwendet, um die Information so wenig wie möglich vor der MPEG-Codierung zu verfälschen.

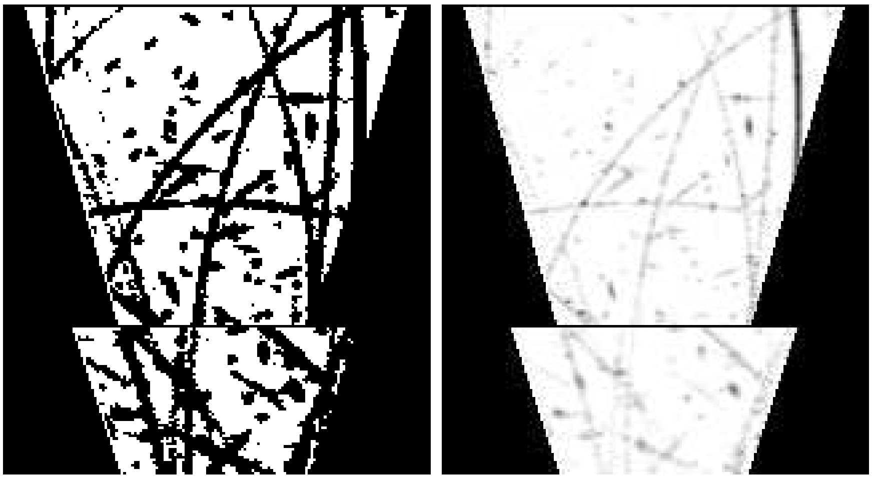


Abbildung 5.17: Beispiel einer Bilddatei.
Links: Alle ADC Werte ungleich Null sind schwarz markiert.
Rechts: ADC Werte in Graustufen. Dieses Format wird für die MPEG-Codierung verwendet.

Formate zuläßt, werden die Einzelbilder in das ppm[9]-Format mittels des Linux-Programms *pgmtoppm* konvertiert, bevor sie an den Encoder und den Decoder weitergereicht werden. Anschließend werden die decodierten (ppm-) Bilder wieder zurück konvertiert und in das Binärformat umgewandelt. Abb. 5.16 zeigt das Vorgehen schematisch.

Die Einzelbilder entsprechen dem Blick auf die Endkappen zu einem bestimmten Zeitschritt. Abb. 5.17 zeigt die Aufteilung eines Einzelbildes in inneren und äußeren Sektor sowie die mit schwarz aufgefüllten Bereiche, in denen keine Pads sind. Für jedes Pad wurde ein Pixel gewählt. Aus diesem Grunde und da die Pads in den inneren und äußeren Sektoren verschiedene Dimensionen haben, schließen die Ränder nicht bündig aneinander an. Das linke Bild in Abb. 5.17 vermittelt zudem einen guten Eindruck von der Auslastung der TPC.

Abb. 5.18 zeigt den Einfluß der verlustbehafteten Codierung durch den verwendeten MPEG-Encoder bei verschiedenen Bitraten. Bei niedrigen Bitraten zeigen sich deutlich die Weichzeichnungseffekte und die Artefakte durch die Interpolation und die Quantisierung. Einzelne 8x8 Pixel große Blöcke sind hier gut zu erkennen.

Abb. 5.19 zeigt den Einfluß der MPEG-Codierung bei verschiedenen Bitraten und somit verschiedenen Größen des entstandenen Films auf die Spurerkennungseffizienz.

Erst bei sehr hohen Bitraten von über 1.5 Mbps erreicht die Effizienz Werte, die in der Nähe der erreichten Effizienz mit den Originaldaten liegen. Diese hohen Bitraten können das Datenvolumen relativ zu den lauflängen-codierten Originaldaten jedoch nicht mehr verkleinern.

[9] **P**ortable **p**ix**m**ap, ein Datei-Format von Jeff Poskanzer, siehe z.B. [16].

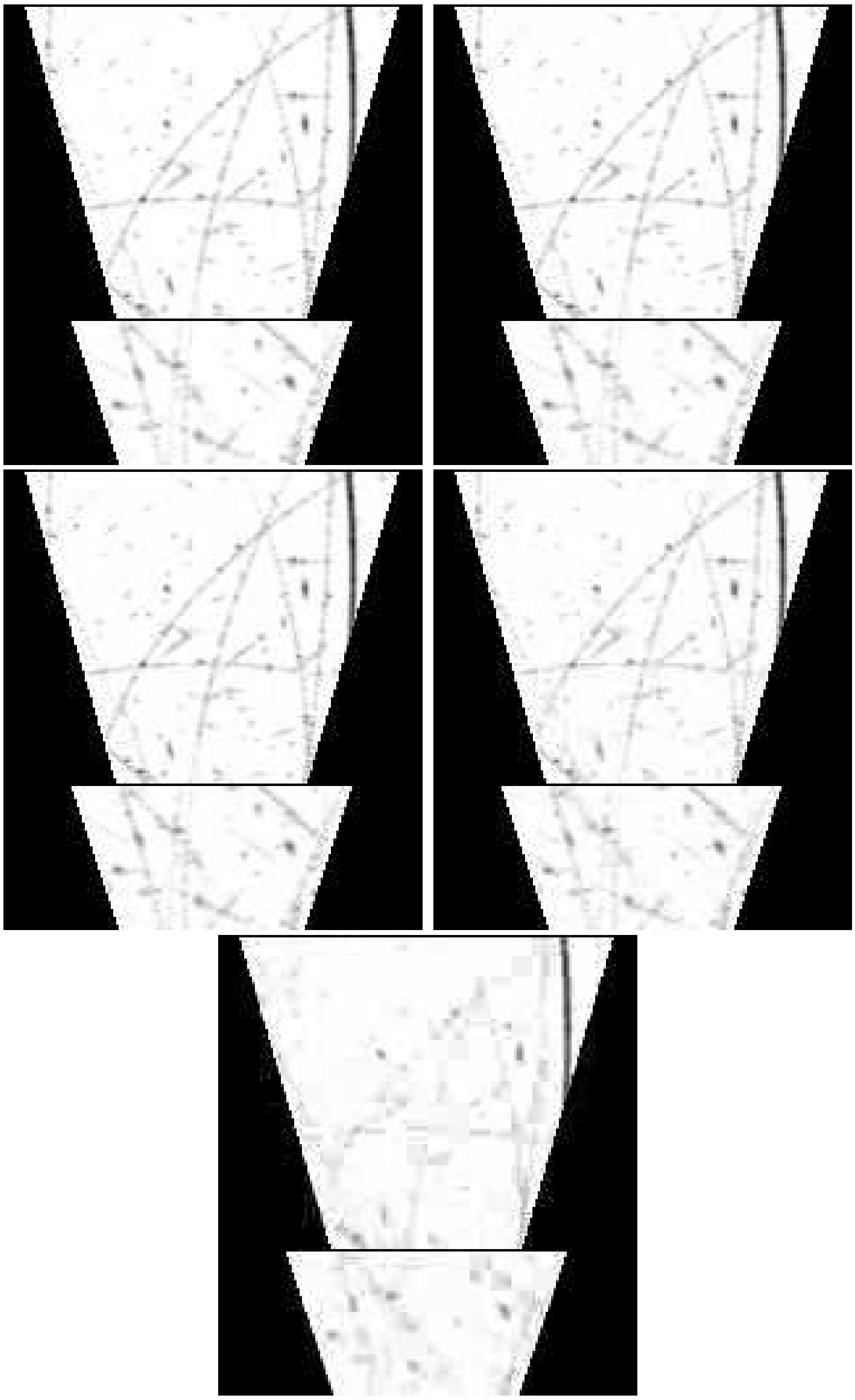


Abbildung 5.18: Originalbild und decodierte Einzelbilder bei verschiedenen Bitraten. Von links nach rechts und von oben nach unten: Originalbild, Bitraten 1.5 Mbps, 1.0 Mbps, 0.5 Mbps und 0.1 Mbps.

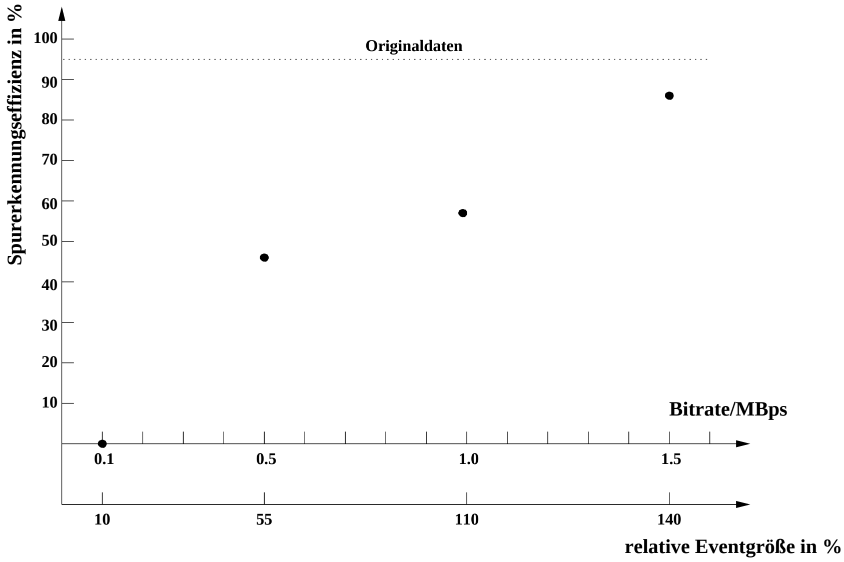


Abbildung 5.19: Spurerkennung nach MPEG-Codierung in Abhängigkeit von der Bitrate.

Der MPEG-Codierer kann seine Eingabedaten, die ja künstlich durch das Bildformat um etwa einen Faktor 18 relativ zu dem Referenzdatenvolumen vergrößert wurden, bei den verwendeten Bitraten auf 1-8% reduzieren. Dies entspricht allerdings relativ zur Größe der Originaldaten nur Kompressionsraten von 10-140%.

Bei Ereignisgrößen, die größenmäßig denjenigen entsprechen, die mit dem Vektorquantisierer erreicht werden können, liegt die Spurerkennungseffizienz nur bei etwa 50%.

Durch die Rückkopplung der Bitrate auf den Quantisierer wird bei mangelnder Korrelation zwischen den Bildern, d.h. wenn zu wenige Verschiebungsvektoren bei der Bewegungsschätzung gefunden werden konnten, Originalblöcke gespeichert werden mußten und so bereits zu viele Bits für eine Bildgruppe verbraucht wurden, die Qualität des Films und damit auch die der decodierten Einzelbilder sichtbar verschlechtert. Je niedriger die Bitrate - je besser die Kompression - desto häufiger tritt dieser Effekt auf und macht die Daten unbrauchbar. Durch die Einschränkung auf die Erzeugung einer festen Bitrate also ist die MPEG-Codierung für diese spezielle Anwendung ungeeignet.[10]

Eine stärkere Korrelation zwischen den Einzelbildern, die eventuell durch das Aneinanderhängen von Einzelbildern vom Strahlrohr nach außen anstatt in Zeitrichtung erzielbar

[10] Die Verwendung eines anderen Videocodierers (*ppm2fli* von Klaus Ehrenfried, siehe z.B. http://vento.pi.tu-berlin.de/ppm2fli/main.html), das die Daten in das Autodesk Animator-Format (FLI), siehe z.B. [16], transformiert, erzielt bei Kompressionsraten von etwa 140% (durch zusätzliche Anwendung von gzip 88%) eine Spurerkennungseffizienz, die derjenigen der Originaldaten entspricht. Zur weiteren Kompression muß hier allerdings die Zahl der verwendeten Quantisierungsstufen wesentlich erniedrigt werden, wodurch die Daten wiederum unbrauchbar werden. Eine eventuelle Anpassung des Verfahrens zur Reduzierung der Farbanzahl auf die logarithmische Verteilung der Daten entspräche einer Quantisierung, die auch auf den Daten direkt ausgeführt werden könnte, um durch die Huffman-Codierung ausgenutzt zu werden.

wäre, könnte diesen Effekt vermindern und etwas bessere Kompressionsraten bewirken.

Letztendlich ist die MPEG-Codierung durch die Unterdrückung hoher Frequenzanteile nach der DCT natürlich darauf angelegt, die Qualität eines Films subjektiv für das menschliche Sehsystem zu erhalten und nutzt dessen Eigenschaften geschickt für die Kompression aus. Die decodierten Daten allerdings sind deutlich verändert, was dann in der objektiven Analyse spürbar wird.

Kapitel 6

Zusammenfassung und Ausblick

"Begin at the beginning", the King said gravely,
"and go on till you come to the end: then stop."
Lewis Carroll (1832-1898)

In dieser Arbeit wurden verschiedene Techniken zur Kompression der TPC-Daten von ALICE untersucht.

Als verlustloses Verfahren kam die Huffman-Codierung zum Einsatz, die sich durch die genaue Kenntnis der inhärenten Struktur der Daten und des Datenformats in einer angepaßten Implementierung den Ergebnissen von Standardanwendungen (*pack* und *compact*) als überlegen erwies. Das Datenvolumen konnte hiermit auf 65% reduziert werden. Die Anwendung dieses Verfahrens auf die differenzierten Daten brachte keinen Vorteil.

Die verlustbehaftete Vektorquantisierung wurde im Detail auf ihre Anwendbarkeit untersucht. Die erreichten Kompressionsraten lagen zwischen 64% und 48%[1], je nach Größe des zugelassenen Fehlers.

Die relativ geringen Kompressionsraten bei diesem Verfahren erklären sich aus der Struktur der Daten: die meisten eindimensionalen Pulse haben inklusive Vor- und Nachgeschichte eine Länge von 4 Zeit-Bins. Mit den Headerinformationen von 13 Bits ergibt sich so eine Größe von 45 Bits. Haben nun die Code-Wörter eines Vektorquantisierers die Dimension 4 und kann jedem Eingabe-Vektor ein Code-Wort zugeordnet werden - was bereits dem Idealfall entspricht - , so erhält man z.B. für ein Code-Buch mit 1024 Einträgen (10 Bits) eine maximale Kompression von 51% (ohne Codierung der Headerinformationen). Die Effizienz der Kompression durch einen Vektorquantisierer wird also von vornherein durch den hohen Anteil an Zusatzinformationen bzw. die relativ kurzen Pulse eingeschränkt.

Die Auswirkungen der Modifikation der Daten durch diese Art der Komprimierung wa-

[1] Die ermittelten Werte stimmen mit denen anderer Untersuchungen überein [35].

ren meßbar, aber sehr klein. Als empfindlichste Größe erwies sich hierbei die Ortsauflösung. Durch die Anwendung der Schwelle und der Kompression durch den Vektorquantisierer wurden relative Verschlechterungen der Ortsauflösung bei voller Multiplizität in transversaler Richtung von 3% in den inneren Kammern bzw. 6% in den äußeren Kammern verursacht.

Vergleicht man nun die Huffman-Codierung und die Vektorquantisierung hinsichtlich ihrer Anwendbarkeit, so läßt sich feststellen, daß trotz der Möglichkeit der verlustbehafteten Kompression die Vektorquantisierung keine wesentlich besseren Komprimierungsergebnisse als die verlustlose Huffman-Codierung erzielt. Auch bei der Inbetriebnahme der TPC ist der benötigte Installationsaufwand für die Huffman-Codierung kleiner, da der Erzeugung des Baums zur Codierung die Generierung und das Training des Code-Buchs bei der Vektorquantisierung gegenüber steht, die einen wesentlich höheren Aufwand implizieren. Weichen die Daten allerdings stark vom Modell und den damit verbundenen Annahmen ab, z.B. durch ein unerwartetes Rauschen, so werden die Daten zwar durch keines der beiden Verfahren verfälscht, jedoch beträgt die maximale Vergrößerung des Datenvolumens bei der Vektorquantisierung nur 1/8, während sich die Auswirkungen bei der Huffman-Codierung so einfach nicht vorhersagen lassen. Hier kann man allerdings durch die Anwendung eines adaptiven Verfahrens den Baum zur Codierung mit wenig Aufwand an die neuen Gegebenheiten anpassen.

Die Verwendung bereits existierender und sehr effizienter Videokompressionstechniken aus der Bildverarbeitung erwies sich als nicht praktikabel. Das MPEG-Verfahren ist durch die Rückkopplung der erzeugten Bitrate und der damit verbundenen Verschlechterung der Bildqualität bei zu geringer Korrelation zwischen den Einzelbildern für diese Anwendung ungeeignet. Nichtsdestotrotz könnte die Bewegungsschätzung durch die Korrelation aufeinander folgender Zeitscheiben oder durch die Verfolgung der Cluster einer Spur von innen nach außen durch die Kammer eine Möglichkeit zur Kompression sein, wenn sie auf diese spezielle Anwendung angepaßt wird. Die Transformation in das FLI-Videoformat konnte bei benötigter Qualität keine überzeugenden Kompressionsraten erzielen.

Die rasante Entwicklung auf dem Gebiet der Bilddatenkompression allerdings könnte schon bald Verfahren hervorbringen, die bei gleicher Qualität noch bessere Kompressionsraten erzielen. Ein erste Untersuchung der Kompressionseigenschaften von neueren Techniken zur Einzelbildkompression wurde unternommen. Abb. 6.1 zeigt die Ergebnisse von JPEG-, Wavelet- und Fraktalkompression[2] auf ein Einzelbild bei Kompressionsraten, die eine deutlichere Verbesserung der Kompression gegenüber den getesteten Verfahren darstellen würden. Mit bloßem Auge jedoch sind die Entfremdungen gegenüber dem Original aus Abb. 5.17 von Seite 73 zu erkennen. Die Analyse nach der Decodierung zeigte dann, daß die verwendeten Verfahren bei den geforderten Kompressionsraten auch tatsächlich die in den Daten enthaltenen Informationen unkenntlich machten.

[2] Zur Erläuterung der verwendeten Techniken sei auf [31], [36] verwiesen.

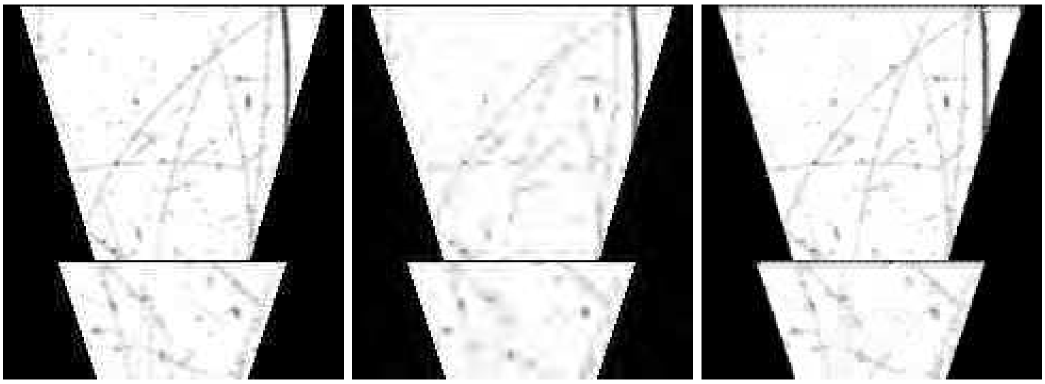


Abbildung 6.1: Decodierte Einzelbilder bei Kompressionsraten, die das Gesamtdatenvolumen auf etwa 30% reduzieren würden.
V.l.n.r.: JPEG, Kompression durch Wavelets und Fraktale Kompression.

Ein weiterer klarer Nachteil bei der Verwendung bereits implementierter Bildkompressionsverfahren wie diesen ist die Unkenntnis darüber, was genau mit den einzelnen ADC Werten geschieht.

Hier nicht untersuchte Verfahren zur Modellierung von Clustern und Spuren, bei deren Anwendung dann neben den Modellparametern noch die Abweichungen vom Modell gespeichert werden, versprechen Kompressionsraten bis auf 10%. Für die Verwendung solcher Verfahren allerdings muß die Rekonstruktion der Spuren bereits während der Datenaufnahme geschehen. Zu diesem Zweck wird zur Zeit für die ALICE TPC ein Online-Tracker entwickelt, der auf der Level-3 Rechner-Farm laufend diese Aufgabe erfüllen soll.

Anhang A

Ergebnisse aller Parametersätze

Die Abb. A.1, A.2 und A.3 geben einen Überblick über die Erkennungs- und Kompressionsraten einzelner Parametersätze. Auf der Abszisse ist die jeweils maximal tolerierte prozentuale Ladungsabweichung zwischen Daten- und Code-Buch-Vektor aufgetragen.

Im dritten Bild jeder Zeile ist der Betrag der gemittelten Abweichung der Ladung mit der zugehörigen Standard-Abweichung aufgetragen.

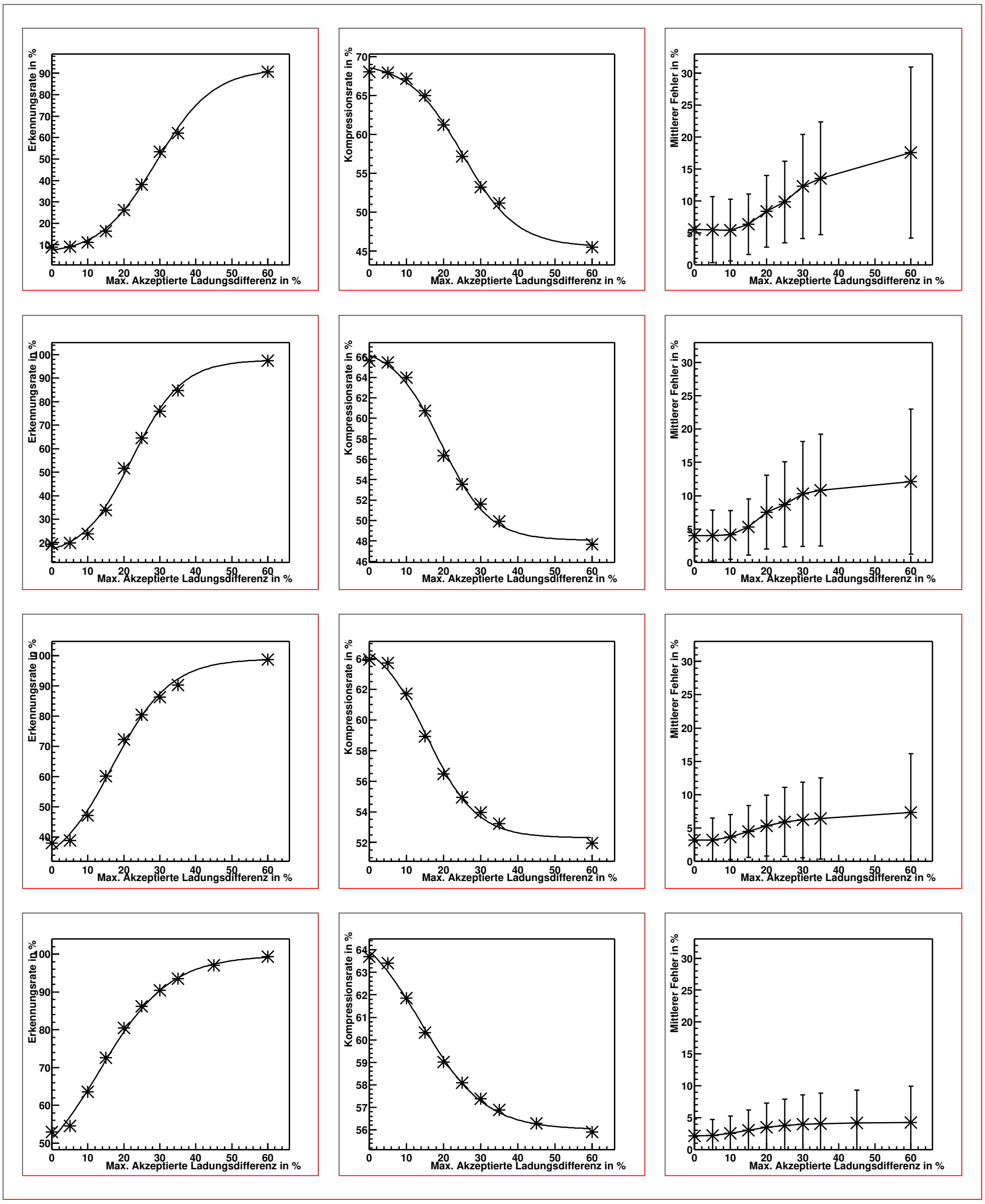


Abbildung A.1: Erkennungs- und Kompressionsraten sowie der tatsächlich akzeptierte Fehler für die Vektordimension 2. Die Code-Buch-Größen sind 32, 64, 128 und 256 (von oben nach unten).

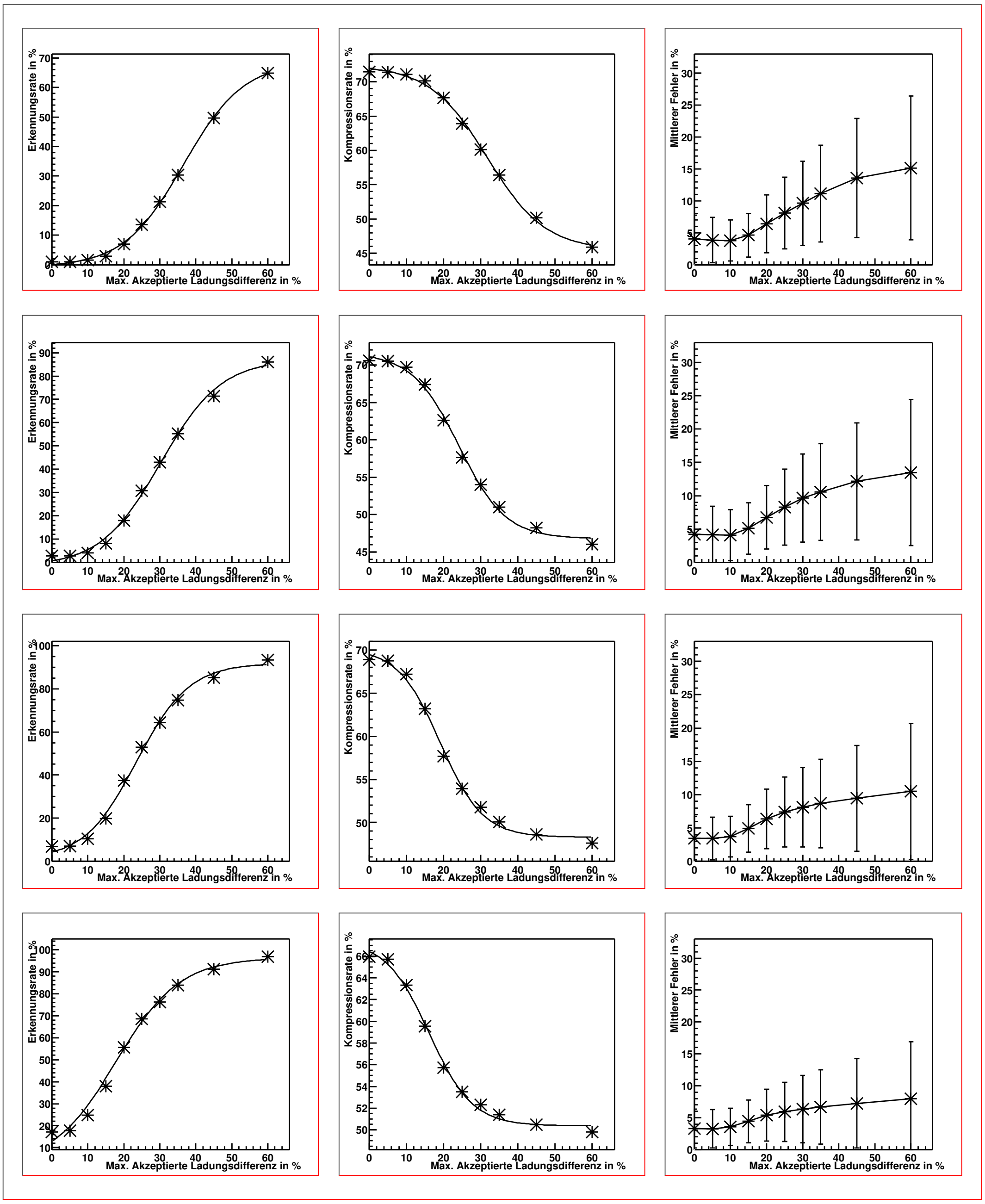


Abbildung A.2: Erkennungs- und Kompressionsraten sowie der tatsächlich akzeptierte Fehler für die Vektordimension 3. Die Code-Buch-Größen sind 64, 128, 256 und 512 (von oben nach unten).

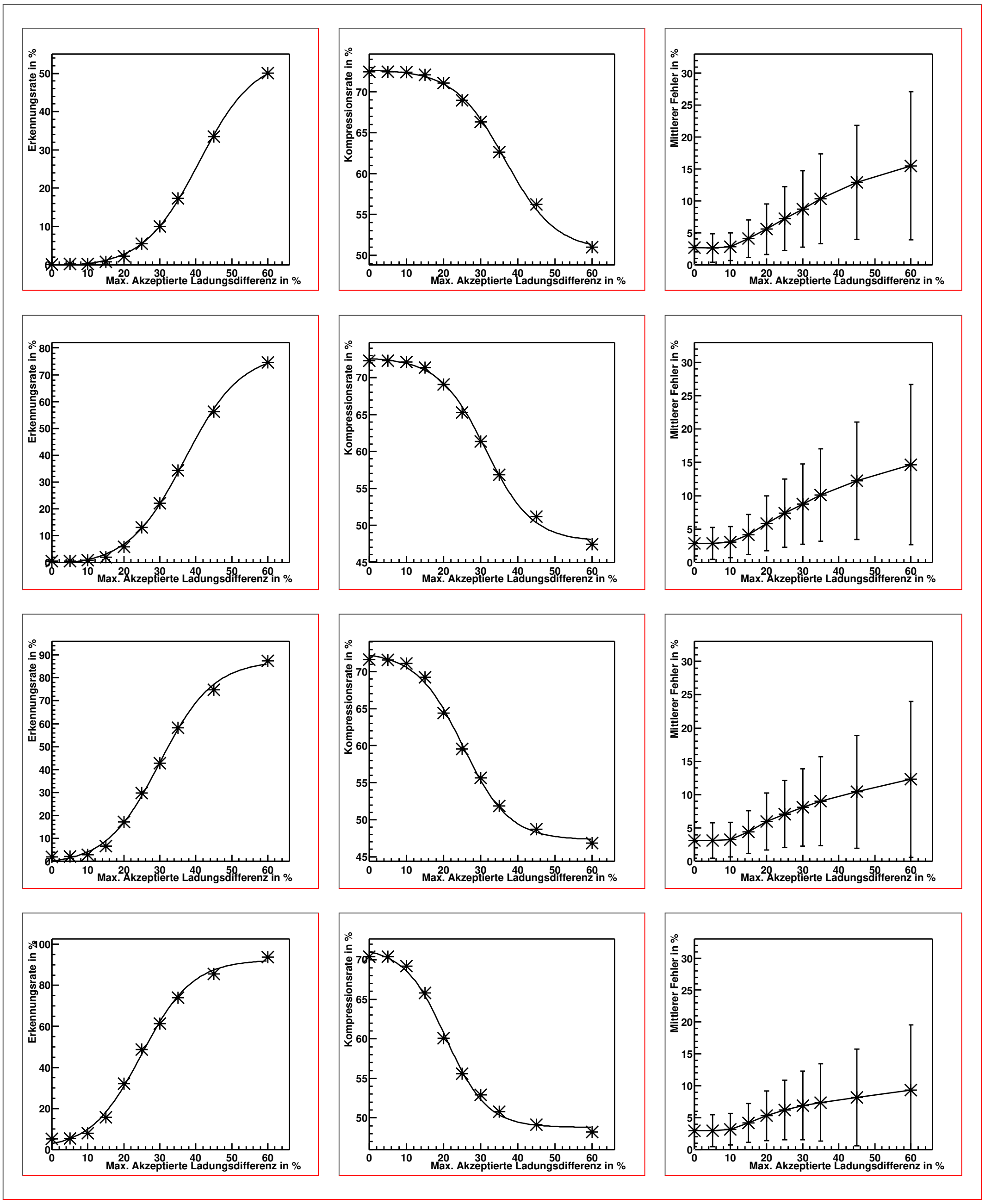


Abbildung A.3: Erkennungs- und Kompressionsraten sowie der tatsächlich akzeptierte Fehler für die Vektordimension 4. Die Code-Buch-Größen sind 128, 256, 512 und 1024 (von oben nach unten).

Anhang B

Ergebnisse der Verfahren

Alle untersuchten Verfahren, die die in den Daten enthaltene Physik nicht verfälschen, sind in Tabelle B mit den erreichten Kompressionsraten aufgeführt.

Kompressionsverfahren	verlustbehaftet	Rel. Größe in %
Nullunterdrückte Roh-Event-Größe	ja	100
Verwendetes Datenformat (siehe Kap. 4.1)	ja	∼88
UNIX gzip (LZ77)	nein	80
UNIX compress (LZW)	nein	92
UNIX pack (statische Huffman-Codierung)	nein	80
UNIX compact (adaptive Huffman-Codierung)	nein	80
Arithmetische Codierung (Implementation aus [33])	nein	80
Huffman-Codierung (differenzierte Daten)	nein	73
Huffman-Codierung (ADC-Werte)	nein	65
Vektorquantisierer	ja	64-48

Tabelle B1.:Kompressionsverfahren im Vergleich. Die angegebenen Größen beziehen sich auf die Größe der nullunterdrückten lauflängencodierten Roh-Event-Größe. Angewandt wurden alle Verfahren auf Daten, die im in Kapitel 4.1 definierten Datenformat vorlagen.

Anhang C

Die CD

Die beigefügte CD enthält neben den Postscript- und Pdf-Versionen dieser Arbeit (Verzeichnis */thesis*) die erstellten Quellen sowie alle verwendeten Programme.

Die Binärdaten der simulierten Ereignisse sowie einige Ergebnisse der Analyse sind ebenfalls enthalten (Verzeichnisse */data* bzw. */results*).

Für eine detailliertere Inhaltsübersicht und tiefergehende Erläuterungen sei auf die READMEs in den entsprechenden Unterverzeichnissen verwiesen.

Literaturverzeichnis

[1] G. Alberici et al. Image Compression for the Silicon Drift Detectors in the ALICE Experiment. ALICE/PUB 2000-11, 10. Mai 2000.

[2] Die ALICE Kollaboration. Technical Proposal for A Large Ion Collider Experiment at the CERN LHC. CERN/LHCC 95-71, 15. Dezember 1995.

[3] Die ALICE Kollaboration. Technical Design Report of the Inner Tracking System (ITS). CERN/LHCC 99-12, 18. Juni 1999.

[4] Die ALICE Kollaboration. Technical Design Report of the Time Projection Chamber (TPC). CERN/LHCC 2000-001, 7. Januar 2000.

[5] Die ALICE Kollaboration. A Transition Radiation Detector for Electron Identification within the ALICE Central Detector (TRD). Addendum to the ALICE Technical Proposal. CERN/LHCC 99-13, LHCC/P3-Addendum 2, 7. Mai 1999.

[6] Die ALICE Kollaboration. Technical Design Report of the Time Of Flight System (TOF). CERN/LHCC 2000-12, 16. Februar 2000.

[7] Die ALICE Kollaboration. Technical Design Report of the Photon Spectrometer (PHOS). CERN/LHCC 99-4, 5. März 1999.

[8] Die ALICE Kollaboration. Technical Design Report of the High Momentum Particle Identification Detector (HMPID). CERN/LHCC 98-19, 14. August 1998.

[9] Die ALICE Kollaboration. Technical Design Report of the Dimuon Forward Spectrometer. CERN/LHCC 99-22, 13. August 1999.

[10] Die ALICE Kollaboration. Centauro And STrange Object Research (CASTOR). A detector for ALICE at very forward rapidity dedicated to the identification and study of 'centauros' and 'strangelets'. Technical proposal (Current status report). ALICE/97-07 Internal Note, 11. März 1997.

[11] Die ALICE Kollaboration. Technical Design Report of the Zero Degree Calorimeter (ZDC). CERN/LHCC 99-5, 5. März 1999.

[12] Die ALICE Kollaboration. Preliminary Technical Design Report for the Forward Multiplicity Detector based on Micro Channel Plates, Februar 1999.

[13] Die ALICE Kollaboration. Technical Design Report of the Photon Multiplicity Detector (PMD). CERN/LHCC 99-32, 30. September 1999.

[14] A. L. S. Angelis et al. Centauro And STrange Object Research (CASTOR). A Detector for ALICE at very high forward rapidity dedicated to the identification and study of 'Centauros' and 'Strangelets'. ALICE Internal Note/ CAS 97-07, 11. März 1997.

[15] J. Belikov, CERN, Private Korrespondenz.

[16] G. Born, Referenzhandbuch Dateiformate. Datenbanken, Tabellenkalkulation, Text, Grafik, Multimedia, Sound und Internet. Addison-Wesley. Bonn, Reading, Massachusetts [u.a.], 1997.

[17] S. Brandt. Datenanalyse. BI Wissenschaftsverlag. Mannheim, Leipzig, Wien, Zürich 1992.

[18] B. Buchanan, Handbook of Data Communications and Networks. Kluwer Academic Publishers. Boston, Dordrecht, London 1999.

[19] CERN Press Release. New State of Matter created at CERN, 10. Februar 2000.

[20] R. M. Fano. Ph.D. Thesis. Massachusetts Institute of Technology, Cambridge, 1949.

[21] A. Gersho und R. M. Gray. Vector Quantization and Signal Compression. Kluwer Academic Publishers, Boston, Dordrecht, London, 1992.

[22] http://root.cern.ch

[23] http://AliSoft.cern.ch/Offline

[24] D. Huffman. A method for the Construction of Minimum Redundancy Codes. Proceedings of the IRE 40(9):1098-1101, 1952.

[25] R. L. Glückstern, Nucl. Instr. Meth. 24 (1963) 381.

[26] B. Jähne, Digitale Bildverarbeitung. Springer Verlag. Berlin, Heidelberg, 1997.

[27] K. Kleinknecht. Detektoren für die Teilchenstrahlung. Teubner Verlag, Stuttgart 1984.

[28] The LHC Study Group, The Large Hadron Collider, Conceptual Design Report, CERN/AC/95-05 (LHC).

[29] Y. Linde, A. Buzo und R. M. Gray. A Fake Process Approach to Data Compression. IEEE Transactions on Communication, COM-28:84-95, Januar 1980.

[30] S. P. Lloyd. Least squares quantization in PCM. Veröffentlicht in einer Spezialausgabe über Quantisierung der IEEE Transactions on Information Theory, März 1982.

[31] T. Milde, Videokompressionsverfahren im Vergleich. d-punkt, Verlag für digitale Technologie. Heidelberg, 1995.

[32] MPEG Software Simulation Group (MSSG), http://www.mpeg.org/MSSG.

[33] M. Nelson, J.-L. Gailly. The Data Compression Book. M&T Books. New York 1995.

[34] B. Povh, K. Rith, C. Scholz und F. Zetsche. Teilchen und Kerne. Springer Verlag, Berlin, Heidelberg, New York 1997.

[35] D. Röhrich, Universität Bergen/Norwegen. Private Mitteilung.

[36] D. Salomon. Data Compression - The Complete Reference. Springer Verlag, New York, Berlin, Heidelberg 1998.

[37] C. E. Shannon und W. Weaver. The Mathematical Theory of Communication. University of Illinois Press. Urbana 1949.

[38] J. A. Storer. Data Compression - Methods and Theory. Computer Science Press. Rockville 1988.

[39] A. Vestboe, Universität Bergen/Norwegen. Private Korrespondenz.

[40] T. A. Welch. A Technique for High-Performance data Compression. IEEE Compter 17(6):8-19, Juni 1984.

[41] J. Ziv und A. Lempel. A Universal Algorithm for Sequential Data Compression. IEEE Transactions on Information Theory IT-23(3):337-343, 1977.