Tracking mit der Hough–Transformation für die Zentrale Driftkammer des GSI–4π–Experiments

Diplomarbeit
von Dieter Best

Universität Mainz
August 1992

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Als Zarathustra dreißig Jahre alt war, verließ er seine Heimat und den See seiner Heimat und ging in das Gebirge. Hier genoß er seines Geistes und seiner Einsamkeit und wurde dessen zehn Jahre nicht müde. Endlich aber verwandelte sich sein Herz, – und eines Morgens stand er mit der Morgenröte auf, trat vor die Sonne hin und sprach zur ihr also:

Du großes Gestirn! Was wäre dein Glück, wenn du nicht die hättest, welchen du leuchtest!

Zehn Jahre kamst du hier herauf zu meiner Höhle: du würdest deines Lichtes und dieses Weges satt geworden sein, ohne mich, meinen Adler und meiner Schlange.

Aber wir warteten deiner an jedem Morgen, nahmen dir deinen Überfluß ab und segneten dich dafür.

Siehe! Ich bin meiner Weisheit überdrüssig, wie die Biene, die des Honigs zu viel gesammelt hat, ich bedarf der Hände, die sich ausstrecken.

– FRIEDRICH NIETZSCHE, ALSO SPRACH ZARATHUSTRA, ERSTER TEIL, ZARATHUSTRAS VORREDE

Kapitel 1

Einführung – Pattern Recognition in der Physik

Die Zeit ist da, sagte das Walroß, um über viele Dinge zu sprechen.

— L. Carrol

Seit den Tagen der Wilsonschen Nebelkammer müssen Teilchenspuren, sogenannte *Tracks,* in Gaskammern erkannt und gemessen werden. Dieser Prozeß wird als *Pattern Recognition* bezeichnet.

Anfangs war das Aufzeichnungsmedium ein Film. Filme werden auch heute noch verwendet bei sogenannten *optischen* Detektoren, wie zum Beispiel Blasenkammern, optischen Sparkkammern und optischen Streamerkammern (siehe Abbildung 1.1), bei denen geladene Teilchen sichtbare Tracks hinterlassen, die fotografiert werden können.

Seit etwa 1960 werden auch Computer dafür eingesetzt. Dies setzt voraus, daß die Daten in digitaler Form vorliegen. Drei verschiedene Methoden wurden dafür entwickelt:

1. Fotos werden digitalisiert, d.h. sie werden in kleine Zellen aufgeteilt, und der *Grauwert* jeder Zelle wird aufgenommen.

2. Anstatt das Bild zu fotografieren, wird es mit einer TV Kamera aufgenommen. Das so erhaltene Analogsignal, eine elektrische Spannung, wird digitalisert.

3. Sogenannte elektronische Detektoren wurden entwickelt. Diese basieren auf anderen Effekten als der Lichtemission, um den Durchgang von

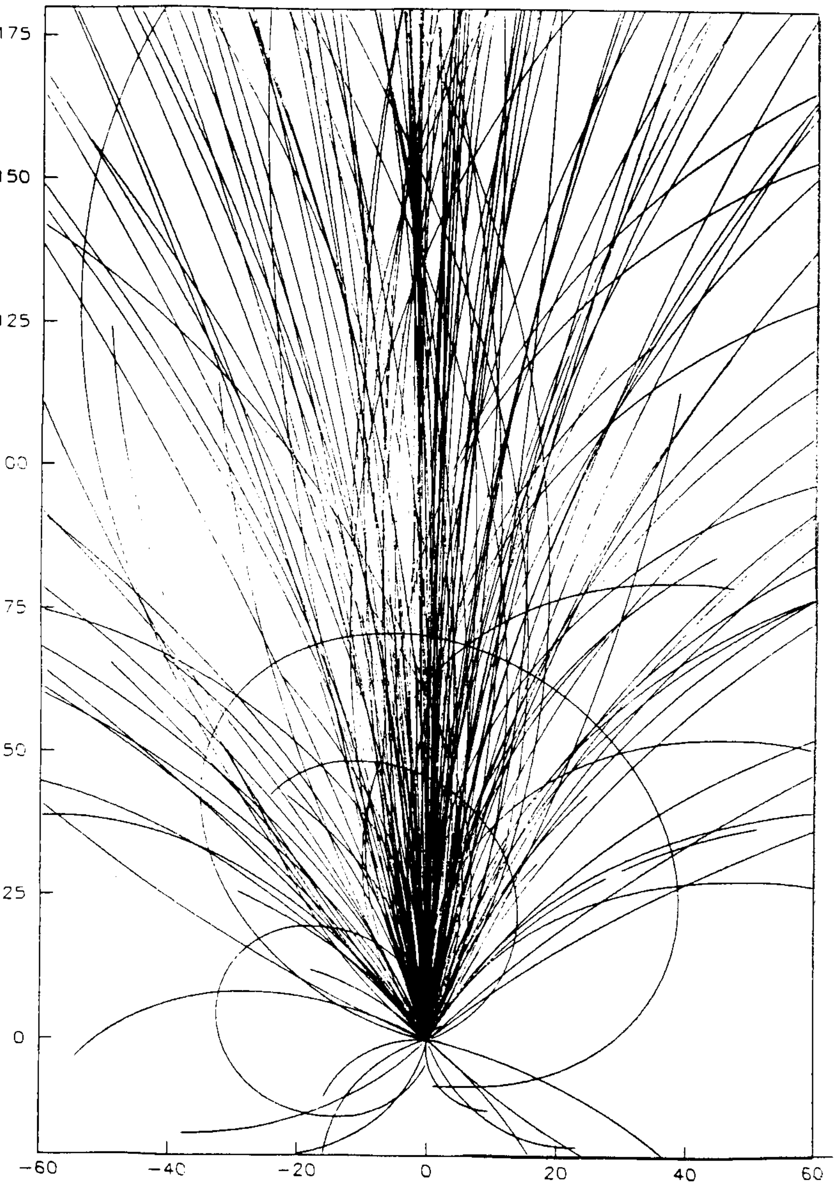

Abbildung 1.1: Simuliertes Streamerkammerbild für eine zentrale Kollision von ^{16}O+Xe bei 200 GeV/u, generiert als Superposition von 16 p+Xe Reaktionen aufgenommen im Experiment NA35 [STO85].

Teilchen festzustellen. Sie liefern entweder eine direkte digitalisierte Information, beispielsweise die Drahtnummer, oder sie produzieren ein Analogsignal, das dann digitalisiert wird.

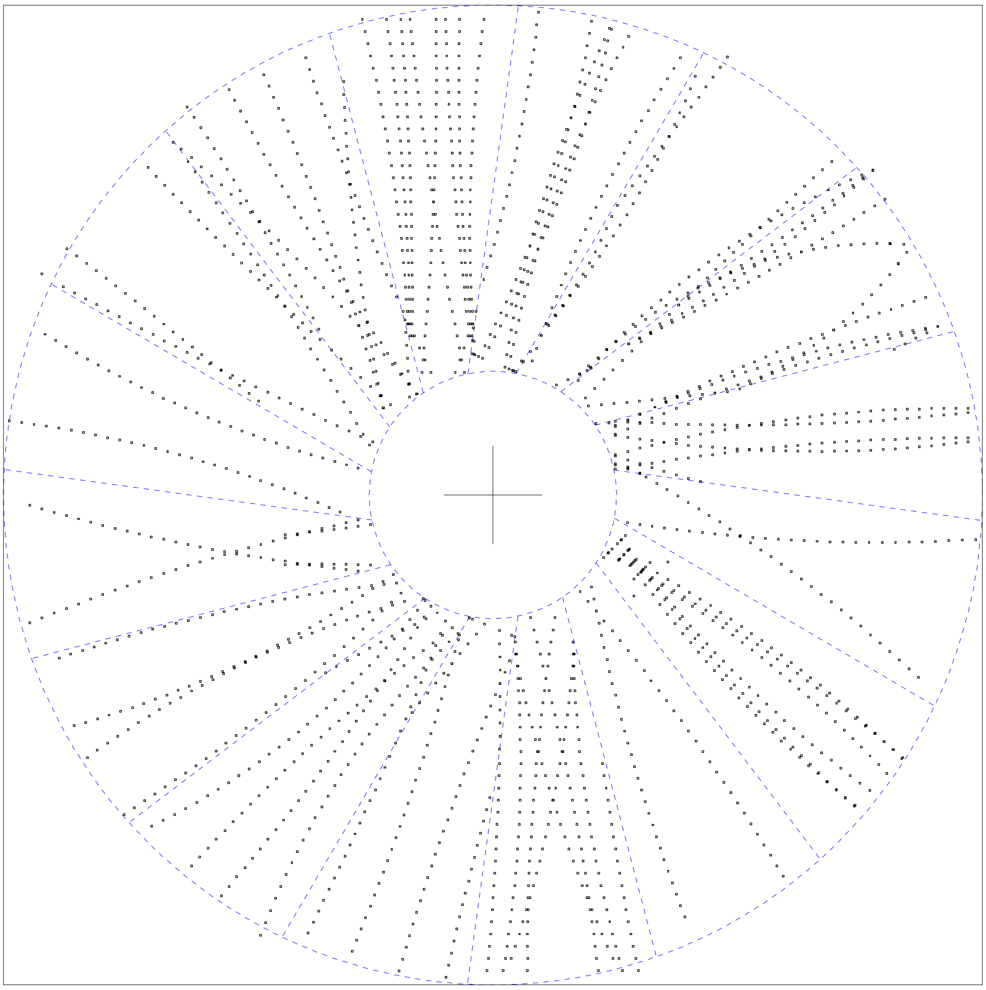

Abbildung 1.2: Projektion einer simulierten Au+Au–Reaktion bei 1 GeV/u in der SIS–4π–CDC (zentrale Driftkammer) senkrecht zur Strahlachse. Die Kammer befindet sich in einem solenoidalen Magnetfeld parallel zur Strahlachse. Das Kreuz zeigt die Targetposition an.

Elektronische Detektoren (Abb. 1.2) haben heute fast vollständig Blasenkammern oder andere optische Detektoren verdrängt, hauptsächlich weil man mit ihnen besser triggern kann, weil sie eine höhere Auflösung und eine kürzere Totzeit haben.

Die automatische Verarbeitung der Daten ist nun die Regel. Die Software, um diese Aufgabe zu erfüllen, ist ein integraler Bestandteil des Experiments. Darunter ist die *Spurerkennung*, das *Track–Finding*, von entscheidender Bedeutung für die meisten Experimente. Dies geht so weit, daß beim Design moderner Detektoren die Track–Finding–Methode, die angewandt werden soll, mit berücksichtigt wird [GRO87]. Bei elektronischen Detektoren, die typischerweise Bilder wie in Abbildung 1.2 liefern, besteht das Problem der Spurerkennung darin, aus den gemessenen Koordinaten, den sogenannten *Hits*, die Teilchenspur räumlich zu rekonstruieren. So erhält man die Richtung und über die Krümmung der Spur im Magnetfeld den Betrag des Teilchenimpulses.

Ziel dieser Arbeit ist es, die Leistungsfähigkeit einer neuen Methode zur Spurerkennung in Driftkammern, der *Hough–Transformation* [HOU62], zu zeigen. Als Einführung in die Problematik soll im folgenden die relevante Physik und der Detektor, mit dem diese Physik untersucht wird, beschrieben werden. Daran schließt sich ein Überblick der vorhandenen Track–Finding–Methoden an. Dann wird die Hough–Transformation und deren Möglichkeiten beschrieben. Es folgen darauf zwei Kapitel über Spuranpassung, dem sogenannten *Track–Fitting*, und die Rekonstruktion des primären Vertex, dem *Vertex–Fitting*, die für ein vollständige Analyse nötig sind und einen Vergleich mit den simulierten Ereignissen erst ermöglichen beziehungsweise es erlauben, physikalische Informationen zu extrahieren.

Kapitel 2

Die Physik

One never notices what has been done; one can only see what remains to be done.

– Marie Curie

Geht man davon aus, daß die Kernmaterie, die uns umgibt, sich thermodynamisch beschreiben läßt, so kann man ihr unter anderem folgende Eigenschaften zuordnen:

- Die Dichte ρ ist etwa $\rho_0 \approx 0.17$ Nukleonen/fm^3 .
- Die Temperatur T ist im Grundzustand 0.
- Der Druck P ist 0.

Nun ist es einmal von grundlegendem kernphysikalischem Interesse, wie der Druck sich in Abhängigkeit von der Temperatur und Dichte verhält, also die *Zustandsgleichung der Kernmaterie* zu bestimmen:

$$P = P(\rho, T).$$

Eine Fragestellung ist zum Beispiel, ob es Phasenübergänge gibt.

Abbildung 2.1 zeigt ein Phasendiagramm für Kernmaterie in den Größen ρ und T. Mehrere neue Phasen von Kernmaterie werden in diesem Diagramm erwartet. In der Quantenfeldtheorie, respektive QCD, erwartet man zum Beispiel einen Phasenübergang zum Quark–Gluon–Plasma.

Eine zentrale Rolle spielt die nukleare Zustandgleichung aber auch in der Astrophysik für das Verständnis der Struktur und Dynamik von Sternen, insbesondere der Dynamik von Super–Nova–Explosionen, und

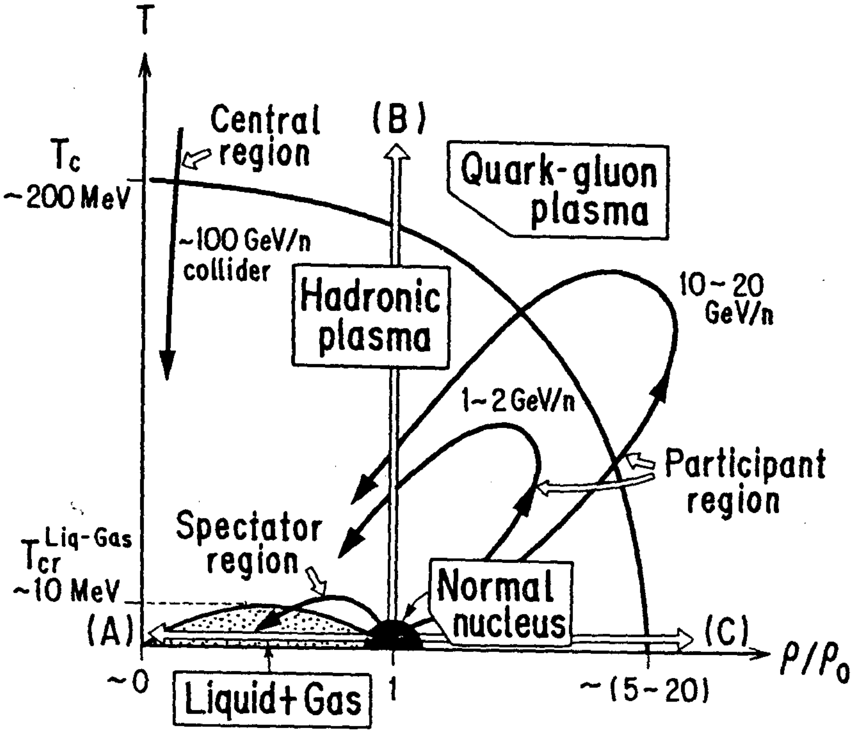

Abbildung 2.1: Erwartete neue Formen von Kernmaterie in der (ρ,T)–Ebene [NAG84]. Mögliche Pfade während eines Schwerionenstoßes sind mit durchgezogenen Linien angezeigt. (A) Übergang vom flüssigen in den gasförmigen Zustand. (B) Übergang von normaler Kernmaterie zu hadronischem Plasma und schließlich Quark–Gluon–Plasma wenn ρ konstant ist und T nur erhöht wird. (C) Übergang von normaler Kernmaterie zu Quark–Gluon–Plasma bei konstanter Temperatur T und erhöhter Dichte ρ.

---

der Stabilität von Neutronensternen. Diese Prozesse hängen stark von der Kompressibilität der Kernmaterie ab. Ebenso werden die Verhältnisse Bruchteile von Sekunden nach dem Urknall entscheidend beeinflußt von der Vorgängen, die in diesem Zusammenhang von Interesse sind.

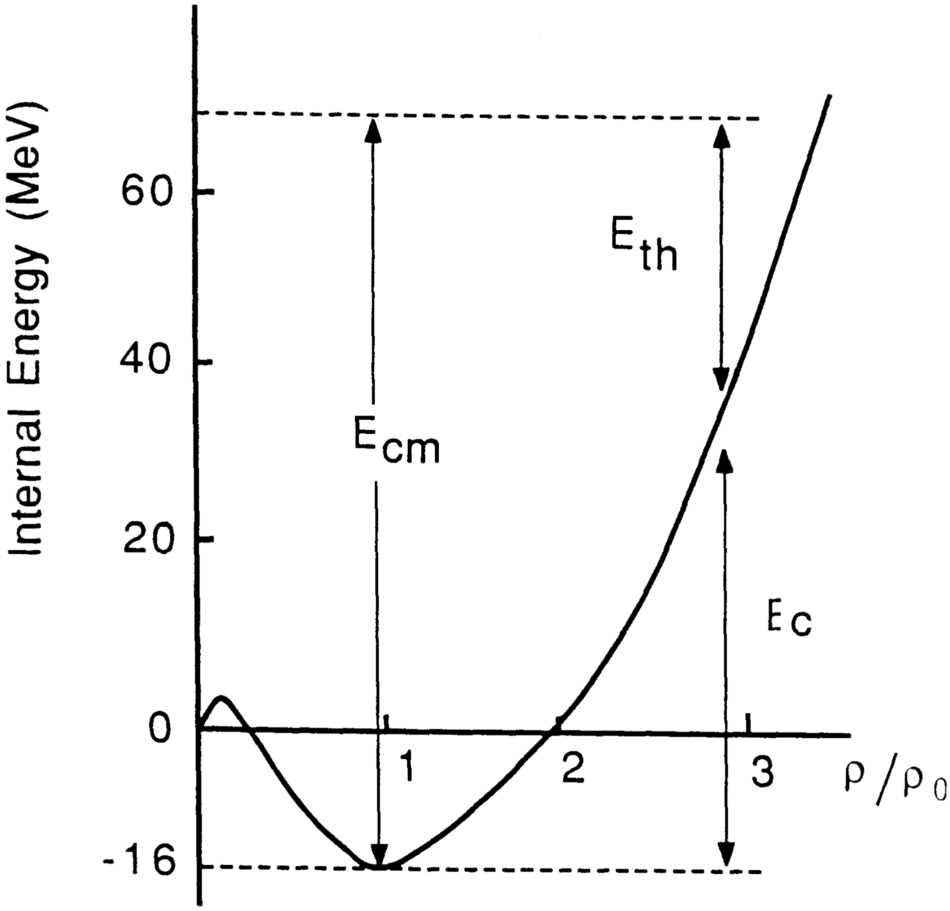

Abbildung 2.2: Erwarteter Verlauf der nuklearen Zustandsgleichung [STO86]: Die Skizze zeigt die Funktion $W(\rho, T = 0)$, die die Energie pro Baryon in Kernmaterie mit der Dichte in Verbindung bringt. Die Dichte ρ wird dabei in Einheiten von $\rho_0 = 0.17$ Nukleonen$/fm^3$ gemessen. Die Reaktion fängt am tiefsten Punkt an (vor der Reaktion), dann kommt die Kompressionsphase und schließlich die Expansionsphase (Expansion bis zum freeze–out). Während der zweiten Phase wird die Energie aufgeteilt auf thermische Energie (E_T) und Kompressionsenergie (E_c).

Diese astrophysikalischen Vorgänge sind aber nicht oder nur sehr schwer einer Beobachtung zugänglich. Durch Stöße schwerer Kerne mit relativistischen Energien erreicht man aber im Labor Zustände hochangeregter und verdichteter Kernmaterie.

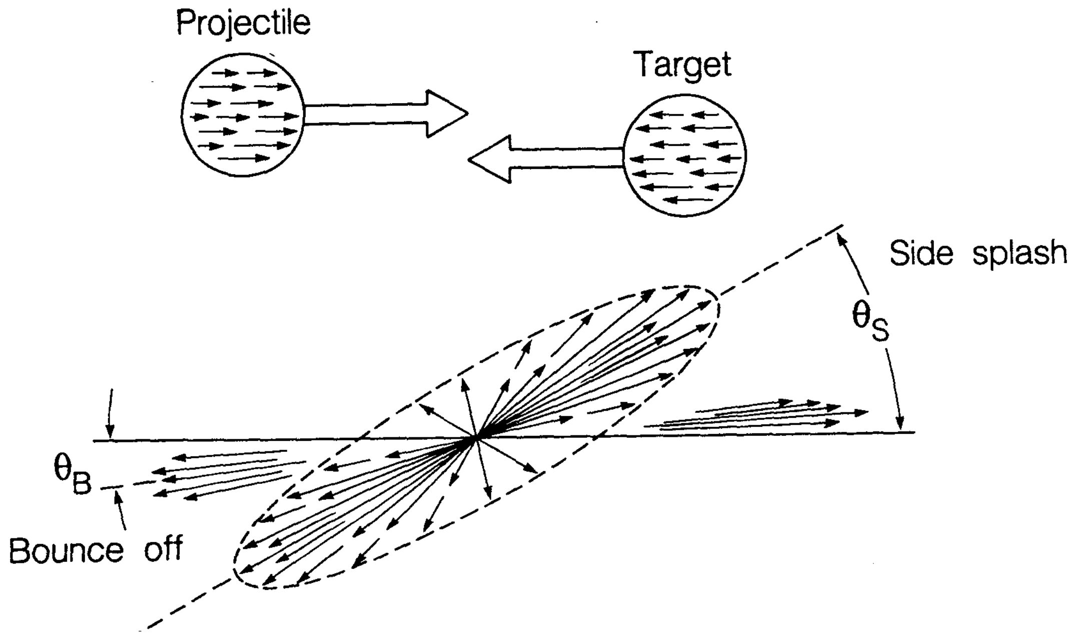

Abbildung 2.3: Schematisierte Ansicht des Side–Splash und Bounce–Off Effektes in relativistischen Schwerionenstößen [RIT85]

Während einer Schwerionenkollision beschreibt man im Phasendiagramm einen Pfad. Die Zeitentwicklung einer solchen Kollision hängt entscheidend von der nuklearen Zustandsgleichung ab.

Abbildung 2.2 zeigt den erwarteten Verlauf der nuklearen Zustandsgleichung in den Variablen interne Energie W pro Nukleon gegen Kompression. Die Kurve $W(\rho/\rho_0, T = 0)$, mit einer Parabelform um ρ_0, gibt die Grundzustandsenergie pro Nukleon von ausgedehnter Kernmaterie wieder: Das ist die Funktion, die man von den Daten zu extrahieren hofft. Sie wird als Zustandsgleichung der Kernmaterie bezeichnet.

Eines der ersten Modelle zur Beschreibung relativistischer Schwerionenstöße war das sogenannte "Feuerball"–("fireball"–) Modell. Abbildung 2.4 zeigt die Modellvorstellung einer Schwerionenkollision. Durch den Stoßparameter der Reaktion wird bestimmt ("clean cut geometry"), wieviele Nukleonen miteinander wechselwirken ("participants") und wieviele nicht auf die Reaktionszone treffen ("spectators").

Dynamische Modelle für Schwerionenstöße bei relativistischen Energien sagen eine Kompressions–Expansions–Abfolge der Participants voraus. Die Reaktion fängt mit der Grundzustandsenergie von -16 MeV/A

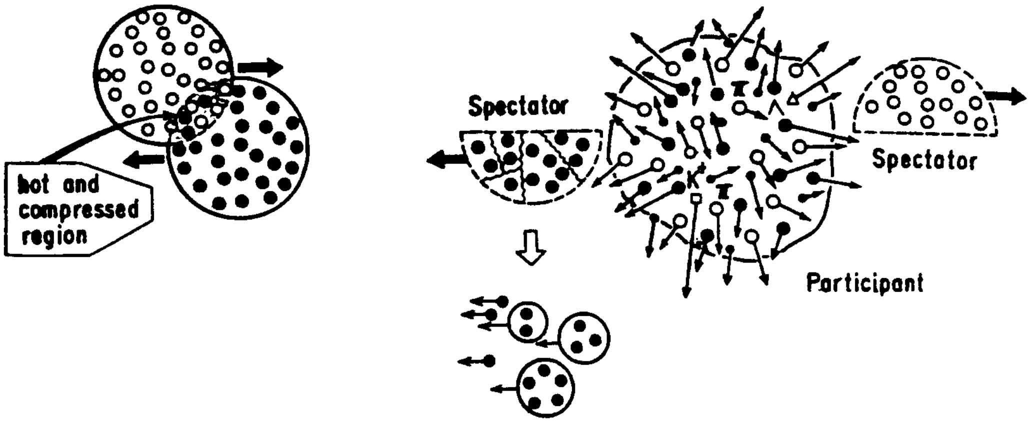

Abbildung 2.4: Erwartetes Raum–Zeit–Verhalten einer Schwerionenkollision bei hohen Energien [NAG84]

---

an, die relative CM–Energie ist durch die Strahlenergie gegeben. Darauf folgt die Kompressionsphase. Nach der maximalen Kompression beginnt die Expansion des Systems bis zum freeze–out [STO86], der dadurch definiert ist, daß es im System keine Stöße mehr gibt.

Aufschluß über die Kompressionseffekte liefern die experimentelle Beobachtung des sogenannten "side splash" und "squeeze out" (Abbildung 2.3).

Gegenwärtig ist man aber noch weit davon entfernt, die dynamischen Prozesse bei einer Schwerionenkollision im Detail bestimmen zu können. Mit Bose–Einstein–Interferometrie, dem sog. Hanbury–Brown/Twiss–Effekt, kann man Aufschlüsse über die beteiligten Wechselwirkungsvolumina bzw. Zeitskalen erhalten und daraus auf die Dichte schließen.

Pionen und K^+–Mesonen ergeben verschiedene Dichten, da die gemessenen K^+–Mesonen aufgrund der größeren mittleren freien Weglänge von Kaonen aus einer früheren Phase der Kollision stammen als die Pionen, die mit viel größerer Wahrscheinlichkeit absorbiert werden.

Aufschluß über die Temperatur erhält man unter gewissen Bedingungen aus den gemessenen Energieverteilungen von beispielsweise Pionen unter einem bestimmten Winkel. Man wählt einen Winkel von 90^o im Schwerpunktystem, weil dort der Spectator–Beitrag am kleinsten ist. Die Form des Spektrums ist etwa exponentiell:

$$E(d^3\sigma/dp^3) \propto exp(-E^{cm}/E_0).$$

Die Größe E_0 kann für ein thermodynamisches System im globalen

Gleichgewicht mit einer Temperatur identifiziert werden, da die Form des Spektrum von einer thermalisierten Quelle durch die Boltzmann–Verteilung gegeben ist.

Auf diese Weise kann man versuchen, über den Vergleich mit dynamischen Theorien (beispielsweise BUU, QMD, hydrodynamische Modelle) für jede emittierte Teilchenspezies experimentelle Punkte in der (ρ, T)–Ebene zu erhalten. Indem man diese Punkte miteinander verbindet, kann man etwas über die Zeitentwicklung einer Schwerionenkollision lernen.

"Inklusive" (Einteilchen–)Spektren zeigen zwar thermische beziehungsweise statistische Eigenschaften, doch ihre Interpretation ist durch die Vielzahl der möglichen Beiträge nicht eindeutig. Die möglichen Reaktionsmechanismen lassen sich aber durch den Nachweis möglichst aller Teilchen und eine konsistente Beschreibung der verschiedenen Teilchenspezies einschränken. Dies strebt man durch den Aufbau eines "exklusiven" (4π–) Experiments an.

Kapitel 3

Der 4π–Detektor

When choosing between two evils, I always like to try the one I've never tried before.

– Mae West

Durch einen möglichst vollständigen Nachweis aller Teilchen, die in einem Kernstoß erzeugt werden, will man Informationen über die sogenannte Zustandsgleichung der Kernmaterie erhalten. Daher wurde an der GSI in Darmstadt begonnen, ein 4π–Experiment aufzubauen, das Detektoren mit hoher Effizienz zum Nachweis geladener Teilchen, wie Protonen, leichte Kernbruchstücke und Pionen enthält.

Der 4π–Detektor ist ein hochmodulares Detektorsystem mit mehreren Detektormodulen [GOB87, GSI1189, GSI0491]. Abbildung 3.1 zeigt eine Seitenansicht des Detektors. Der Detektor besteht aus folgenden Elementen:

- äußere Plastikwand und innere Plastikwand (0–Grad Detektor)
- Ionisationskammern (Parabola) und Rosace
- Zentrale Driftkammer (CDC)
- Helitron
- Plastik Barrel und Čerencov Barrel
- supraleitender Solenoid

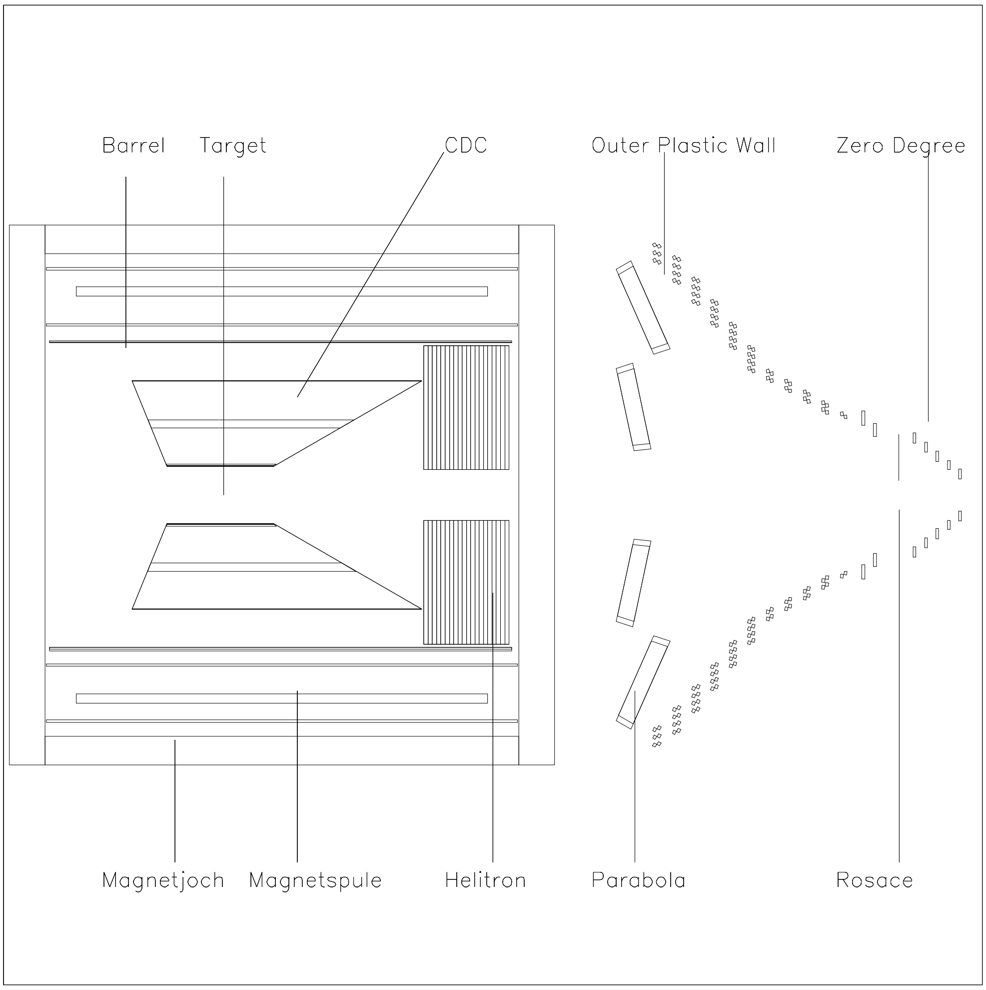

Abbildung 3.1: Seitenansicht des 4π–Detektors

Der Detektor wird in zwei Schritten aufgebaut. Phase 1 besteht aus der **Vorwärtswand**, die den vollen Azimut und einen polaren Winkelbereich von 1^o bis 30^o abdeckt. Die Vorwärtswand besteht aus zwei Teilen:

- **Plastikwand**,
- **Cluster Detector Shell**.

Sie besteht aus insgesamt 764 Szintillator–Streifen. Diese messen die Flugzeit der Reaktionsprodukte relativ zu einem in–beam START Detektor. Die Plastikwand liefert einen schnellen Online Trigger für die Multiplizität geladener Teilchen genauso wie für deren azimutalen oder polaren Verteilung.

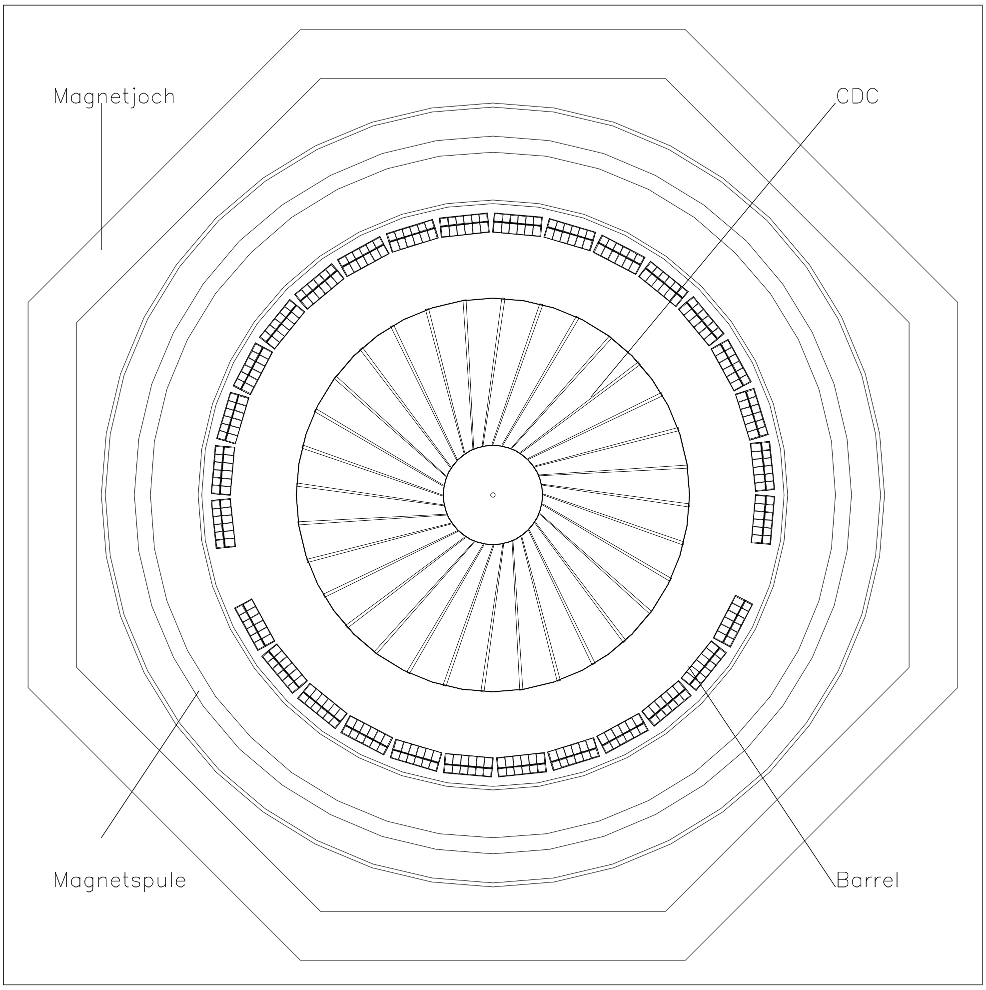

Abbildung 3.2: Querschnitt vom Aufbau des 4π–Detektors

Die beiden Teile sind im einzelnen:

- **Äußere Plastikwand:** Die äußere Plastikwand, die den polaren Winkelbereich von 7^o bis 30^o abdeckt, besteht aus 512 Szintillator–Streifen in 8 Sektoren. Sie werden beidseitig ausgelesen. Die gemessene Position auf dem Streifen liefert den Azimut, den polaren Winkel erhält man durch die Position des Streifens.

- **Innere Plastikwand (0–Grad Detektor):** Der innere Bereich der Vorwärtswand besteht aus 252 trapezförmigen Szintillatoren, die nur einseitig ausgelesen werden. Sie sind in 7 konzentrischen

Ringen um die Strahlachse angeordnet, die den polaren Winkelbereich von 1^o bis 7^o abdecken.

Die Identifizierung der Elemente beruht auf einer $\Delta E - TOF$–Messung. Sie liefert die Ladung Z des Fragments, falls sie nicht im Szintillator gestoppt werden. Dies führt zu einer Minimalenergie für die Teilchen: 45 MeV für Protonen und etwa 85 MeV/u für Kohlenstoffkerne. Eine der Hauptanforderungen an den Detektor ist aber die Messung von schweren Fragmenten ("clusters") als auch von schnellen geladenen Teilchen. Daher gibt es vor der Vorwärtswand noch eine dünnere ΔE–Shell.

Dieser Cluster–Detektor bedeckt den gleichen Winkelbereich von 1^o bis 30^o. Aus technischen Gründen wurden für die beiden polaren Winkelbereiche verschiedene Lösungen verwirklicht:

- **Ionisationskammern (Parabola):** Über 6^o wird eine Wand aus 16 Ionisationskammern benutzt, die in zwei konzentrischen Ringen angeordnet sind und den Outer Plastic Wall abdecken. Jede kammer enthält 8 Anoden. Die resultierende Granularität von 128 Elementen wird als ausreichend für die erwartete Multiplizität schwerer Fragmente angesehen.

- **Rosace:** Für den Winkelbereich von 1^o bis 6^o wird ein Array von 60 dünnen (2mm) Szintillator–Paddles für die ΔE–Messung benutzt.

Phase 2 vervollständigt den Aufbau durch Detektoren zur Aufnahme von Teilchenspuren, sogenannten *Tracking–Detektoren*:

- **Zentrale Driftkammer:** Die zentrale Driftkammer [GOB88] ist eine zylindrische Driftkammer mit einem äußeren Radius von etwa 80 cm und einem inneren von etwa 20 cm.

  Der Vorwärtskonus hat einen Öffnungswinkel von 30^o. Er zeigt genau auf die Targetposition und gibt damit den Teilchen, die mit kleineren polaren Winkeln emittiert werden, den Weg frei. Sie werden dann von den Detektoren in der Vorwärtsrichtung registriert.

  Der Rückwärtskonus hat einen Öffnungswinkel von 60^o. Er war ursprünglich nicht vorgesehen, mußte aber aus Gründen der mechanischen Stabilität eingeführt werden.

  Damit wird der volle Azimut und ein polarer Winkelbereich von 30^o bis etwa 140^o von der zentralen Driftkammer (CDC)abgedeckt.

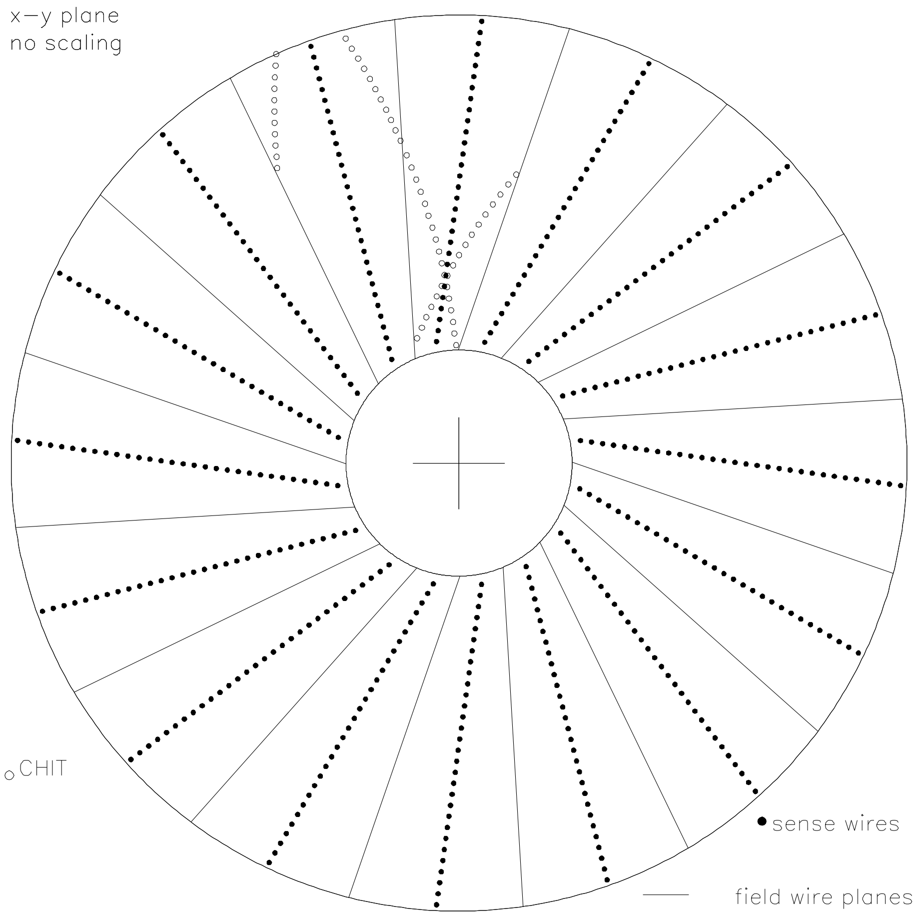

Abbildung 3.3: Spur und Spiegelspur in der CDC: Ein Querschnitt senkrecht zur Strahlachse zeigt die Segmentierung der zylindrischen Driftkammer. Sie besteht aus 16 radialen Sektoren mit 60 Signaldrähten. Sie laufen parallel zur Strahlrichtung und sind in einer Ebene angeordnet. Allerdings werden nur 30 davon ausgelesen (volle Punkte). Die durchgehende Spur von Kreisen zeigt eine Teilchenspur an, die Spiegelspur wird einmal zerrissen, wenn das Teilchen von einem Sektor in einen anderen geht, und außerdem kommt sie aufgrund der geneigten Sektoren nicht vom Target (Kreuz in der Mitte).

Die innere Länge beträgt etwa 86 cm die äußere 190 cm. Die Driftkammer ist aus einer EPOXY–Struktur aufgebaut. Der Innenmantel besteht aber aus einer dünnen Mylarfolie, um die Vielfachstreuung so klein wie möglich zu halten.

Der Querschnitt senkrecht zur Strahlachse des Detektor–Systems in Abbildung 3.2 zeigt die Segmentierung der Driftkammer. Die zentrale Driftkammer ist vom Typ einer Jet–Kammer [KLE87] und ähnlich der JADE–Driftkammer am DESY [DRU80] oder der DIOGENE–Driftkammer [ALA87] in Saclay aufgebaut:

Das zylindrische Volumen der CDC ist in 16 radiale Sektoren aufgeteilt. In jedem Sektor befinden sich 60 Signaldrähte mit einem Abstand von 10 mm. Sie verlaufen parallel zur Strahlachse und sind in einer Ebene angeordnet (Abbildung 3.3), die von innen nach außen zeigt. In dieser Ebene sind die Drähte leicht gegeneinander versetzt, sie weisen ein sogenanntes *Staggering* auf, um die elektrostatische Stabilität der Drahtanordnung zu erhöhen. Ein Charakteristikum der 4π–Driftkammer ist, daß die Sektoren nicht auf das Target ausgerichtet sind, vielmehr sind sie, wie Abbildung 3.2 zeigt, mit einem *Tilt–Winkel* von 8^o verdreht.

Die zentrale Driftkammer enthält ein Gasgemisch von 88% Argon, 10% Methan und 2% Isobutan und arbeitet unter Atmosphären–Druck. Die Elektronen, die von den durchfliegenden Teilchen im Kammergas freigesetzt werden, haben eine Driftgeschwindigkeit von etwa 40 μm/ns.

Das elektrische Feld in der CDC steht orthogonal auf der Ebene der Signaldrähte. Die Driftkammer befindet sich im solenoidalen Magnetfeld eines supraleitenden Magneten mit einer Feldstärke von maximal 6 kG. Die Magnetfeldlinien verlaufen parallel zur Strahlachse und senkrecht zu den elektrischen Feldlinien. Aufgrund der langen Driftwege (maximal 15 cm) macht sich die Lorentzkraft bemerkbar. Die Driftzellen, also die Bereiche, aus denen die Elektronen auf einen Signaldraht driften, sind daher gegen die Drahtebene um einen Lorentzwinkel von etwa 20^o verdreht.

Die Signaldrähte werden an beiden Enden ausgelesen. Aus den Ankunftszeiten wird die Driftzeit bestimmt und daraus die x– und y–Koordinate. Da man nicht weiß, von welcher Seite der Driftzelle die Elektronen auf den Draht gekommen sind, gibt es zu jedem Punkt noch einen sogenannten *Spiegelpunkt* und daher zu jeder Spur noch eine *Spiegelspur*. Aber aufgrund des Tiltwinkels bietet sich eine einfache Möglichkeit, Spiegelspuren in der Phase der Spurerkennung zu verwerfen, da die Spiegelspur nicht aus der Targetregion kommt.

Das ist in Abbildung 3.3 illustriert. Die z–Koordinate wird über Ladungsteilung berechnet.

Um die Kosten für die Elektronik zu reduzieren, werden von den 60 Signaldrähten aber nur 30 benutzt, so daß im allgemeinen 30 Punkte pro Track über eine Strecke von mehr als 60 cm gemessen werden. Auf diese Weise erhält man außerdem eine Ausweichmöglichkeit für eventuell gerissenen Drähte. Zusätzlich kann man versuchen, eine Gruppierung der auszulesenden Drähte zu finden, die eine bessere Impulsauflösung ermöglicht.

Die Ortsauflösung soll in der XY–Ebene $\sigma_{r\phi} = 300\mu m$ und in z–Richtung $\sigma_z = 20mm$ betragen ($\approx 1\%$ der Drahtlänge). Angestrebt wird eine Doppelspurauflösung von 5-10mm.

Die zentrale Driftkammer erlaubt aber nicht nur die räumliche Rekonstruktion von gekrümmten Teilchenspuren und damit eine Impulsmessung, sondern bietet auch die Möglichkeit, den differentiellen Energieverlust dE/dx zu bestimmen. Die dE/dx–Auflösung sollte 5-10% betragen. Indem man dE/dx gegen den Impuls aufträgt ist eine Teilchenidentifizierung möglich.

- **Helitron:** Die HELITRON–Driftkammer erfüllt die gleichen Aufgaben wie die zentrale Driftkammer für den polaren Winkelbereich von 7^o bis 30^o. Es werden Helix–Trajektorien für geladene Teilchen aufgenommen, die in Vorwärtsrichtung emittiert und vom Magnetfeld abgelenkt werden. Eine Impulsauflösung von $\Delta p/p \leq 4\%$ für Teilchen–Korrelation–Messungen wird angestrebt. Mit Hilfe der Flugzeitinformation von der Vorwärtswand kann man eine Teilchenidentifikation durchführen, insbesondere hofft man $\pi^{\pm}$, p, d, t, ^{3}He und ^{4}He trennen zu können.

Teilchen, die mit polaren Winkeln größer als 30^o emittiert werden, sind hauptsächlich Isotope von H und He, oder es handelt sich um Pionen. Aufgrund der speziellen Eigenschaften von Kaonen (strangeness, lange mittlere freie Weglänge), ist es von großem Interesse, Kaonen zu messen. Da diese aber bei SIS–Energien selten sind, müssen zusätzliche Anstrengungen unternommen werden. Daher wird die CDC umgeben von einem Barrel–Detektor, der wiederum aus zwei Teilen besteht:

- **Plastik Barrel:** Der Plastik Barrel ist eine TOF–Detector–Shell aus 180 Plastik–Szintillator–Streifen. Sie sind 240 cm lang, 3.2 cm breit, 4 cm tief und werden an beiden Enden ausgelesen. Diese

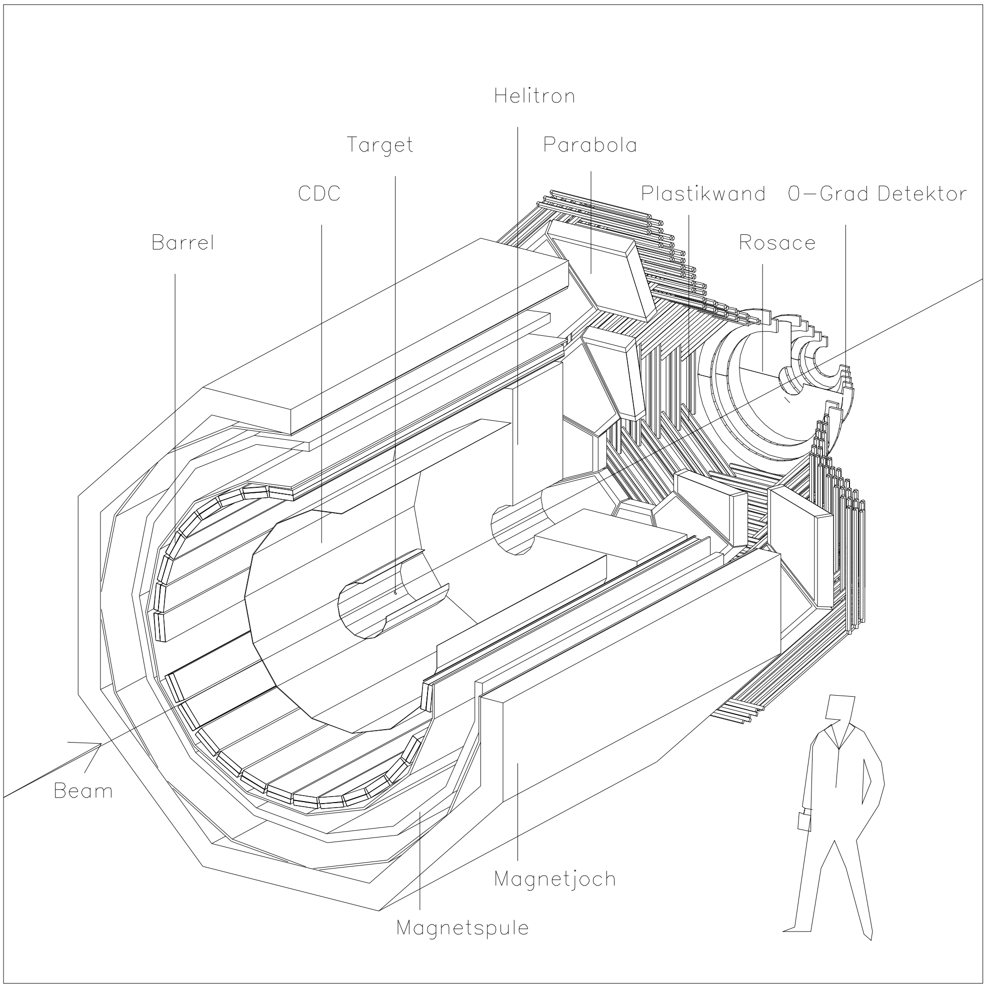

Abbildung 3.4: 3D–Ansicht vom Aufbau des 4π–Detektors

TOF–Information kann zusätzlich zum differentiellen Energieverlust der CDC benutzt werden, um die Teilchenidentifikation zuverlässiger zu machen.

- **Čerencov Barrel:** Der TOF–Detektor wird von einer zweiten Schale umgeben. Diese besteht aus einer gleichen Anzahl von wassergefüllten Čerencov–Zählern, die 3cm dick sind. Ein spezielles Salz dient als Wellenlängenschieber. Die Schwelle für Kaonen ist 560 MeV/c. Die einzelnen Elemente beider Detektoren sind zu 30 Modulen gruppiert, wovon jedes 6 TOF– und 6 Čerencov–Zellen enthält.

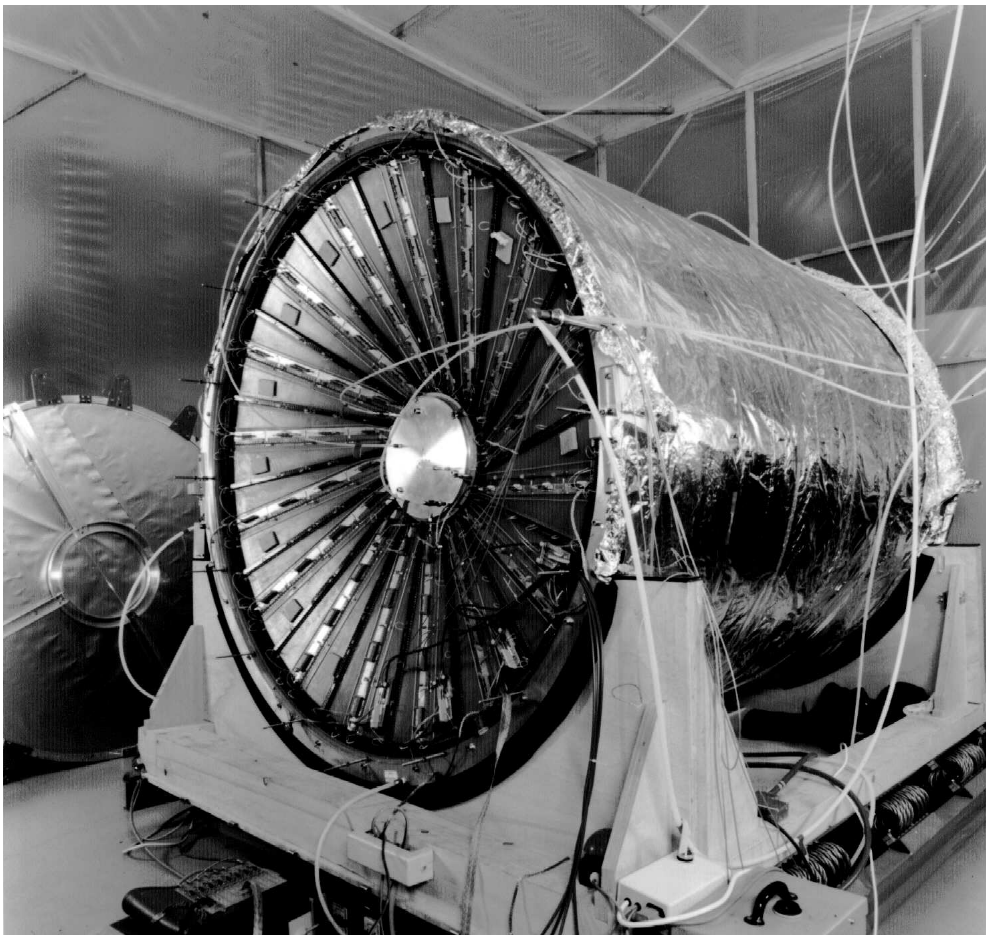

Abbildung 3.5: Testaufbau der zentralen Driftkammer in einem Zelt in der Targethalle der GSI.

Alle diese Detektoren arbeiten in einem Magnetfeld von etwa 0.6 T, das von einem **supraleitenden Solenoiden** erzeugt wird. Das Magnetfeld ist im Bereich der CDC auf 2% homogon, so daß man in guter Näherung von Helix–Trajektorien für die geladenen Teilchen ausgehen kann. Die dreidimensionale Abbildung 3.4 verschafft einen besseren Eindruck vom ganzen Detektor, insbesondere von der Anordnung der Barrel–Module.

Mit dem Aufbau von Phase 2 und der Einführung von Tracking–Detektoren wurde eine neue Form von Analyse–Software nötig – Pattern Recognition für die Driftkammern.

Im allgemeinen wird Pattern Recognition als schwieriges Problem betrachtet in dem Sinne, daß es keine allgemeinen und in irgendeiner Weise optimalen Strategien gibt, um ein gegebenes Problem zu lösen.

Dies trifft auch für die zentrale Driftkammer des 4π–Experimentes an der GSI zu, obgleich bereits viel Erfahrung mit Pattern Recognition für Driftkammern in der Hochenergiephysik gesammelt wurde. Das folgende Kapitel stellt die wichtigsten Ansätze zur Spurerkennung vor. Diese haben sich für die bis dahin erreichten Trackmultiplizitäten in Driftkammern als erfolgreich erwiesen.

Bei hadronischen Reaktionen und insbesondere Schwerionenkollisionen wird man aber mit viel höheren Trackmultiplizitäten als bisher konfrontiert. Daher haben sich die Ansprüche an das Auflösungsvermögen und den dynamischen Bereich erhöht. Gleichzeitig wurden die Detektoren in den letzten Jahren verbessert.

Damit wurde das Argument, Drahtkammern liefern entweder zu wenige oder zu ungenaue Meßwerte von gekrümmten Trajektorien, um globale Pattern Recognition Methoden anwenden zu können, obsolet:

In dieser Arbeit wird die Leistungsfähigkeit der sogenannten *Hough–Transformation* für das Pattern Recognition in Jet–Kammern vorgestellt. Damit werden die Tracking–Möglichkeiten auf viel höhere Track–Dichten erweitert, als das mit den bisher zur Verfügung stehenden Methoden möglich war.

Die Stärke der in dieser Arbeit entwickelten Implementierung der Hough–Transformation, die sogenannte *Adaptive Hough–Transformation*, besteht darin, daß sie die vom Detektor zur Verfügung gestellte globale Information ausnutzt, um Tracks mit kleinem Abstand im Phasenraum aufzulösen, eine wichtige Eigenschaft für Anwendungen zum Beispiel in der Pionen–Interferometrie. Diese Eigenschaften bleiben selbst bei 100% Untergrund erhalten. Dies wird einmal in einem Test mit Simulationen gezeigt und dann durch die Anwendung auf die ersten experimentellen Daten bestätigt.

Kapitel 4

Methoden der Spurerkennung

The golden rule is that there is no golden rule.
– George Bernard Shaw

4.1 Die Aufgabe der Spurerkennung

Ist eine Menge von Ortsmessungen in einem Detektor gegeben, so ist es die Aufgabe der Spurerkennung, des *Track–Finding*, diese Menge in Teilmengen, sogenannte Klassen, aufzuspalten, so daß

1. jede Klasse Meßwerte enthält, die von einem Teilchen stammen könnten, und

2. eine Klasse alle Meßwerte enthält, die nicht einem bestimmten Teilchen mit gewisser Sicherheit zugeordnet werden konnten.

Diese Definition gibt ein realistisches Ziel vor. Sie reduziert Track–Finding zu einem *Cluster–Analyse–Problem.* Dabei ist ein *Cluster* folgendermaßen definiert [AND72]:

> *A cluster is loosely defined as a collection of vectors or points that are close together.*

Daher ist man in der *Clusteranalyse* hauptsächlich damit beschäftigt, passende Abstandsdefinitionen zwischen Paaren von Objekten zu entwickeln und Algorithmen aufzustellen, die mit dieser Definition arbeiten.

Der *minimum spanning tree* Algorithmus gehört hierher und wird später noch genauer beschrieben, da er eine der wenigen Beispiele ist, eine nicht–heuristische Methode auf *real–life*–Track–Finding anzuwenden.

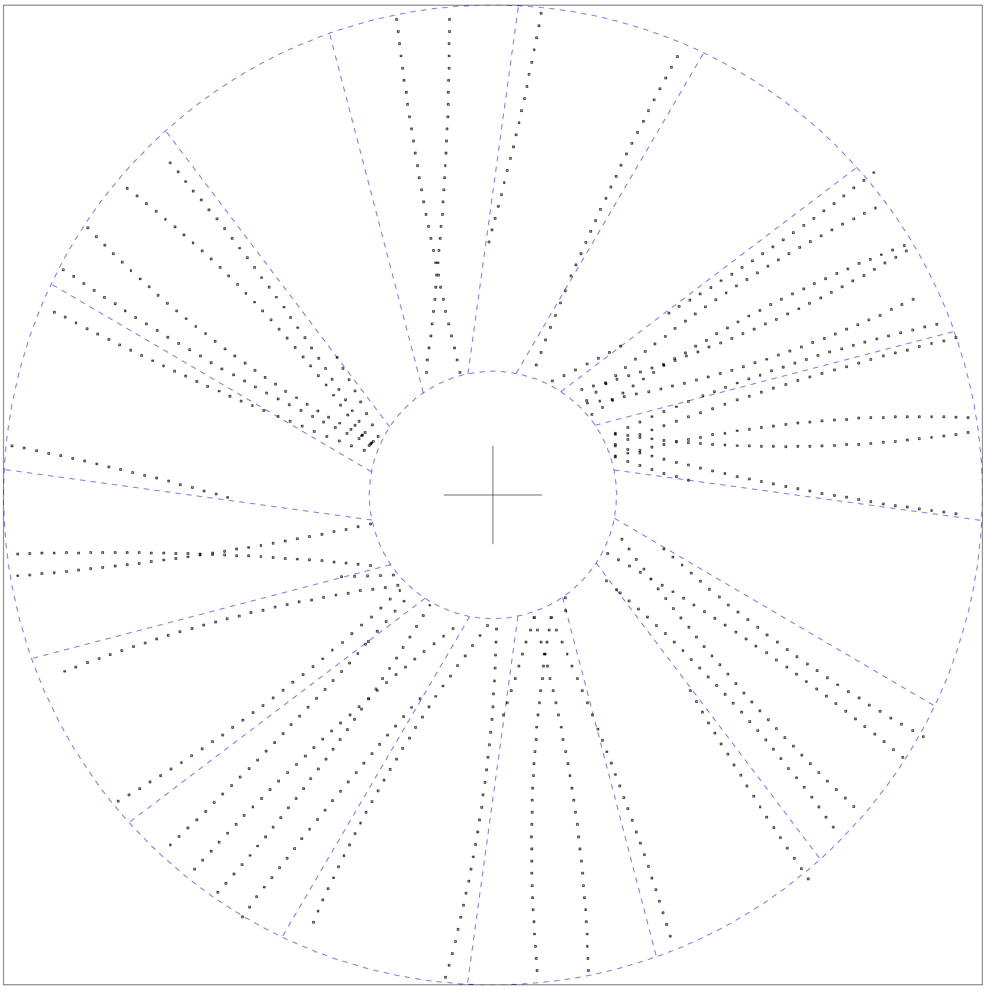

Abbildung 4.1: Beispiel für ein Ereignis niedriger Multiplizität: Ca+Ca Ereignis bei 1GeV/u. Die Tracks sind Kreisbögen, die durch den Koordinatenursprung gehen.

4.2 Methoden für die Spurerkennung

Alle Methoden bestehen aus zwei Schritten:

- Im ersten Schritt wird eine Teilmenge von Meßwerten gewählt, ein sogenannter *Track–Kandidat* (eigentliche Spurerkennung).
- Im zweiten Schritt wird untersucht, ob die Teilmenge ein akzeptabler Track ist, unter Umständen wird eine Spuranpassung (*Fit*) durchgeführt.

Beides zusammen wird als *Tracking* bezeichnet.

Diese Zwei–Schritte–Prozedur wird oft noch in weitere Schritte unterteilt, um das Verfahren zu beschleunigen. In einer Initialisierungsphase wird eine bestimmte Anzahl von Meßwerten ausgewählt und entweder akzeptiert oder verworfen. Werden sie akzeptiert, so werden weitere Meßwerte dazugenommen und diese neue Gesamtmenge untersucht. Diese Vorgehensweise wird wiederholt, bis man am Schluß einen Endtest auf den vollständigen Track anwendet.

Der Geschwindigkeitsvorteil gegenüber der einfachen Zwei–Schritt–Methode, die erst den vollständigen Trackkandidaten auswählt und dann die Entscheidungsfunktion anwendet, beruht auf zwei Gründen:

- Erstens hat die Reduktion der Meßwerte pro Trackkandidat eine beträchtliche Reduktion der Anzahl dieser Kandidaten zur Folge.

- Zweitens kann die Anwendung der endgültigen Entscheidungsfunktion ziemlich zeitaufwendig sein, zum Beispiel, wenn ein iterativer Fit mit einem Trackmodell ausgeführt werden soll.

Eine Folge von Enscheidungsfunktionen, angefangen mit einer einfachen aber schnellen, bis hin zu einer präzisen, aber langsamen Methode, am Ende, verbessert die Geschwindigkeit, wenn in jedem Zwischenschritt ein Großteil der falschen Trackkandidaten verworfen wird. Dabei muß darauf geachtet werden, daß kein guter Trackkandidat durch irgendeinen der einfachen approximativen Tests verworfen wird.

Wenn für eine gegebene Methode alle möglichen Tracks gefunden wurden, bleibt für gewöhnlich eine Anzahl von Punkten zurück, die keinem Track zugeordnet werden konnten.

Wenn verschiedene Methoden angewendet werden, oder dieselbe Methode in einer anderen Art und Weise, ist ein gutes Maß für die *Effizienz* notwendig. Ein solches Maß muß von Experiment zu Experiment neu gewählt werden.

4.3 Die kombinatorische Methode

Die einfachste Methode, in einer Punktmenge Teilchenspuren zu finden, besteht in der vollständig kombinatorischen Methode. Diese Methode arbeitet folgendermaßen:

1. teile die Menge aller Ortsmessungen in alle möglichen Teilmengen auf und

2. passe jede Teilmenge (=Trackkandidat) an ein Trackmodell an, nenne diese einen Track, falls der Fit gut ist.

Obwohl diese Methode in einigen einfachen Fällen, bei denen man es mit wenigen Punkten zu tun hat, angewendet werden kann, ist sie für die meisten praktischen Anwendungen zu zeitaufwendig.

Eine der Aufgaben ist es aber, den Zeitaufwand so gering wie möglich, dabei jedoch die Effizienz so hoch wie möglich zu halten. Das ist nicht immer einfach.

4.4 Klassifizierung der Methoden

Außer der vollständig kombinatorischen Methode, können alle weiteren Methoden zur Spurerkennung als *lokal* oder *global* klassifiziert werden.

Eine Methode heißt *global*, wenn alle Punkte durch den Algorithmus in der gleichen Art und Weise behandelt werden. Der Algorithmus produziert eine Liste von Tracks, oder wenigstens eine Liste, in der die Tracks einfacher gefunden werden können als unter den Originaldaten. Der Algorithmus kann deshalb als eine Transformation aller Koordinaten betrachtet werden. Der Zeitaufwand für eine globale Methode ist daher im Prinzip proportional zur Anzahl N der Punkte in einem Ereignis.

Im Gegensatz dazu wird bei einer *lokalen Methode* ein Trackkandidat nach dem anderen behandelt. Man fängt mit einigen wenigen Punkten an. Das ist die Track–Kandidat–Initialisierung. Dann sammelt man weitere Punkte, zum Beispiel durch Interpolation oder Extrapolation des benutzten Trackmodells. Werden zusätzliche Punkte gefunden, so werden diese zum Trackkandidaten hinzugenommen, sonst wird der Trackkandidat nach einer bestimmten Anzahl von Versuchen verworfen. Da lokale Methoden immer fruchtlose Versuche machen müssen, um einen Trackkandidaten zu finden, und so die gleichen Punkte in verschiedenen Kombinationen immer wieder benutzen, wächst der Zeitaufwand schneller als linear mit der Anzahl der Punkte.

Globale Methoden sind unabhängig davon, in welcher Ordnung die Punkte in den Algorithmus eingehen. Das trifft für lokale Methoden nicht zu, da die Behandlung jedes Punktes von der Initialisierung abhängt und vom bisherigen Verlauf der Spurerkennung.

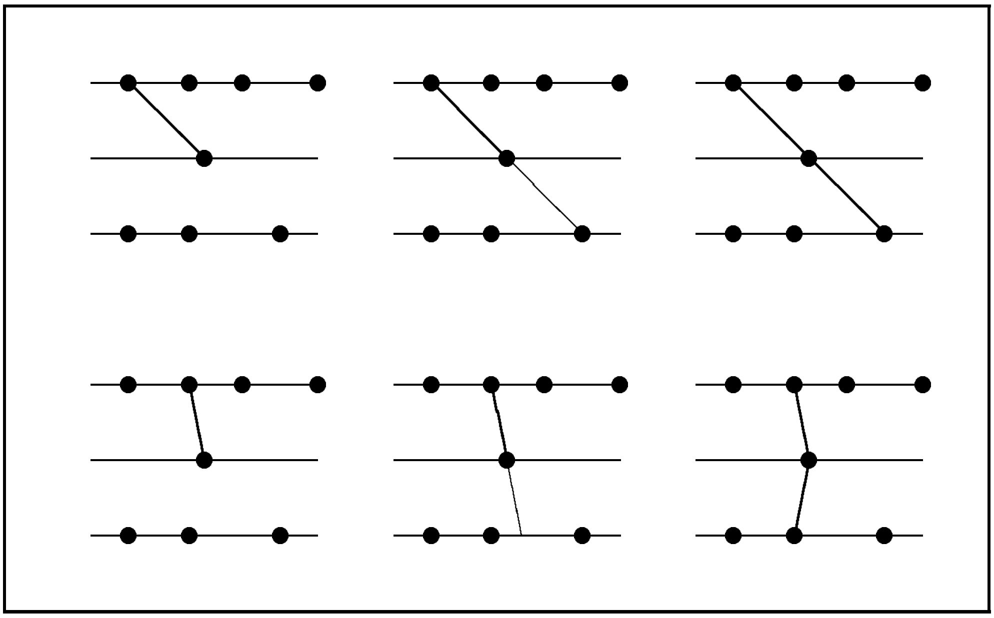

Abbildung 4.2: Die Track–Following–Methode: Die Track–Following–Methoden benutzen eine kombinatorische Initialisierung der Track–Kandidaten um die ersten drei Punkte eines Tracks zu finden. In der obigen Abbildung werden zwei Beispiele gezeigt (erste und zweite Reihe), die mit linearer Extrapolation arbeiten.

4.5 Lokale Methoden

4.5.1 Die Track–Following–Methode

Bei dieser Methode wird am Anfang ein Track–Segment, ein sogenanntes *tracklet,* gewählt, das aus einigen wenigen Punkten, drei oder vier, besteht. Dieses Tracklet wird so entfernt wie möglich von der primären Wechselwirkungszone gewählt, weil dort die Tracks im allgemeinen weitaus separierter sind als anderswo.

Im nächsten Schritt wird durch Extrapolation in die Richtung der primären Wechselwirkungszone ein weiterer Punkt gesucht. Diese Extrapolation kann von nullter Ordnung sein wie in Abbildung 4.2, indem man einfach den nächsten Nachbarpunkt wählt, von erster Ordnung, indem man mit einem Geradenstück weitergeht, oder von zweiter Ordnung, indem man eine parabelförmige Fortsetzung wählt, oder irgendeine höhere Ordnung, oder indem man irgendein Trackmodell benutzt, zum Beispiel einen Kreis oder eine Helix. Das Ziel ist, in jedem Fall einen schnellen Algorithmus zu haben, um weitere Punkte zu finden. Die Parabelextrapolation reicht in vielen Fällen mit Magnetfeld aus, da sie das Vorzeichen der Krümmung erhält und so das Verhalten einer richtigen Teilchenspur wiedergibt. Außerdem ist diese Methode sehr schnell, da eine Parabel durch drei Punkte geschrieben werden kann als

$$y = a_1(x)y_1 + a_2(x)y_2 + a_3(x)y_3. \tag{4.1}$$

Dabei hängen die Koeffizienten nur von den Drahtpositionen in der Kammer ab, so daß sie vorher nur einmal berechnet werden müssen.

Dieser als *Track–Following–Methode* bezeichnete Ansatz benötigt nur ein lokales Trackmodell, da er nur nach den nächsten Punkten sucht, wobei die bisher gefundenen dazu benutzt werden, das Trackmodell zu extrapolieren. Diese Vorgehensweise erlaubt ein einfaches und daher schnelles Trackmodell. Werden aber andererseits die Abstände zwischen den Drähten zu groß, so ist das approximative Modell nicht präzise genug, und wegen den Meßfehlern wird selbst ein absolut korrektes Tracking, das mit bereits gefundenen Punkten arbeitet, problematisch sein.

Die Track Following Methode benutzt eine kombinatorische Initialisierung der Trackkandidaten. Wenn dann einmal einige Punkte zu diesem Kandidaten hinzugefügt wurden, so ist dies dann schon sehr wahrscheinlich ein guter Track und der *computational overhead* ist ziemlich gering. Dementsprechend ist der Zeitaufwand proportional einer Zahl zwischen N und N^2 [1].

Beispiele für die Track–Following–Methode findet man beim ALEPH am CERN [NOR88], beim DIOGENE in Saclay [POI83], oder auch [TRZ88].

4.5.2 Die Track–Road–Methode

Bei der sogenannten *Track–Road–Methode* wird keine Extrapolation wie bei der *Track–Following–Methode* benutzt, sondern die viel präzisere Interpolation, um weitere Punkte für einen Track zu finden. Man wählt Anfangspunkte an beiden Enden des Tracks und einen in der Mitte im Falle gekrümmter Tracks. Ein einfaches Trackmodell wird dann benutzt, um weitere Punkte für diesen Track zu finden, indem man eine *Road* gewisser Breite um das Trackmodell legt. Dieses Trackmodell kann exakt sein, wie

[1] N = Anzahl der Punkte

zum Beispiel ein Kreis im Falle eines homogenen solenoidalen Magnetfeldes, oder eine Gerade in einem feldfreien Raum, oder es kann approximativ sein. In diesem Fall muß man die Breite der Road mit Monte–Carlo–Tracks bestimmen. Im Prinzip kann die Road um so enger sein, je besser das Trackmodell ist. In der Praxis kann man allerdings aufgrund systematischer Fehler in den Ortsmessungen und wegen des Untergrundes nicht die theoretische Breite von 3σ der Detektorauflösung wählen. Die Track–Road–Methode ist langsamer als die Track–Following–Methode, aber manchmal ist sie die einzige Methode, die angewendet werden kann, besonders im Falle räumlich weit auseinanderliegender Detektoren. Die meisten modernen Detektoren liefern allerdings eine genügend hohe Dichte an Meßwerten, so daß eine schnellere Methode angewendet werden kann.

Da in einem Magnetfeld drei Punkte für die Initialisierung gebraucht werden, muß man wegen Detektorineffizienzen die verschiedensten Kombinationen benutzten. Die meisten Kombinationen werden dabei falsch sein, so daß der Zeitaufwand dieser Methode typischerweise proportional einem Faktor zwischen N^2 und N^3 ist.

Diese Methode wurde auch beim DIOGENE in Saclay verwendet [POI83].

4.5.3 Die Track–Element–Methode

Bei der *Track–Element–Methode* wird der Track in zwei Schritten konstruiert. Zuerst werden mit wenigen Punkten kurze Trackelemente (Tracklets) gebildet. Diese werden bei modular aufgebauten Detektoren in natürlichen Untereinheiten des Detektors konstruiert, bei einer Driftkammer zum Beispiel in einer Driftzelle. Von diesem Trackkandidaten ausgehend wird eine Extrapolation oder Interpolation nullter Ordnung (nächster Nachbar), erster Ordnung (Geradenstück) oder zweiter Ordnung (Parabelextrapolation) ausgeführt, um Trackelemente zu finden. In einem zweiten Schritt werden dann diese Elemente mit der Track–Following–Methode oder irgendeiner anderen Methode zur Spurerkennung zu einem vollständigen Track kombiniert (Abbildung 4.3).

Der große Vorteil dieser Methode ist ihre Geschwindigkeit im Vergleich zu Methoden, die alle Punkte mehr oder weniger direkt verwenden. Sie ist deshalb geeignet für Detektoren mit einer hohen Punktdichte. Außerdem kann bei Driftkammern das Problem der Spiegelpunkte auf der Trackelement–Ebene gelöst werden. Ursprünglich wurde sie für die Analyse von Blasenkammerbildern verwendet. Anwendung fand sie bei der

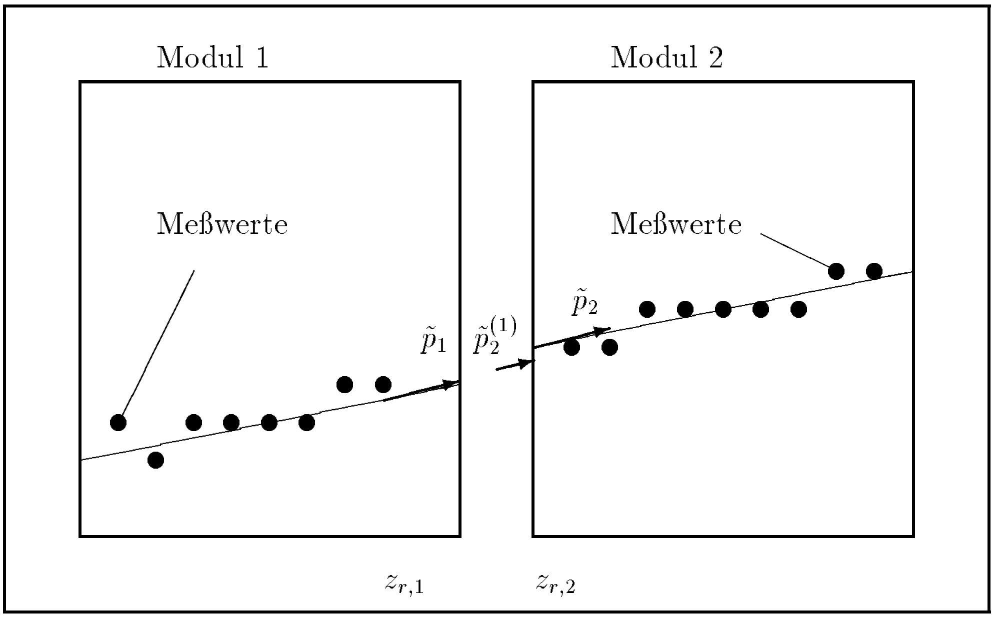

Abbildung 4.3: Die Track–Element–Methode: Die Track–Parameter $\tilde{\mathbf{p}}_1$ werden zu der neuen Referenzfläche propagiert. $\tilde{\mathbf{p}}_1^{(1)}$ und $\tilde{\mathbf{p}}_2$ werden als direkte Messungen der wahren Trackparameter an der Referenzfläche $z_{r,2}$ betrachtet.

JADE Driftkammer am DESY [OLS80]. Eine ähnliche Methode wurde beim UA1 Detektor am CERN benutzt[PIM85].

4.6 Globale Methoden

Einen ganz anderen Weg gehen die globalen Methoden, die auch auf dem größeren Gebiet der *Cluster–Analyse* Verwendung finden. In diesem Fall werden alle Punkte in der gleichen Art und Weise behandelt.

4.6.1 Die Histogramm–Methode

Bei den Histogramm–Methoden benutzt man Funktionen der Punktkoordinaten. Die Werte dieser Funktionen werden in ein Histogramm eingetragen. Die Tracks sollten dann *Cluster* oder *Peaks* in diesen Histogrammen bilden. Diese sollten dann einfacher zu finden sein als die Tracks in

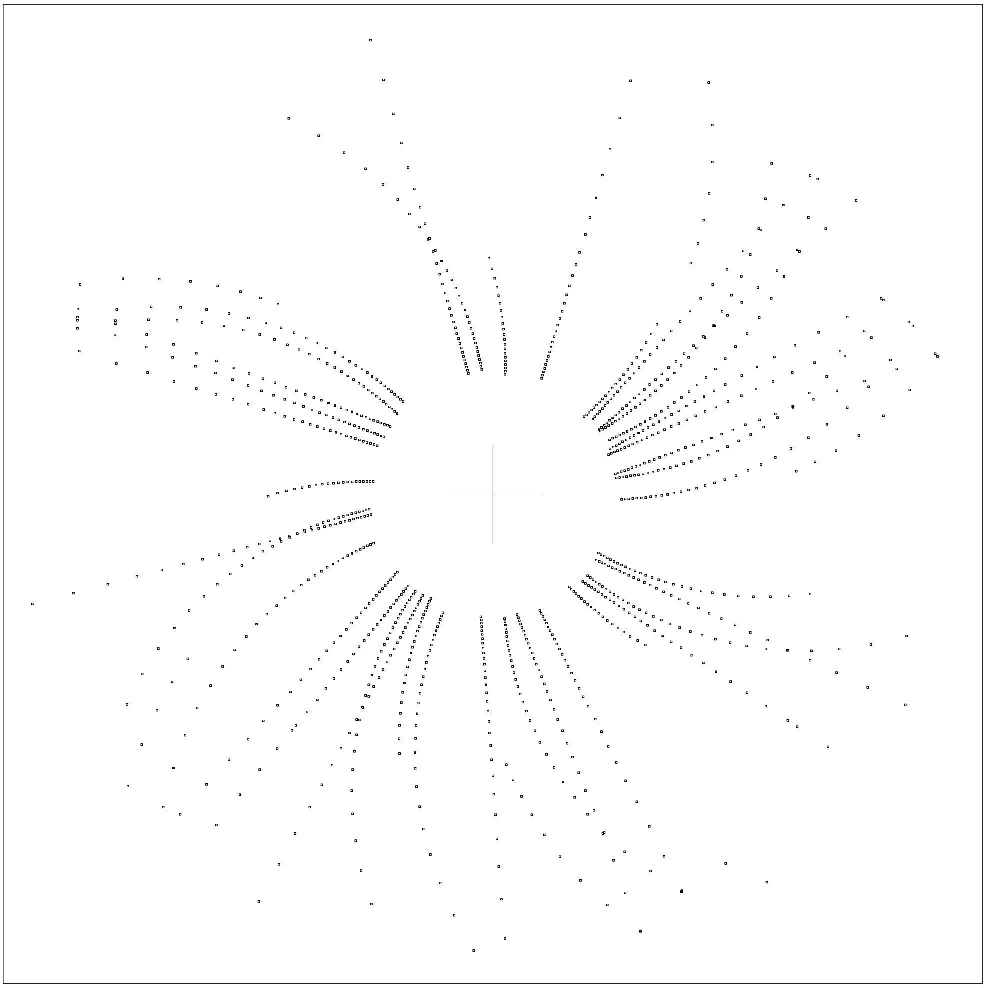

Abbildung 4.4: Die Kreisbögen nach der konformen Abbildung: Die ursprünglichen Kreisbögen sind in Abbildung 4.1 zu sehen. Der Abstand einer Geraden zum Koordinatenursprung ist proportional zur Krümmung des ursprünglichen Kreisbogens.

---

dem ursprünglichen Punktemuster, dem *Hit–Pattern*.

Ein Beispiel für eine solche Methode ist die *inverse konforme Abbildung* [HAN87, HAN88], die man anwenden kann, um kreisförmige Teilchenspuren, die alle durch den Ursprung (= primärer Wechselwirkungspunkt) verlaufen, zu finden:

$$u = \frac{x}{x^2 + y^2}, \tag{4.2}$$

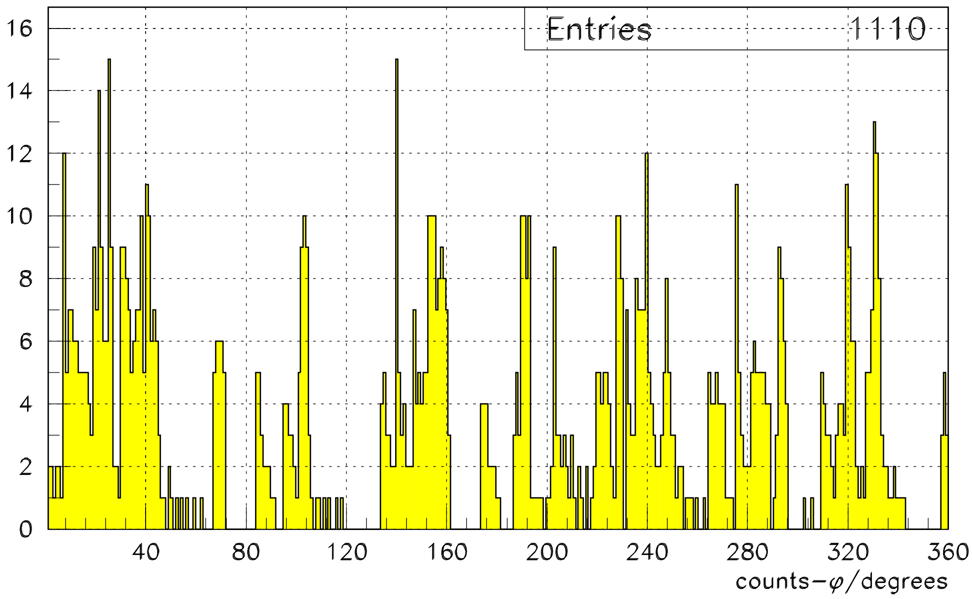

Abbildung 4.5: ϕ–Histogramm der Tracks in Abbildung 4.4. Die Spuren mit hohem Transversalimpuls p_T bilden *Cluster* oder *Peaks*, die leicht gefunden werden können.

$$v = \frac{y}{x^2 + y^2}. \tag{4.3}$$

Diese Transformation erzeugt Geradenstücke in der uv–Ebene (Abbildung 4.4). Der kürzeste Abstand vom Ursprung ist dabei gegeben durch

$$d = \frac{1}{2R}, \tag{4.4}$$

mit R = Trackradius. Die Histogramm–Methode, angewendet auf die konforme Abbildung, toleriert einen gewissen Abstand der Spur vom Ursprung und wird die Spuren immer noch korrekt auffinden (Abbildung 4.5). Diese Methode wurde am CERN im Experiment R807 angewendet [DAH79].

4.6.2 Das Template Matching

Diese Methode erfordert eine Liste aller möglichen Tracks und kann daher nur angewendet werden, wenn diese Zahl nicht zu hoch wird. In diesem Fall kann eine binäre Suche oder ein Hashalgorithmus sehr schnelles

0000000000	0000000000	0000000000	0000000000
0001111000	0000001000	0001111000	0001111000
0010000100	0000011000	0010000100	0010000100
0100000010	0000101000	0100000100	0000000100
0100000010	0001001000	0000000100	0000000100
0100000010	0010001000	0000000100	0000000100
0100000010	0000001000	0000001000	0000001000
0100000010	0000001000	0000010000	0000001000
0100000010	0000001000	0000100000	0000000100
0100000010	0000001000	0001000000	0000000100
0100000010	0000001000	0010000000	0000000100
0010000100	0000001000	0111111100	0010000100
0001111000	0000001000	0000000000	0001111000
0000000000	0000000000	0000000000	0000000000

Tabelle 4.1: Ein Beispiel für template matching: Templates für die vier Ziffern 0, 1, 2 und 3. Sie werden dargestellt durch eine Verteilung der Einsen in einem 10 mal 14 Gitter mit Nullen.

Trackfinding liefern. Diese Methode wurde für den MARK III Detektor am SPEAR [BEC85] verwendet.

Jede Kombination von Drähten, die gefeuert haben, erzeugt ein bestimmtes Muster (*template*). Die Unterteilung des Detektors in Zellen führt zu einer kleinen Zahl von vernünftigen Kombinationen. Bereiche, wo Tracks nahe zusammenkommen, führen zu Konfusion und müssen mit *klassischen* Methoden (Kombinatorik zum Beispiel) aufgelöst werden.

Diese Methode ist besonders geeignet für zylindrische Driftkammern, da die Einteilung des Detektors in Zellen eine natürliche Basis für diesen Algorithmus bietet.

Im Falle des MARK III Detektors funktioniert die Methode im wesentlichen folgendermaßen: Zu Beginn wird eine *cell image matrix* gefüllt. Eine Kombination von Zellen, die gefeuert haben, wird dann mit diesem *Dictionary* verglichen. Diese Methode ist global, außer in Regionen, wo es einen Überlap der Tracks gibt.

Template matching wird oft als schnelle Trigger–Logik implementiert, so zum Beispiel beim CELLO–Detektor am DESY [BEH81].

Für das Experiment E711 am Fermilab wurde ein template matching Algorithmus auf einem schnellen Vektor–Prozessor implementiert [GEO86].

Charakteristisch ist, daß die Koordinaten eines Objektes nur die Werte 0 oder 1 annehmen können, wie in dem Beispiel in Abbildung 4.1. Dort werden die vier Zahlen 0, 1, 2 und 3 gezeigt, die auf ein Gitter abgebildet wurden. Die Ziffern 0 und 1 zeigen die Grauwerte Weiß und Schwarz an.

In ähnlicher Weise ergibt die Messung eines Objektes einen Vektor $\mathbf{x}$. Wird die Klasse durch m Prototyp–Templates $\mathbf{y_m}$ beschrieben, muß ein Vergleich durchgeführt werden, um das Template zu finden, das am besten paßt. Diese Prozedur wird *template matching* genannt und besteht einfach aus einer Vektormultiplikation

$$n_m = \mathbf{y_m^T} \cdot \mathbf{x}. \tag{4.5}$$

m läuft dabei über alle Templates. Gruppiert man alle template Vektoren in eine Matrix Y, reduziert sich die ganze Prozedur zu einer einzigen Matrixmultiplikation:

$$\mathbf{n} = \mathbf{Y^T x}. \tag{4.6}$$

Die maximale Komponente von $\mathbf{n}$ zeigt die beste Übereinstimmung an.

4.6.3 Der Minimum–Spanning–Tree

Um die *Minimum–Spanning–Tree–Methode* zu verstehen, muß man einige Begriffe aus der *Graphentheorie* kennen. Ein *Graph* besteht aus *Knoten* und *Kanten*. Die Knoten können zum Beispiel Punkte im Ortsraum sein, die Kanten könnten durch die Verbindungslinien dargestellt werden. Ein *Baum* beziehungsweise *tree* ist ein Graph ohne Schleifen. Ein *spanning tree* (Abbildung 4.6) ist ein zusammenhängender Graph ohne Schleifen. Ein *minimum spanning tree* ist ein Baum in einem Graphen, für den die Summe der Kantengewichte minimal für alle zusammenhängenden Bäume in diesem Graphen ist.

Bäume für ein Ereignis wie in Abbildung 4.1 kann man konstruieren, indem man die mit dem Detektor gemessenen Koordinaten, die sogenannten *Hits*, mit den Knoten identifiziert und den Abstand zwischen Paaren von Hits mit den Kantengewichten. Vereinzelte Hits erscheinen für gewöhnlich als Knoten mit nur einem Nachbar und können daher leicht eliminiert werden. Tracks können stückweise gefunden und zu vollständigen Tracks kombiniert werden, indem man Ähnlichkeitskriteria verwendet, wie zum Beispiel gemeinsame Punkte, ähnliche Steigung oder Krümmung.

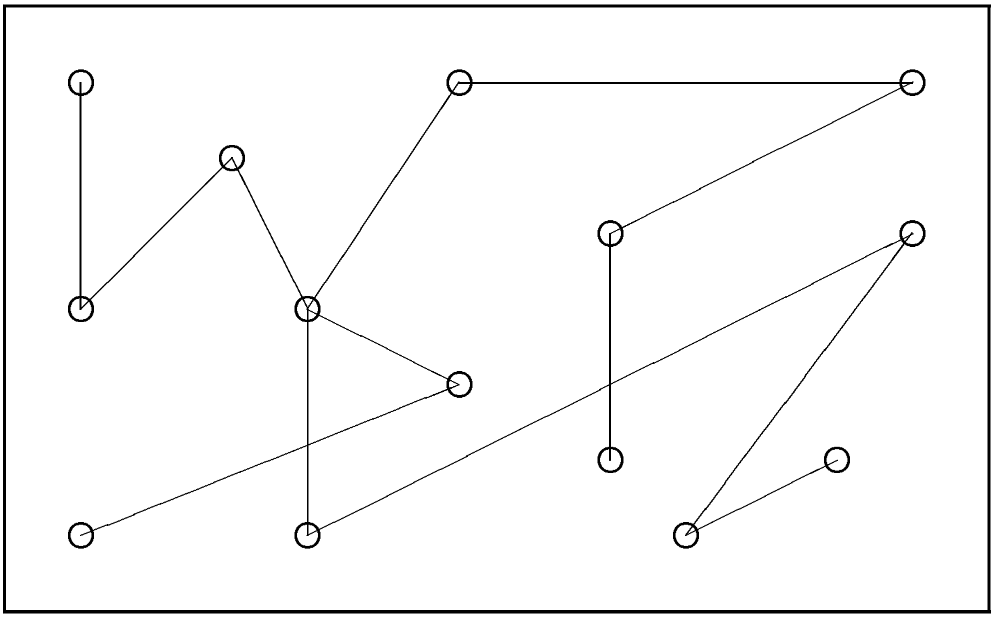

Abbildung 4.6: Ein spanning tree: Jeder Knoten ist mit mindestens einem anderen Knoten verbunden. Dies ist äquivalent zu der Tatsache, daß kein geschlossener Pfad (Loop) existiert.

Eine Variation dieser Methode wurde für den TASSO Detektor am DESY verwendet [CAS81]. Es konnte gezeigt werden, daß diese Methode effizienter als die Road–Methode ist. Das Basiselement in diesem Algorithmus ist nicht ein Punkt, sondern ein Punktepaar. Dieses Paar hat einen gewissen Abstand und eine bestimmte Richtung. Diese Paare werden zu einem Graphen kombiniert, falls sie gemeinsame Punkte und ähnliche Richtung haben. Die zweite Bedingung ist besonders effektiv, um Spiegelpunkte in einer Driftkammer (Links–Rechts–Ambiguität) zu verwerfen.

Auf diese Weise werden Track–Segmente mit einer minimalen Länge konstruiert. Diese erlauben es, Track–Parameter (Kreisradius und Kreismittelpunkt) zu bestimmen. Diese Track–Segmente werden dann zu vollen Tracks gruppiert, indem man Ähnlichkeitskriteria basierend auf den Segment–Parametern verwendet.

Kapitel 5

Einführung der Hough–Transformation

Die *Hough–Transformation* (HT) ist eine seit Jahrzehnten bekannte, sehr leistungsstarke Methode zur Parameterextraktion.

Sie wird bei vielen Bildverarbeitungsproblemen [DUD72, KIM75] wegen ihrer Vorteile angewendet:

- Die Anzahl der Operationen ist proportional zur Anzahl der Punkte.
- Sie ist robust gegen Untergrund.
- Sie ist sensitiv auch auf nur teilweise vorhandene Objekte.

In der Hochenergiephysik ist sie zum ersten Mal auf digitalisierte Streamerkammerbilder [PUE88] angewendet worden. Dort wurden lokale Methoden in Regionen hoher Track–Dichte als problematisch angesehen, und man wandte sich globalen Methoden zur Spurerkennung zu. Sie wurde auch schon auf Driftkammerdaten in der Hochenergiephysik [NOG86, PER86] angewendet. Anfänge, diese Methode auch auf Jet–Kammern [KLE87] anzuwenden findet man in [SCH88] und [SCH89].

Zunächst soll die HT an einem Beispiel erklärt werden: Wie kann man Geradenstücke in der Ebene mit der HT finden?

Man hat einen Bildraum mit einem Koordinatensystem XY (Abb. 5.1). Dann hat man einen quantisierten Parameterraum. Dabei stellt sich die Frage, wie man die Quantisierung macht. Sie hängt von der Genauigkeit ab, mit der man die Geradenparameter bestimmen will. Dabei gehen die Fehlern der Punkte im Bildraum ein. Der Parameterraum kann durch ein zweidimensionales Array dargestellt werden, einen *Akkumulator*.

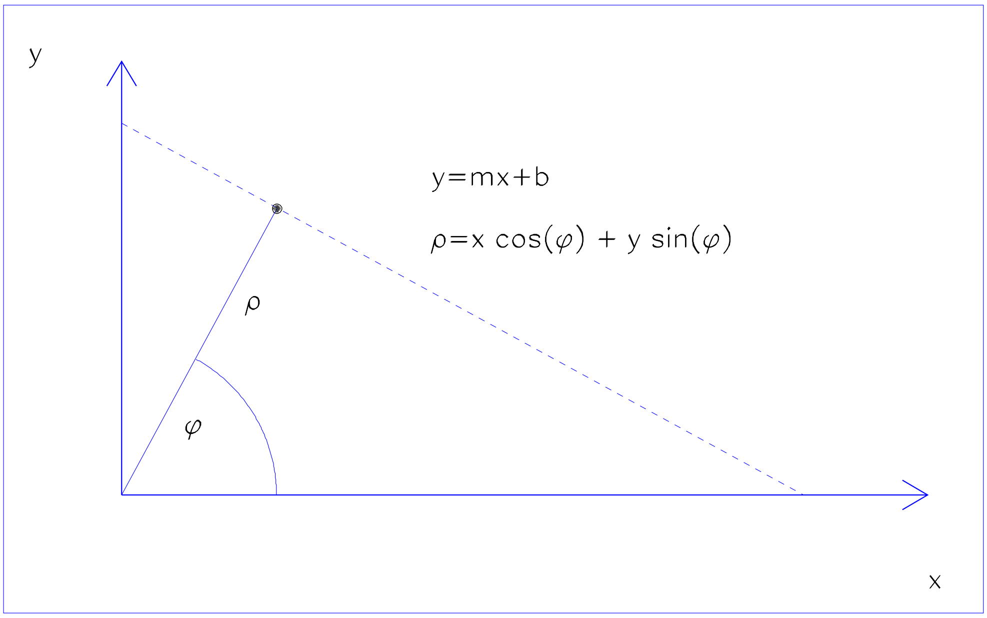

Abbildung 5.1: Die Hough–Transformation für ein Geradenstück

Im Bildraum hat man eine Punktmenge gewisser Dichte. Und man hat ein Modell: Die Punkte sollen auf einer Geraden liegen, sie sollen kollinear sein.

Um die Lagen der Geraden zu bestimmen, kann man auch verschiedene Koordinatensysteme verwenden. Oft sind kartesische Koordinaten von Vorteil, d.h. die Steigung m und der Ordinatenabschnitt c. Die Punkte (x, y), die auf der Geraden liegen, erfüllen dann die Gleichung

$$y = mx + c. \tag{5.1}$$

Unter Umständen sind aber Polarkoordinaten zu bevorzugen, d.h. den kürzesten Abstand ρ der Geraden vom Koordinatenursprung und den Winkel ϕ dieser Strecke gegen die x–Achse. Die Punkte (x, y) erfüllen dann die Gleichung

$$\rho = x\cos(\phi) + y\sin(\phi). \tag{5.2}$$

Die Abbildung des Bildraumes in den Parameterraum heißt HT. Eine solche Abbildung soll nun für das Beispiel der Geraden konstruiert werden. Man kann dazu folgendermaßen vorgehen:

Eine Möglichkeit wäre, die Geradengleichung 5.2 als Abbildungsgleichung zu benutzen. Man nimmt einen Punkt (x_1, y_1), setzt ihn in die Gleichung ein und geht gemäß der gewählten Quantisierung alle Werte

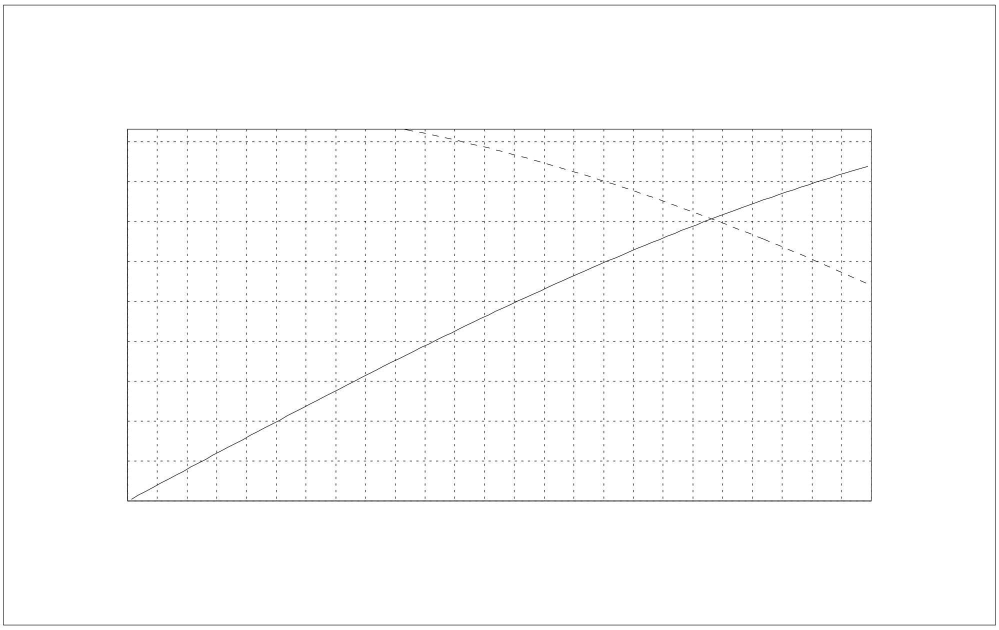

Abbildung 5.2: Parameterraum für ein Geradenstück

für ϕ durch. Man berechnet den dazugehörigen Wert ρ. Dieser fällt in eine der Gitterzellen, den *Bins*. Der Inhalt dieser Bin wird um 1 erhöht. Ein *hit* im Bildraum erzeugt also im Parameterraum als Signal eine Kurve. Diese Implementation ist ein Beispiel für eine *1-to-many Methode*.

Bildet man einen zweiten Punkt auf die gleiche Weise ab, gibt es unter Umständen einen Schnittpunkt. Diese Bin hat einen höheren Eintrag als die Bins in der Umgebung. Auf diese Weise ergibt sich also ein kleiner Peak. Dieser Peak zeigt den Parametersatz für das Geradenstück an, auf dem die beiden Punkte liegen. Im Prinzip ist das die graphische Lösung eines Gleichungssystems aus zwei Gleichungen mit zwei Unbekannten.

Führt man diese Prozedur für alle Punkte aus, so zeigt die Zelle mit dem größten Eintrag, also dem höchsten Peak, die Stelle an, wo sich die meisten Kurven geschnitten haben. Auf einer Geraden mit diesen Parametern liegen also die meisten Punkte.

Die HT transformiert also das Problem, das Geradenstück zu finden, in das Problem, das signifikante Maximum im Akkumulator-Feld zu finden. Die Hough–Transformation erfordert also eine Methode diese Peaks zu finden.

Die 1-to-many Methode hat den Vorteil, linear zu sein. Der Inhalt des Akkumulators, das *Hough–Feld,* ergibt sich als Superposition der Kurvensignale. Jeder Bildpunkt wird in der gleichen Art und Weise behandelt.

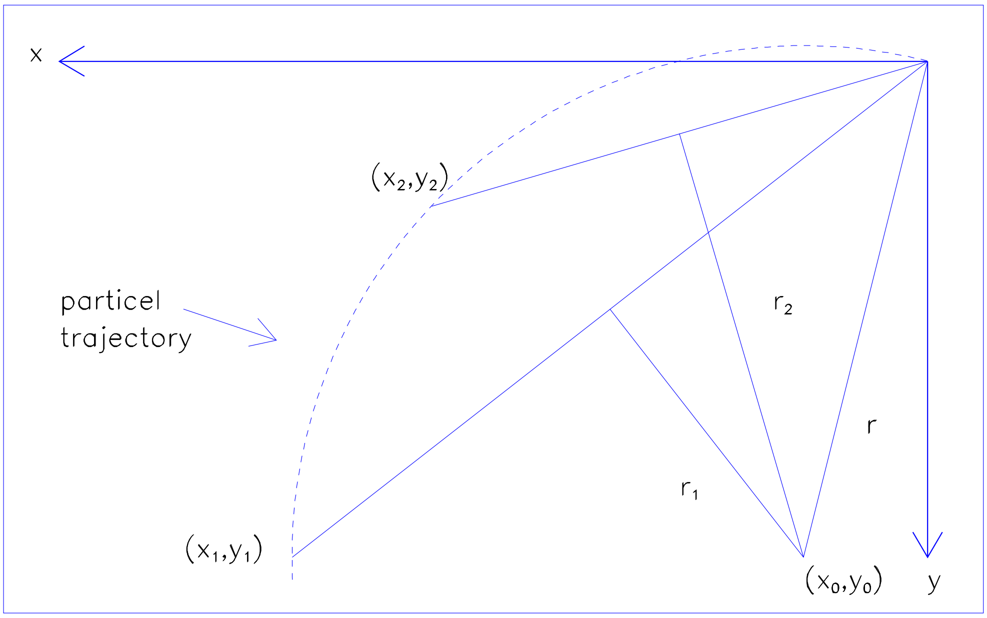

Abbildung 5.3: Die Hough–Transformation, um Kreise zu finden

Diesbezüglich ist diese Art der HT leichter zu verstehen und übersichtlicher. Der Rechenaufwand steht in direkter Beziehung zur Feinheit der Quantisierung des Parameterraumes. Das ist ein Nachteil bei großen Akkumulatorfeldern. Nachteilig ist auch der hohe Background.

Dieses Verfahren kann man leicht verallgemeinern: Sei irgendeine parametrisierbare Kurve gegeben. Hier ein allgemeines zweidimensionales Beispiel mit Parametern a und b:

$$f(x, y, a, b) = 0. \tag{5.3}$$

Hat man zwei feste Punkte, $p_1 = (x_1, y_1)$ und $p_2 = (x_2, y_2)$, und setzt diese ein, so erhält man zwei verschiedene implizit gegebene Kurven im Parameterraum.

$$f(x_1, y_1, a, b) = 0. \tag{5.4}$$

$$f(x_2, y_2, a, b) = 0. \tag{5.5}$$

a hängt von b funktional ab. Liegen die Punkte p_1 und p_2 auf der Kurve f, so haben diese beiden Kurven einen Schnittpunkt.

Man kann nun jeden Punkt gleichartig behandeln, also unabhängig von anderen Punkten und Kurvensignale im Akkumulator zulassen (*1–to–many Methode*), oder diesen Schnittpunkt ausrechnen und nur diesen Schnittpunkt abspeichern (*many–to–1 Methode*).

Im Falle eines Kreises sieht das so aus:

$$f(x,y,a,b) = (x-a)^2 + (y-b)^2 - r^2 = 0. \tag{5.6}$$

Nimmt man an, der Kreis geht durch den Ursprung, dann vereinfacht sich die Gleichung etwas, denn dann gilt

$$a^2 + b^2 = r^2. \tag{5.7}$$

Also:

$$f(x,y,a,b) = x^2 - 2ax + y^2 - 2by. \tag{5.8}$$

Man setzt p_1 und p_2 ein und erhält so zwei Geraden im Parameterraum:

$$\begin{array}{lcl} p_1 : b & = & q_1 a + s_1 \\ p_2 : b & = & q_2 a + s_2 \end{array} \tag{5.9}$$

Den Schnittpunkt dieser beiden Geraden ausrechnen heißt, dieses Gleichungssystem lösen. Man erhält so die Parameter des Kreises, auf dem diese beiden Punkte liegen:

$$a = \frac{s_1 - s_2}{q_2 - q_1} \tag{5.10}$$

$$b = q_1 a + s_1 \tag{5.11}$$

Geometrisch sieht das folgendermaßen aus (Abb. 5.3): Die Annahme, die unserem Ansatz zugrundeliegt ist, daß die Tracks in der XY–Projektion Kreise sind, die alle vom Target kommen. Das Target wird als primärer Wechselwirkungspunkt, als *Common Vertex,* angenommen. Dabei wird dieser Vertex gleichzeitig als Koordinatenursprung gewählt.

Die Idee ist nun, jeden Hit mit dem Vertex zu kombinieren. Jeder Hit wird mit dem Vertex durch eine Gerade verbunden. Auf dieser Geraden wird die Mittelsenkrechte errichtet. Auf diese Weise definiert jeder Hit eine Gerade.

Man legt dann ein Gitter auf die XY–Ebene. Das Gitter entspricht einem zweidimensionalen Feld, dem Akkumulator. Man kann nun wieder diese ganze Gerade in den Akkumulator eintragen. Das ist aber bei einem großen Akkumulator sehr aufwendig. In diesem Fall bietet es sich an, den Schnittpunkt mit einer anderen Geraden ausrechnen und nur diesen Schnittpunkt eintragen. Das wäre ein Beispiel für eine *many-to-1 Methode.*

Man zählt dann die Punkte in einer Gitterzelle, einer *bin*, und erhöht entsprechend den Eintrag im Akkumulator. Auf diese Weise entsteht das Histogramm.

Vorteil der many-to-1 Methode ist:

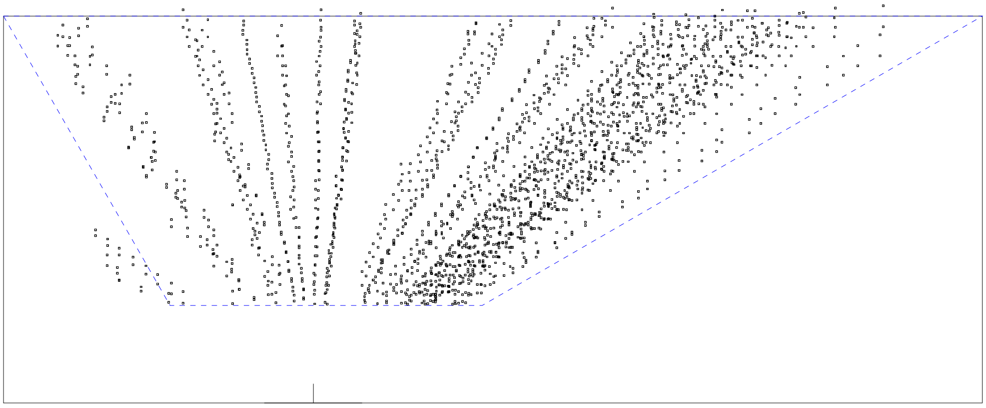

Abbildung 5.4: rz–Projektion des Ereignisses in der CDC von Abbildung 5.5. Die Auflösung des Detektors ist für die z–Koordinate um eine Größenordnung schlechter als für die x– und y–Koordinate.

- weniger Untergrund,
- der Aufwand ist unabhängig von der Akkumulatorgröße.

Nachteil ist, daß der Rechenaufwand quadratisch mit der Zahl der Punkte ansteigt.

Im idealen Fall sieht das so aus: Alle Hits, die zum gleichen Track gehören, erzeugen im Histogramm in der gleichen Bin einen Schnittpunkt. Auf diese Weise entsteht im Histogramm ein Peak. Falsche Kombinationen erscheinen nur als Untergrund im Akkumulator–Feld.

Das Problem, die Spuren in der CDC zu finden, wurde damit transformiert in das Problem, die Peaks im Histogramm zu finden.

Abbildung 5.5 zeigt ein typisches Punktemuster in der XY–Ebene (senkrecht zur Strahlachse) der CDC. Abbildung 5.4 zeigt ein typisches Punktemuster in der RZ–Ebene (parallel zur Strahlachse) der CDC.

Wendet man auf diese Punktemenge die Hough–Transformation an, so ergibt sich ein Histogramm für die Mittelpunktskoordinaten (x_c, y_c) der Kreisspuren wie es zum Beispiel in Abbildung 5.6 zu sehen ist. Die hohen Peaks korrespondieren mit den Tracks.

Als Parameter wurden die Mittelpunktskoordinaten eines Kreises durch den Koordinatenursprung gewählt. Außerdem wird aus technischen Gründen nur ein Viertel des Akkumulators gezeigt. Die rechte obere Ecke ist die Strahlrohrposition. Man sieht, daß dort der Untergrund der Hough–Transformation sehr stark ansteigt. Die strahlenförmigen Gebilde, die davon ausgehen, sind ebenfalls kombinatorischer Untergrund: Ein

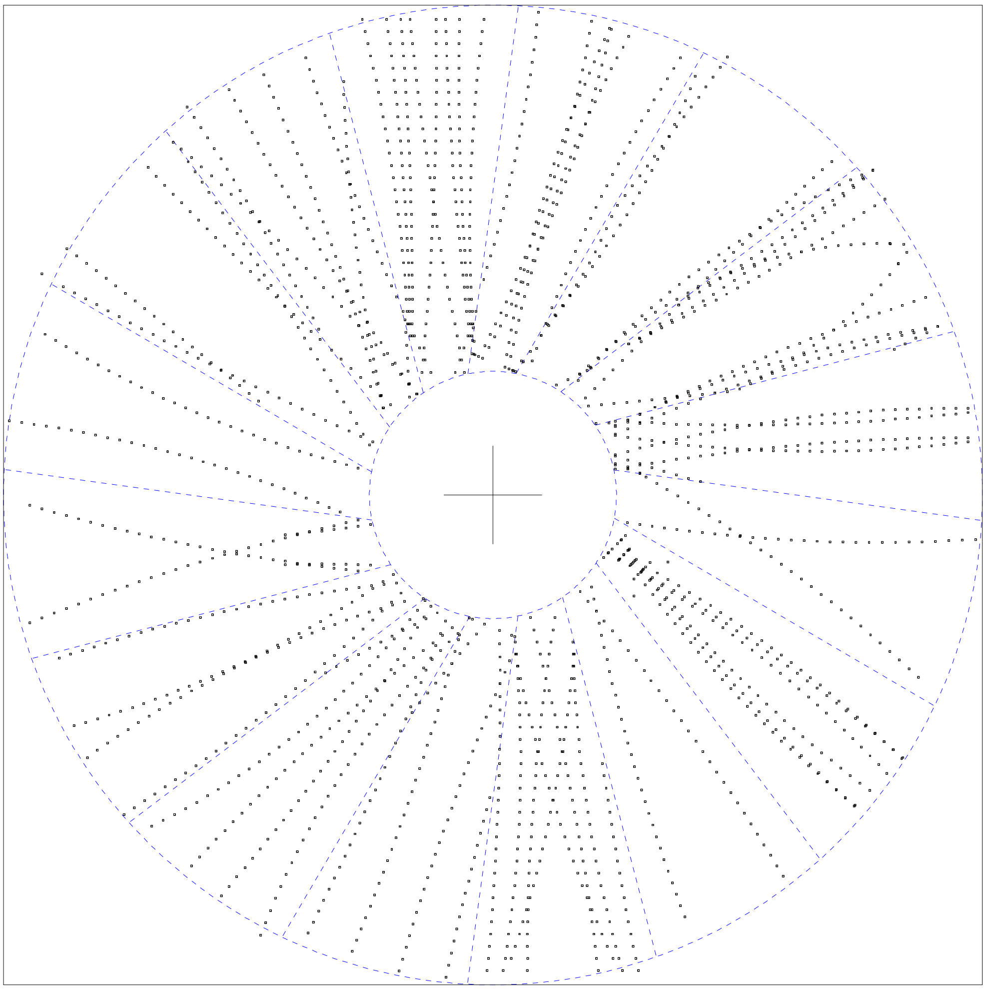

Abbildung 5.5: xy-Projektion eines Ereignisses in der CDC: Au+Au bei 1 GeV/u mit FREESCO simuliert (siehe Kapitel über das FOPI Simulationsprogramm).

falscher Hit kombiniert mit den Hits eines wahren Tracks ergibt diese korrelierten Gebilde im Akkumulator.

Ein Programm muß die richtigen Peaks, die mit den Tracks korrespondieren, durch Filtern im Histogramm extrahieren.

Die Fehler in den Positionen der Hits haben einen wesentlichen Einfluß auf die Gestalt dieser Peaks im Hough–Feld. Man kann davon ausgehen, daß die Fehler der Hits in x– und y–Richtung normalverteilt und voneinander unabhängig sind. Mit einer Fehlerfortpflanzungsrechnung für die many–to–1 Methode kann gezeigt werden [FRO88], daß die Peaks im

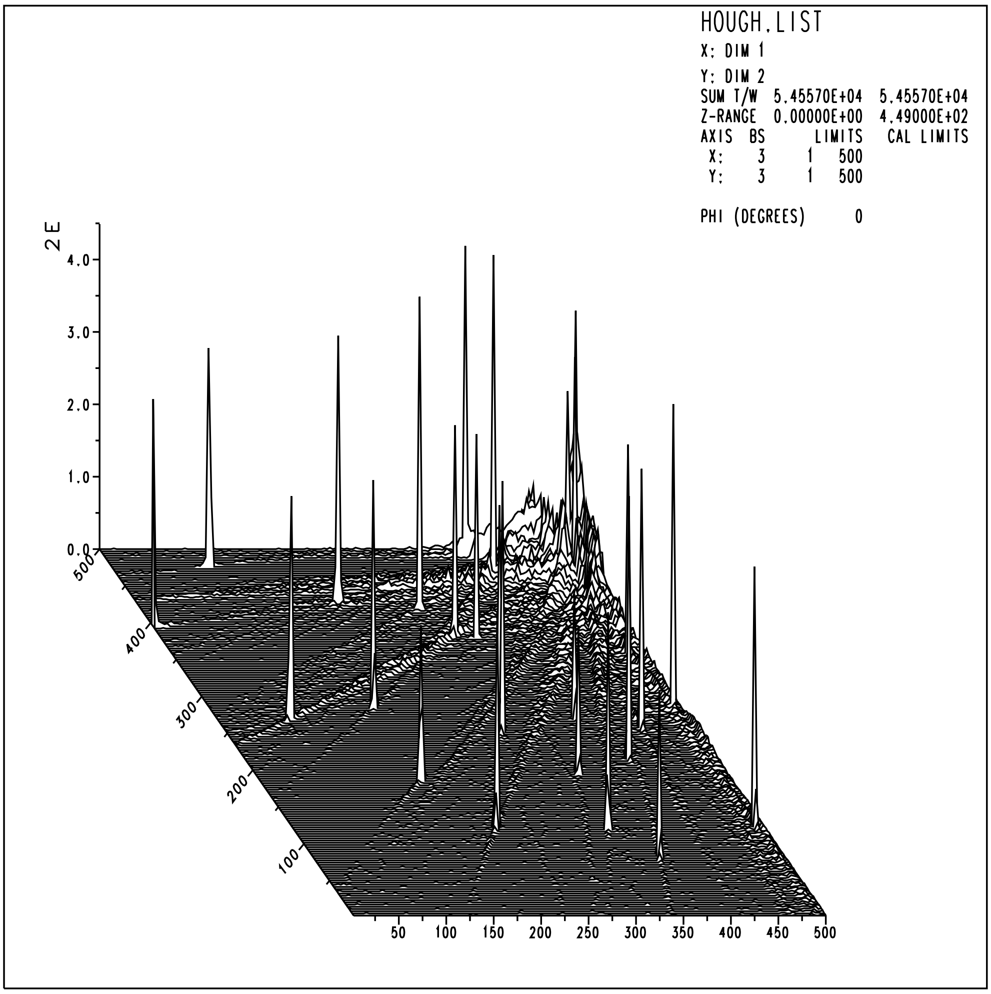

Abbildung 5.6: Akkumulator für eine Verteilung der Hits in der CDC wie es zum Beispiel in Abbildung 5.5 gezeigt wird. Die hohen Peaks im Akkumulator entsprechen den Tracks in dieser Punktmenge.

Hough–Feld dann ebenfalls gaußförmig sind, allerdings mit ellipitischen Niveaulinien. Abbildung 5.7 zeigt die Gestalt der Peaks im einzelnen.

Transformiert man das Zentrum der Gaußverteilung auf den Nullpunkt, so hat ein Peak im Hough–Feld also die allgemeine Form

$$f(x,y) = \frac{1}{2\pi\sigma_x\sigma_y\sqrt{1-\rho^2}}\exp\{-h(x,y)\}. \qquad (5.12)$$

(x,y) ist dabei eine Position im Hough–Feld, σ_x und σ_y sind die Varianzen der zweidimensionalen Gaußglocke, ρ ist der Korrelationskoeffizient.

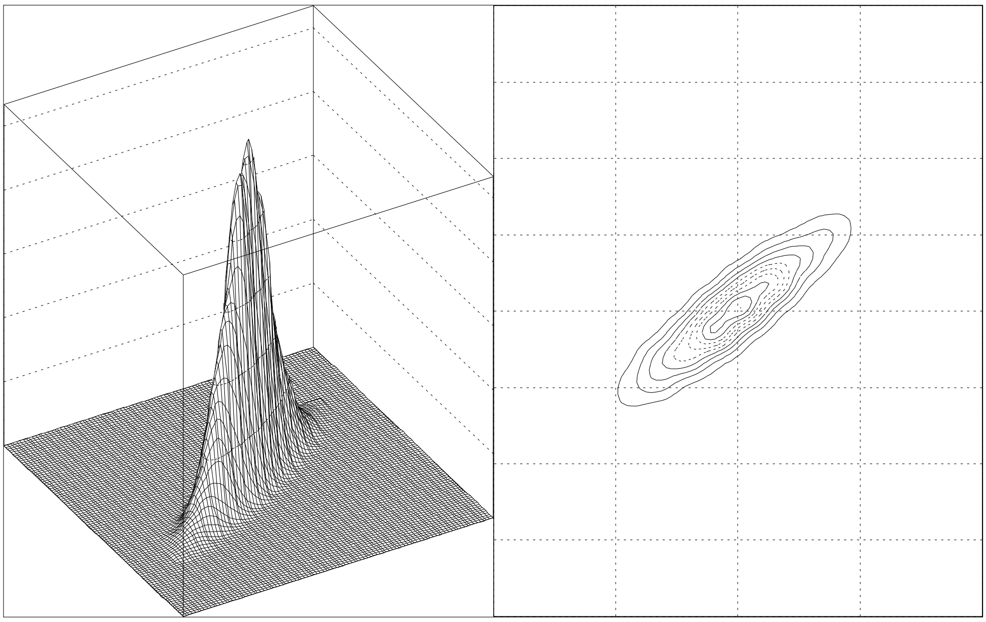

Abbildung 5.7: Verteilung der Schnittpunktefehler. Die Niveaulinien sind Ellipsen.

Bezeichnet man für Punkt i die Varianz in x–Richtung mit einem ungeraden Index (σ_{2i-1}) und die Varianz in y–Richtung mit einem geraden Index (σ_{2i}), so ist

$$\rho = \frac{\sum_{i=1}^{n} \sigma_{2i-1,2i}}{\sqrt{\sum_{i=1}^{n} \sigma_{2i-1}^2 \sum_{i=1}^{n} \sigma_{2i}^2}}. \tag{5.13}$$

Für n Punkte ergibt sich für den Exponent h(x,y) die Gestalt:

$$h(x,y) =$$

$$\frac{1}{1-\rho^2}\{\frac{x^2}{\sum_{i=1}^{n} \sigma_{2i-1}^2} - 2\rho \frac{xy}{\sqrt{\sum_{i=1}^{n} \sigma_{2i-1}^2 \sum_{i=1}^{n} \sigma_{2i}^2}} + \frac{y^2}{\sum_{i=1}^{n} \sigma_{2i}^2}\}. \tag{5.14}$$

Damit hat man die Gaußverteilung für n Hits einer Spur vorliegen. Die Höhe dieser Glockenkurve, die dem Normierungsfaktor der Verteilungsfunktion entspricht, ist ein Maß für die Güte der Schnittpunkte, d.h. falls die Hits große Fehler aufweisen, dann ist die Gaußverteilung flach. Setzt man h(x,y)=K, wobei K eine beliebige positive Konstante ist, so erhält man die Linien konstanter Wahrscheinlichkeit. Sie verlaufen ellipsenförmig. Die Halbachsen der Ellipsen sind:

$$p_1^2 = \frac{\sigma_x^2\sigma_y^2(1-\rho^2)}{\sigma_y^2\cos^2\alpha - \sigma_x\sigma_y\rho\sin 2\alpha + \sigma_x^2\sin^2\alpha}, \tag{5.15}$$

$$p_2^2 = \frac{\sigma_x^2\sigma_y^2(1-\rho^2)}{\sigma_x^2\cos^2\alpha - \sigma_x\sigma_y\rho\sin 2\alpha + \sigma_y^2\sin^2\alpha}. \tag{5.16}$$

Dabei ist p_1 die längere und p_2 die kürzere Halbachse. α ist der Winkel von p_1 gegen die x–Achse:

$$\alpha = \frac{1}{2} arctan(\frac{2\rho\sigma_x\sigma_y}{\sigma_x^2+\sigma_y^2}). \tag{5.17}$$

Kapitel 6

Die adaptive Hough–Transformation

Iteration ist menschlich, Rekursion ist göttlich.
– ANON

6.1 Die Idee der adaptiven HT

Die Frage der Quantisierung des Parameterraums tritt bei Daten mit Fehlern in den Mittelpunkt. Die Quantisierung des Parameterraumes wählt man gemäß der Auflösung für die Parameter, die erreicht werden soll. Andererseits spielen die Fehler der Hits eine entscheidende Rolle. Der Einfluß dieser Fehler kann durch eine Fehlerfortpflanzungsrechnung bei der HT bestimmt werden.

Wenn man die Größe einer Gitterzelle aber nach diesen Kriterien wählt, führt das im allgemeinen zu verhältnismäßig großen Bins. Die Auflösung reicht dann unter Umständen nicht aus, um zwei Peaks, respektive Tracks, die entsprechend dicht beieinanderliegen, zu trennen.

Außerdem funktioniert die Zuordnung 1 Bin – 1 Track nicht mehr. Man muß also wieder zu einem *Backmapping* greifen, einer inversen Abbildung vom Parameterraum in die ursprüngliche Punktmenge. Dazu muß also die Auflösung wesentlich gesteigert werden. Die Idee war daher: Die Daten sollten das Binning selbst regeln. Dies führte zur *adaptiven Hough–Transformation* (AHT) [ILL87].

In der gewöhnlichen HT wird das Problem, die Peaks im Hough–Feld zu finden, durch *exaustive search* in einem genügend fein quantisierten Parameterraum gelöst.

Ein großer Akkumulator ist aber aus mehreren Gründen nicht

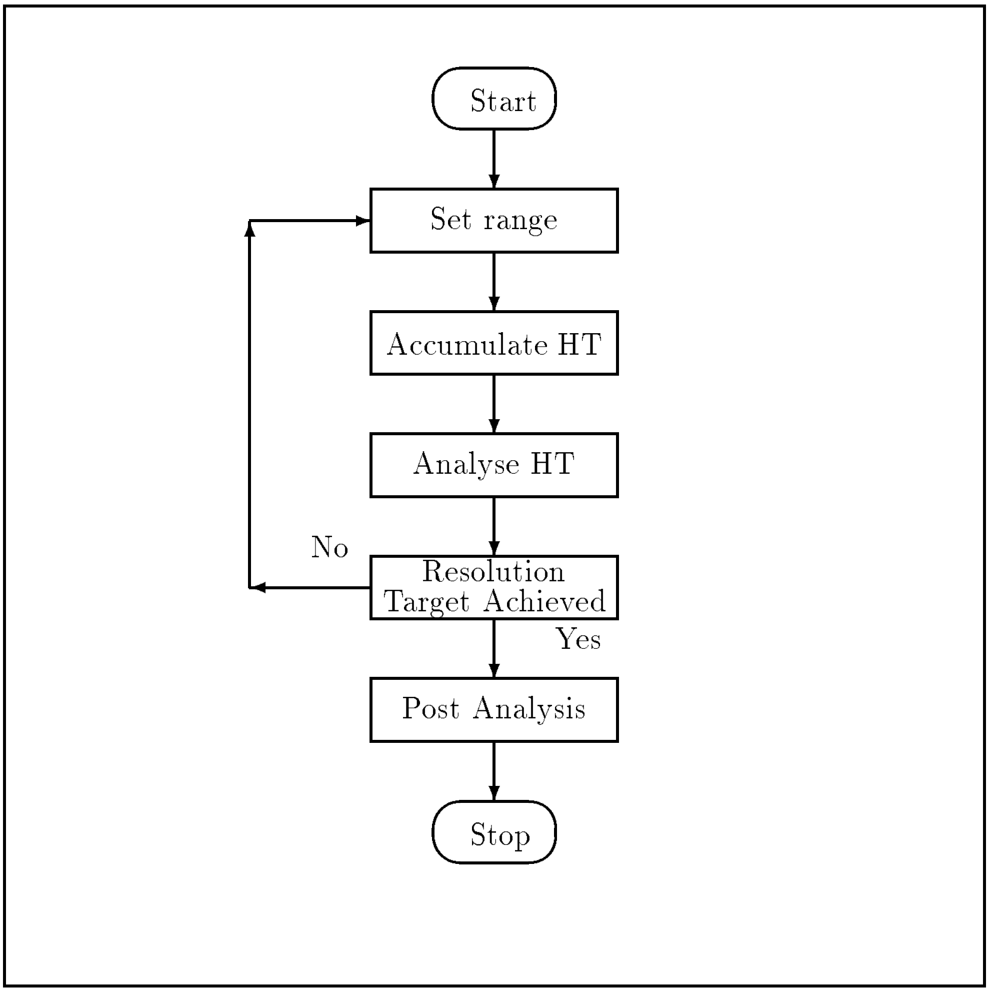

Abbildung 6.1: Flußdiagramm für die AHT

wünschenswert:

- Er nimmt einen großen Speicherplatz ein. Wenn k Parameter bestimmt werden müssen und der Parameterbereich in α Interalle zerfällt, dann hat man α^k Elemente im Akkumulator.
- Viele Zellen müssen getestet werden.
- Man produziert hauptsächlich Untergrund.

Interessant sind aber nur die Zellen in der unmittelbaren Umgebung der Peaks. Die AHT ist eine Methode, die Peaks zu lokalisieren, ohne den Untergrund im Detail auszurechnen. Sie wendet eine intelligente Histogramm– und Such–Strategie an. Diese Methode benutzt einen kleinen Akkumulator und zerlegt den Parameterbereich, der momentan interessiert, in diese wenigen Intervalle.

6.2 Flußdiagramm der AHT

Es soll nun die AHT für einen kleinen Parameterraum beschrieben werden (Abb. 6.1).

Zuerst wird die HT ausgeführt. Die Information im Akkumulator wird dann benutzt, um den Parameterbereich umzudefinieren, so daß interessante Bereiche mit größerer Auflösung untersucht werden können. Deshalb heißt die Methode auch adaptiv, weil sie sich auf die Information, die im Akkumulator enthalten ist, bezieht. Diese Iteration dauert solange, bis die Parameter mit der gewünschten Auflösung bestimmt sind.

Die bestimmten Parameter werden dann dazu benutzt, um mit einem Backmapping die Punkte zu bestimmen, die zu dem Objekt gehören. Diese Punkte werden aus der Datenbank, auf der das Programm arbeitet, herausgenommen, so daß der Zyklus für den nächsten Peak mit weniger Punkten arbeiten kann.

6.3 Das Binning

Eine der ersten Entscheidungen, die man bei der AHT treffen muß, ist die Größe des Akkumulators. Effizienz–Argumente würden zu einem Akkumulator führen, der so klein wie möglich ist: 1×2 oder 2×2. Um aber den Parameterraum interpretieren zu können, braucht man mehr Struktur. Aus diesem Grunde wurde ein Akkumulator mit 9×9 Zellen gewählt.

Um die Hough–Transformation in der *1–to–many Methode* auszuführen, ist es notwendig, die Parameterzellen zu identifizieren, die von der Kurve, die jedem Punkt zugeordnet ist, geschnitten werden. Im zweidimensionalen Fall ist das die Abbildung eines Punktes in eine Gerade im Parameterraum. Die Schnittpunkte dieser Linie mit dem Gitter können leicht ausgerechnet werden.

Für eine Parameterachse, die in β Intervalle unterteilt ist, gibt das $\beta + 1$ Schnittpunkte (Abb. 6.2). Die Zellen auf jeder Seite des Schnittpunktes liegen auf der Linie, und es kann ein Eintrag in einem logischen Feld der gleichen Größe wie der Akkumulator gesetzt werden, um das

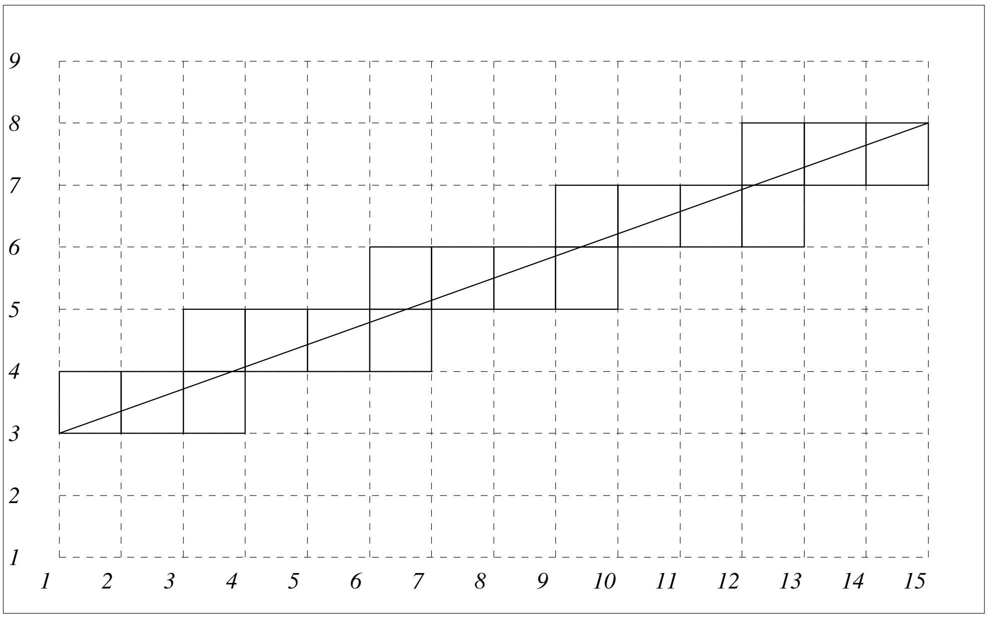

Abbildung 6.2: Binning für die AHT

anzuzeigen. Die Prozedur wird entlang der zweiten Parameterachse wiederholt. Auf diese Weise werden alle relevanten Zellen gekennzeichnet. Der Akkumulator wird dann inkrementiert, je nachdem, ob das logische Feld an dieser Stelle einen Eintrag hat. Die Akkumulation des logischen Feldes, das *Binning* also, erfordert die Berechnung von $2 \cdot (\beta + 1)$ Gitterschnittpunken für jeden Bildpunkt.

6.4 Das Clusterfinding mit dem Connected Components Algorithmus

Wenn der Parameterraum auf diese Weise populiert worden ist, muß er analysiert werden. Die signifikanten Maxima werden nach einem Verfahren bestimmt, das den Ideen von [LUM83] beziehungsweise [LSZ83] nachgebildet wurde:

Der *Connected Components Algorithmus* (CCA) ist eine Technik, binäre Bilder in *Regionen* aufzuteilen. Das resultierende Bild kann dann entsprechend den Wünschen des Benutzers weiter verarbeitet werden

Der CCA ist ein Operator, der ein binäres Eingabebild B in ein symbolisches Feld S transformiert. Das symbolische Bild besitzt dabei folgende Eigenschaften:

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
0	0	1	1	1	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0
0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0	1	0	0
0	0	0	0	1	1	1	0	0	0	1	1	1	1	1	0	0	1	0	0
0	0	0	0	0	1	1	1	0	0	1	1	1	1	0	0	1	1	0	0
0	0	0	0	0	0	1	1	0	0	0	1	1	0	0	0	1	1	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1
0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	1	1	1	1
0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	1	0	0
0	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0

Abbildung 6.3: Binäres Eingabebild für den Algorithmus, der die zusammenhängenden Regionen von 1–Werten identifizieren soll.

- Alle Punkte in B, die den Wert 0 haben, bleiben 0 in S.
- Jede *maximal zusammenhängende Teilmenge* von 1–Werten in B, ein sogenannter *Cluster*, wird mit einem *Label* versehen, einer bestimmten positiven ganzen Zahl.

Der CCA wird mit den Abbildungen 6.3 und 6.4 illustriert: Aus einem binären Eingabebild (Abbildung 6.3) wird ein symbolisches Ausgabebild (Abbildung 6.4) produziert.

Die Idee des CCA kann auf nichtbinäre Bilder verallgemeinert werden, falls es eine Eigenschaft gibt, aufgrund der jedem Bildpunkt eindeutig ein binärer Wert zugeordnet werden kann.

Sei beispielsweise ein Eingabebild, dessen Bildwerte aus der Menge $\{1, 2, 3, 4, 5\}$ kommen, gegeben. Mit dem CCA ist man dann in der Lage, die maximal zusammenhängenden Menge von Bildpunkten, deren

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0
0	0	1	1	1	0	0	0	0	0	0	0	2	2	0	0	0	0	0	0
0	0	0	1	1	1	1	0	0	0	0	2	2	2	2	0	0	3	0	0
0	0	0	0	1	1	1	0	0	0	2	2	2	2	2	0	0	3	0	0
0	0	0	0	0	1	1	1	0	0	2	2	2	2	0	0	3	3	0	0
0	0	0	0	0	0	1	1	0	0	0	2	2	0	0	0	3	3	3	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	3	3	3	3
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	3	3	3	3
0	0	4	0	0	4	0	0	0	0	0	0	0	0	0	0	3	3	3	3
0	4	4	4	4	4	4	4	4	0	0	0	0	0	0	0	0	3	0	0
0	4	4	4	4	4	4	4	0	0	0	0	0	0	0	0	0	0	0	0
0	4	4	4	4	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	4	0	0	0	0	0	0	0	0	5	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	5	5	5	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	5	5	5	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	5	5	5	5	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	5	5	5	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	5	5	0	0	0	0	0	0

Abbildung 6.4: Dieses Bild erhält man als Ergebnis des *Connected Components Algorithmus*, wenn er auf das Eingabebild in Abbildung 6.3 angewendet wird.

Wert {4} ist, zu finden. Der CCA betrachtet das Eingabebild als binär, wobei der Wert {4} der 1 entspricht und die Werte {1, 2, 3, 5} der 0. Mit dem CCA erhält man also segmentierte Bilder. Auf jedem Teil kann dann separat weiter analysiert werden.

Das Ziel des CCA ist es, jeder maximal zusammenhängenden Gruppe von 1–Werten in einem binären Bild einen und genau einen Label zuzuordnen. Dies erreicht man dadurch, indem man jede Zeile in dem binären Bild von links nach rechts und von oben nach unten durchläuft. Die Einzelheiten werden an einem Ausschnitt aus einem binären Bild klar:

```
line r-1 x x B x x
line r     x C A x x
```

A, B, C sind die binären Werte, die untersucht werden, x steht für binäre Werte, die nicht betrachtet werden. Jeder Bildpunkt mit dem Wert 0 bleibt 0. Angenommen, Bildpunkt A in Zeile r ist 1. Man will dann diesem Bildpunkt einen Label zuordnen derart, daß seine Nähe zu B und C, die voher mit einem Label versehen worden sind, berücksichtigt wird.

Es gibt mehrere mögliche Labels für A, je nachdem, welche Labels B und C schon haben. Haben B und C den Label 0, dann erhält A den nächsten freien Label. Hat C den Wert 0 und besitzt B schon einen Label, dann wird A der Label von B zugewiesen. Analog geht man vor, wenn B den Wert 0 hat und C schon ein Label zugeordnet wurde: A erhält den Label von C. Wenn beide, B und C, einen Label ungleich 0 besitzen, dann gibt es für A zwei Möglichkeiten:

- Besitzen B und C den gleichen Label, dann erhält A auch diesen Label.
- B und C weisen verschiedene Labels ungleich 0 auf. In diesem Fall verbindet Bildpunkt A zwei Cluster, die vorher als zwei verschiedene Regionen angesehen wurden.

Der zweite Fall wird in dem folgenden Beispiel illustriert:

```
          Original Picture

      line          pixel value

       1             0 0 1 0 0 0 1
       2             0 0 1 1 0 0 1
       3             0 0 1 1 1 0 1
       4             0 0 1 1 1 1 1


           Processed Image

      line          label value

       1             0 0 1 0 0 0 2
       2             0 0 1 1 0 0 2
       3             0 0 1 1 1 0 2
       4             0 0 1 1 1 1 A
```

Bildpunkt A verbindet die Regionen 1 und 2 und bekommt den kleineren der beiden Labels zugeordnet. Dazu wird noch die Äquivalenz von

Label 1 und 2 abgespeichert, so daß alle Bildpunkte mit dem Label 2 in diesem Bild eventuell zu Label 1 geändert werden können. Tabelle 6.1 zeigt noch einmal im Überblick die Labelzuordnung falls $A = 1$.

B	C	welcher Label
0	0	nächster Label
0	1	Label von C
1	0	Label von B
1	1	kleinerer Label

Tabelle 6.1: Labelzuordnung

Diese Methode ist sehr geradlinig mit Ausnahme des letzten Schrittes, wenn sich herausstellt, daß zwei Label äquivalent sind. Der Zeitaufwand der Prozedur hängt empfindlich von der Art und Weise ab, wie diese äquivalenten Label aufgelöst werden.

Die Standardmethode ist, zwei Vektoren einzuführen, wovon einer den Label der aktuellen Region enthält, der andere den zugehörigen kleinsten äquivalenten Label. Das Bild wird von oben nach unten durchlaufen, und alle äquivalenten Label werden in den beiden Vektoren gespeichert. Dann braucht man einen zweiten Durchgang, um jeden Label durch seinen kleinsten äquivalenten Label zu ersetzen.

Diese Standardmethode arbeitet zuverlässig, wird aber für große Bilder uneffektiv. Mit wachsender Bildgröße kann die Zahl verschiedener Regionen sehr groß werden. Für ein Bild der Größe 2048×2500 Bildpunkte beträgt sie bereits 80000.

Da der Akkumulator in der hier diskutierten Anwendung klein ist und diese Prozedur nur ein Mal pro Iteration durchgeführt werden muß, ist der Aufwand gering und damit akzeptabel.

Der Algorithmus wird nun als eine präzise Sequenz von Operationen beschrieben. Um dies in konsistenter Weise ausführen zu können, wird die Notation von Tabelle 6.2 eingeführt.

Der Algorithmus sieht im einzelnen folgendermaßen aus:

Name	Erklärung
NLines	number of rows in the image
L	row index
NCOLUMN	number of columns
C	column index
I(L,P)	input value (0 or 1) at row L, column P
LABEL(L,P)	output value (integer label) at row L, column P
EQTABLE	equivalence table
A	the set of pixels on the current or previous line and adjacent to the pixel (L,P) that have input value 1 and are already labeled
M	the label initially assigned to pixel (N,P)
CREATE	a function that returns an empty equivalence table
ADJACENT	a function that determines if two pixels are adjacent
ISEMPTY	a function that determines if a set is empty
NEWLABEL	a function that returns the next chronological integer label that has not been used
ADD	a function that adds a pair to an equivalence table
RESOLVE	a procedure that takes in an equivalence table and constructs a partition of the labels in that table into their equivalence classes
EQUIVCLASS	a function that returns the equivalence class of a given label
MINLAB	a function that returns the minimum label in a given equivalence class, which is used to represent the class
LA	the set of labels of the current or previous line and adjacent to pixel (L,P) that are not 0

Tabelle 6.2: Variablendeklaration für den CCA

Pass 1
"create a single empty equivalence table"
EQTABLE := CREATE();
"go trought each line L of the image"

for L:=1 **to** NLINES **do**
begin
"set all labels initially to 0"
for C:=1 **to** NColumn **do**
begin

```
        LABEL(L,C) := 0;
        "examine each pixel C on line L"
        for C:=1 to NColumn do
            begin
            "disregard 0 labels"
            if I(L,P) ≠ 1 then NEXT endif;
            "find the set of adjacent labeled pixels"
            A := {(N,Q)|(I(N,Q) = 1)
            and (LABEL(N,Q) ≠ 0)
            and ((N = L)or(N = L − 1))
            and ADJACENT((N,Q),(L,P))};
            "choose the minimum label possible for the new pixel"
            if ISEMPTY(A)
            then M := NEWLABEL()
            else M := min{LABEL(N,Q)|(N,Q) ∈ A} endif;
            LABEL(N,P):= M;
            "update equivalence table"
            for X ∈ {LABEL(N,Q)|(N,Q) ∈ AandX = M}
                call ADD((X,M)EQTABLE)
        end
    end
```

Pass 2

"Find equivalence classes. Each nonzero label is replaced with its smallest equivalent label."

```
call  RESOLVE(EQTABLE)
"relable pixels based on equivalence classes"

for  L:=1 to NLINES do
     for C:=1 to NColumn do
         begin
         if I(L,P) = 1
         then LABEL(L,P):=MINLAB(EQUIVCLASS(LABEL(L,P)))
         endif;
     end
```

In den Abbildungen 6.5 und 6.6 wird eine typische Programmausgabe gezeigt.

0	0	0	0	0	39	19	8	0	0	0	0	0	0	0
0	0	0	0	0	451	34	14	0	0	0	0	0	0	0
0	0	0	0	0	28	49	20	0	0	0	0	0	0	0
0	0	0	0	0	3	84	44	5	0	0	0	0	0	0
0	0	0	0	1	0	110	57	4	0	0	0	0	0	0
0	0	0	0	0	3	93	143	11	0	0	0	0	0	0
0	0	0	0	0	1	27	713	27	0	0	0	0	0	0
0	0	0	0	0	0	4	590	40	0	0	0	0	0	0
0	0	0	0	0	0	0	785	223	0	0	0	0	0	0
0	0	0	0	0	0	0	398	603	0	0	0	0	0	0
0	0	0	0	0	0	0	0	65	0	0	0	0	0	0
0	0	0	0	0	0	0	5	990	2	0	0	0	0	0
0	0	0	0	0	0	0	1	723	14	0	0	0	0	0
0	0	0	0	0	0	0	0	450	116	0	0	0	0	0
0	0	0	0	0	0	0	0	158	223	1	0	0	0	0
0	0	0	0	0	0	0	0	15	700	12	0	0	0	0
0	0	0	0	0	0	0	0	6	216	23	0	0	0	0
0	0	0	0	0	0	0	0	4	218	17	5	0	0	0
0	0	0	0	0	0	0	0	2	124	17	11	0	0	0
0	0	0	0	0	0	0	0	12	66	9	9	1	0	0
0	0	0	0	0	0	0	0	35	23	488	12	5	0	0

Abbildung 6.5: Typisches Aussehen des Hough–Feldes

6.5 Der neue Parameterbereich

Nachdem die Peaks in ihrer Ausdehnung mit dem CCA bestimmt wurden, wird der *most significant peak* ausgewählt: Dazu wird die Dichte jedes Peaks bestimmt, d.h.

$$\frac{\text{Anzahl der Counts}}{\text{Anzahl der Zellen}}. \tag{6.1}$$

Der Peak mit der größten Dichte wird ausgewählt. Die Ausdehnung dieses Peaks bestimmt den Parameterbereich für die nächste Iteration. Ein wichtiger Aspekt dabei ist, daß dies unabhängig in jeder Parameterdimension durchgeführt werden kann. Als Zentrum für den neuen Parameterbereich wird der Schwerpunkt des Peaks genommen. Wenn die Ausdehnung eines Parameterbereichs kleiner als 3 Zellen ist, wird er jedoch

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Abbildung 6.6: Herausgefilteter Peak von Abbildung 6.5

auf 3 Zellen gesetzt. Ansonsten hat man Probleme mit Peaks, die sich auf zwei Zellen aufteilen.

In zwei Fällen folgt die Berechnung der Parameterbereiche nicht dieser einfachen Strategie:

1. Immer wenn der Cluster kompakt in der Ausdehnung ist, aber an den Rand es Akkumulators stößt, wird angenommen, daß der Peak außerhalb des derzeitigen Parameterbereichs weitergeht. Die Parametergrenzen werden dann nur verschoben, die Größe einer Zelle bleibt unverändert.

2. Wenn ein Peak eine Linie über den ganzen Akkumulator ist und die Breite einer Zelle einnimmt, werden die Parameter nur in der dünnen Richtung geändert. Das führt zu einem gleichmäßiger populierten und daher leichter zu interpretierenden Akkumulator.

6.6 Die Komplexität der AHT

Die Vorteile der AHT zeigt eine approximative Berechnung der Komlexität der AHT im Vergleich zur gewöhnlichen HT für zweidimensionale Parameterräume. Dabei wird die Annahme gemacht, daß die Auflösung bei jeder Iteration der AHT maximal erhöht wird.

Wenn man jeden Parameter mit $\frac{1}{\alpha}$ $(\alpha > 1)$ seines vollen Bereichs auflösen will, benutzt man in der gewöhnlichen HT einen Akkumulator der Größe α^2.

In der AHT hat man einen kleinen Akkumulator der Größe β^2, $\beta < \alpha$. Der Speicherplatzvorteil ist also

$$R_s = (\frac{\alpha}{\beta})^2. \tag{6.2}$$

Das kann ein sehr großer Quotient sein.

Der Zeitvorteil hängt vom Verhältnis $\frac{1}{\gamma}$ ab, mit dem der Parameterbereich bei jeder Iteration verringert wird. Um die potentiellen Möglichkeiten der AHT zu zeigen, soll immer der gleichen Quotienten für beide Dimensionen und für alle Iterationen angenommen werden.

Die Anzahl der Iterationen n, bevor die AHT die gleiche Auflösung erreicht wie die gewöhnliche HT ist dann gegeben durch

$$(\frac{1}{\gamma})^n = \frac{1}{\alpha}. \tag{6.3}$$

Also:

$$n = \frac{log(\alpha)}{log(\beta)}. \tag{6.4}$$

Wenn die Anzahl der Bildpunkte nur groß genug ist, wird der Hauptteil der Zeit für die Akkumulation verwendet. Bei der AHT hat man bei jedem Punkt $2(\beta + 1)$ Gitterschnittpunkte zu berechnen.

Bei der gewöhnlichen HT würde man α Werte für den abhängigen zweiten Parameter zu berechnen haben, wenn man die α möglichen Werte für den unabhängigen ersten Parameter durchgeht.

Eine einzige Iteration der AHT ist also

$$\frac{\alpha}{2(\beta + 1)} \tag{6.5}$$

effizienter als die gewöhnliche HT.

Also ist die relative Effizienz, um das gleiche Objekt zu finden

$$R_c = O(\frac{\alpha}{2(\beta + 1)} \frac{log(\gamma)}{log(\alpha)}). \tag{6.6}$$

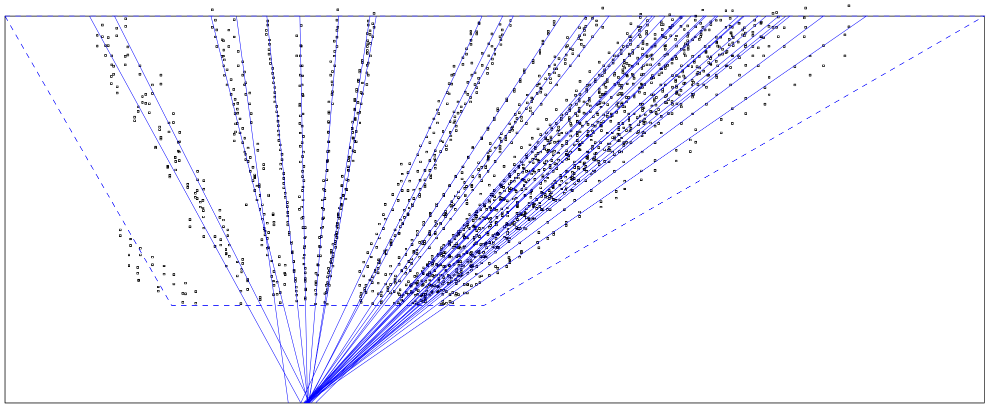

Abbildung 6.7: rz–Projektion eines Ereignisses in der CDC, die Auflösung des Detektors für die z–Koordinate ist um eine Größenordnung schlechter als in der xy–Projektion.

Ein Beispiel zur Komplexität zeigt die folgende einfache Rechnung:

$$\alpha = 1000$$
$$\beta = 9$$
$$\gamma = \frac{\beta}{3}$$
$$R_c = \frac{1000}{20}\frac{log(3)}{log(1000)} = 8.$$

Die AHT ist also potentiell um einen Faktor 8 schneller und benutzt einen Akkumulator, der etwa 10000 mal kleiner ist.

6.7 Die Strategie der Spurerkennung

6.7.1 Die Vorselektion der Daten

Zusätzlich zur xy–Projektion wird die rz–Projektion eines Ereignisses betrachtet. Dort sind die Tracks Sinuskurven, die durch Geraden, die vom Target kommen, approximiert werden können, falls der Transversalimpuls p_T größer als 200 MeV/c ist.

Da die z–Koordinate nur mit einer Auflösung bestimmt werden kann, die eine Größenordnung schlechter ist als die in der x– und y–Koordinate, erscheint dreidimensionales Tracking nicht sehr aussichtsreich. Aber die z–Information kann dazu benutzt werden, die Anzahl der Punkte, die in der xy–Ebene betrachtet werden soll, zu reduzieren.

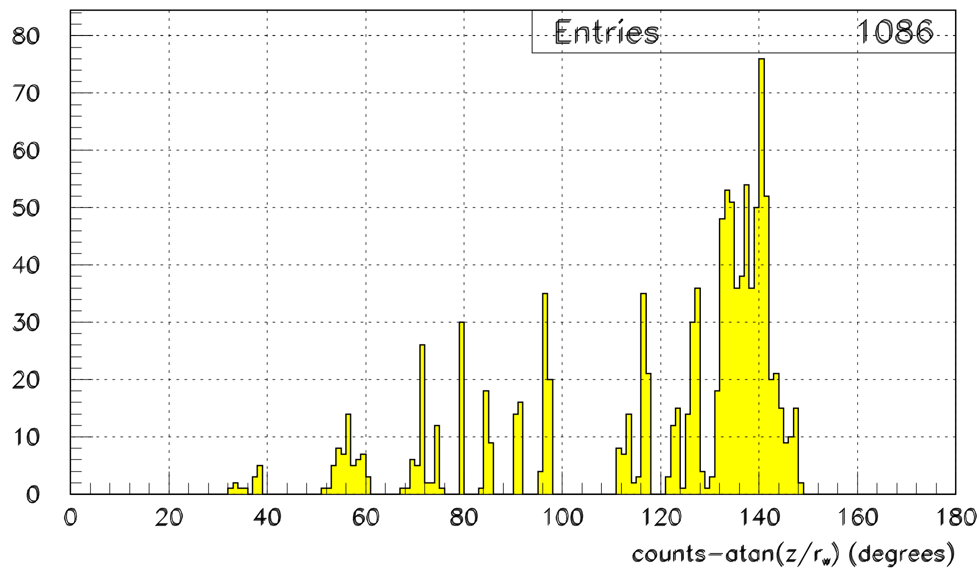

Abbildung 6.8: θ–Histogramm für das Ereignis in Abbildung 6.7. 180^o liegt in Strahlrichtung. Man sieht deutlich die höhere Trackdichte in Vorwärtsrichtung. Da maximal 30 Punkte auf einer Spur liegen, enthält ein Peak der Höhe 30 wahrscheinlich eine Spur.

Diese Vorselektion kann durch ein θ–Histogramm erreicht werden. θ ist dabei der Winkel, unter dem ein Hit in der rz–Ebene vom Koordinatenursprung aus gesehen wird, der sogenannte *Dipwinkel*. Für das Ereignis in Abbildung 6.7 erhält man das Histogramm in Abbildung 6.8. Der erste Schritt in der Spurerkennung besteht darin, mit Hilfe dieses θ–Histogramms die Punktmenge in kleinere Pakete aufzuteilen.

Auf diese Weise wird also ein Ereignis hoher Multiplizität, zum Beispiel Au+Au bei 1 GeV/u, in eines niedriger Multiplizität, vergleichbar mit Ca+Ca bei 1 GeV/u, überführt. Diese Ereignisse niedriger Multiplizität werden dann in der xy–Ebene separat behandelt. Dabei wird ein Sektor nach dem anderen abgearbeitet. Diese Prozedur ist ähnlich der Vorgehensweise des Trackingprogramms für die DIOGENE–Driftkammer am SATURNE in Saclay [POI83].

Unter den so erhaltenen Punktmengen werden mit der AHT Trackkandidaten gesucht.

Das Schema sieht nun folgendermaßen aus (Abbildung 6.9):

- Eine ZEBRA bank [GOE92] mit den zu analysierenden Daten wird eingelesen und ein θ–Histogramm angelegt.

- Mit Hilfe des θ–Histogramms wird eine Partition der Hits angelegt.
- Daraus wird die Eingangsdatenstruktur für die AHT aufgebaut.
- Die AHT wird ausgeführt.
- Die Hits, die zum gleichen Peak beitragen, werden mit einer inversen Abbildung (*Backmapping*) gesammelt. Mit einem Kreisfit werden die Trackparameter in der xy–Ebene für diese Punkte bestimmt. Zusätzlich wird ein robuster Geradenfit in der rz–Ebene auf diese Punkte angewendet. Schließlich wird ein Helixfit ausgeführt, um die endgültigen Trackparameter zu erhalten.
- Als letzter Schritt wird ein Vertexfit ausgeführt.
- Nachdem alle Cluster auf diese Weise untersucht worden sind, können präzisere (aber langsamere) Track– und Vertexfits, z. B. mit Berücksichtigung der Vielfachstreuung, durchgeführt werden.

Für die AHT ist in Abbildung 6.9 ein Flußdiagramm mit mehr Einzelheiten gezeigt.

6.7.2 Das Backmapping

Wurde ein Peak im Hough–Feld und damit ein Parametersatz für einen möglichen Track-Kandidaten identifiziert, so werden die Hits, die zu diesem Peak beigetragen haben, mit Hilfe folgender Algebra bestimmt:

Die CDC–Helix wird durch 5 Parameter bestimmt, die auch beim Track– und Vertex–Fitting benutzt werden:

$\omega = \frac{1}{R}$: Krümmung (mit Vorzeichen, positiv, falls im Uhrzeigersinn)

$\tan(\lambda)$: Tangens des Dipwinkels

ϕ_0 : Emissionswinkel, Winkel am Punkt größter Annäherung an den Koordinatenursprung in der xy–Ebene

d_0 : *Impact–Parameter*, Abstand zum Koordinatenursprung in der xy–Ebene, vorzeichenbehaftet, positiv, falls positiver Drehimpuls.

z_0 : z–Koordinate an der Stelle d_0

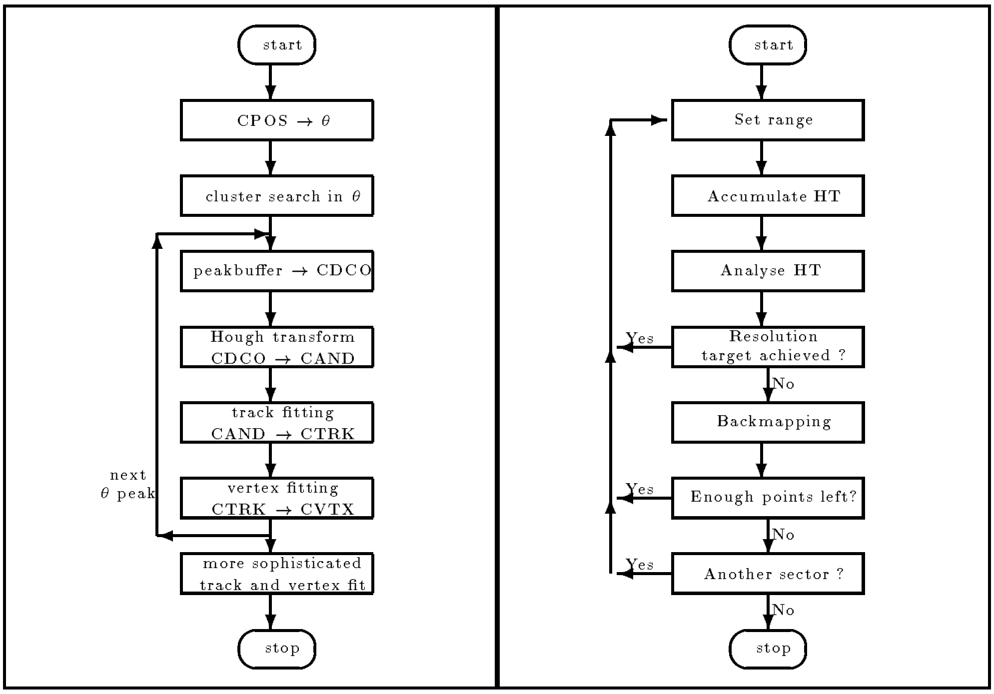

Abbildung 6.9: Flußdiagramm der Spurrekonstruktion

Die Mittelpunktskoordinaten des Kreisbogens, den man erhält, indem man die helixförmige Teilchenspur entlang ihrer Symmetrieachse (= Strahlachse) auf die Ebene senkrecht zur Strahlachse (= xy–Ebene) projiziert, sind gegeben durch:

$$x_c = (d_0 - R) \cdot \sin(\phi_0),$$
$$y_c = (R - d_0) \cdot \cos(\phi_0).$$

Die Driftebene auf einem Draht kann folgendermaßen parametrisiert werden:

x_w, y_w : Drahtkoordinaten,

s : Driftstrecke,

α : Winkel der Driftebene gegen x–Achse (konstant für einen Sektor).

Ein Punkt auf der Driftebene ist gegeben durch:

$$x \;=\; x_w + s \cdot \sin(\alpha),$$

$$y = y_w + s \cdot \cos(\alpha).$$

Der Schnittpunkt der Helix mit der Driftebene kann (in der $r\phi$–Ebene) berechnet werden nach:

$$\begin{aligned}
A &= (x_w - x_c)^2 + (x_w - x_c)^2, \\
B &= (y_w - y_c) \cdot \cos(\alpha) - (x_w - x_c) \cdot \sin(\alpha), \\
D &= B^2 - A + R^2, \\
s_{\pm} &= -B \pm \sqrt{D},
\end{aligned}$$

vorausgesetzt, daß D positiv ist. Die Lösung, die innerhalb der Sektorgrenzen liegt, wird akzeptiert (für richtige Tracks gibt es keine Zweideutigkeit).

Kapitel 7

Das FOPI Simulationsprogram

Bevor ein Experiment mit der Datenaufnahme beginnen kann, müssen das optimale Detektorlayout bestimmt und die Rekonstruktionsprogramme geschrieben und getestet werden. Diese Arbeit umfasst normalerweise mehrere Jahre. Wenn dann experimentelle Daten endlich zur Verfügung stehen, können sie ohne große Zeitverzögerung analysiert werden.

Um dies zu gewährleisten, müssen Ereignisse mit den erwarteten Rohdaten so exakt wie möglich simuliert werden.

Daher wurden mit dem Simulationsprogramm GEANT [BRU87], das seit 1974 am CERN entwickelt wird, und verschiedenen Ereignisgeneratoren Spuren erzeugt, um die Spurrekonstruktion zu testen.

Die wesentlichen Schritte in der Simulation sind:

1. Generiere ein Ereignis.

2. **(a)** Speichere die Parameter der generierten Teilchen (Tracks).

   (b) Propagiere jedes Teilchen durch den Detektor und speichere Energieverlust, TOF, Impuls und Koordinaten entlang des Tracks.

   (c) Benutze die Koordinaten auf einer Spur, um die Schnittpunkte eins jeden Teilchens mit den verschiedenen Detektorkomponenten zu bestimmen.

   (d) Berechne von den Schnittpunkten, welche Drähte in der Driftkammer angesprochen haben.

   (e) Speichere die simulierten Daten mit dem gleichen Format wie die experimentellen Daten ab.

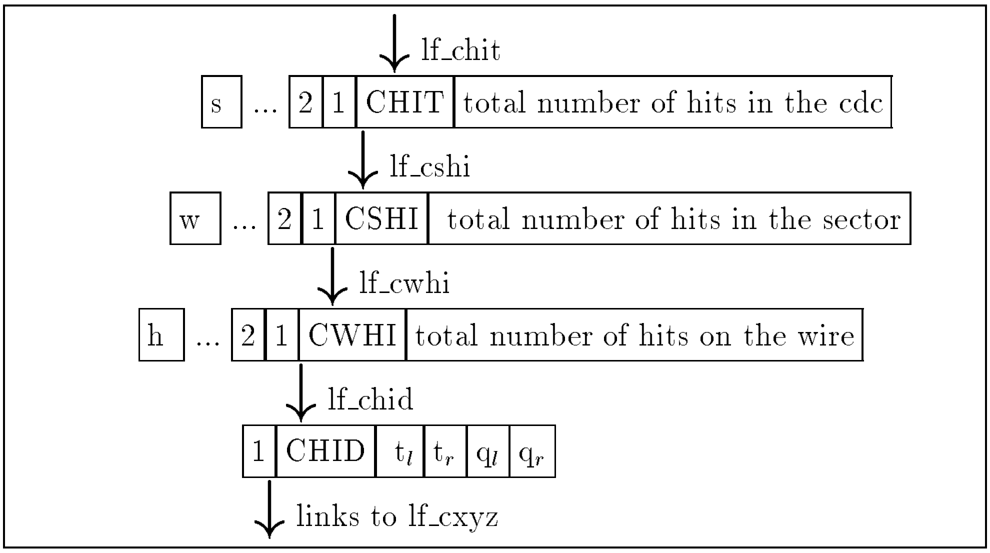

Abbildung 7.1: Struktur der Bank CHIT: t_1,t_2 sind die Ankunftszeiten des Signals an beiden Drahtenden, q_1, q_2 sind die Ladungen, die an beiden beiden Drahtenden ausgelesen werden.

---

3. Verarbeite das Ereignis mit dem Rekonstruktionsprogramm unter Berücksichtigung der Detektoreffizienz und der Ausschmierung aufgrund der Auflösung der verschiedenen Detektorkomponenten.

Die Simulation der Driftzeit wird gemäß einer linearen Driftzeit–Driftweg–Relation durchgeführt, und das endgültige Resultat wird in TDC–Einheiten angegeben. Für einen Track wird der kürzeste Abstand zu jedem Draht berechnet und in die entsprechende Driftzeit umgewandelt.

Die Information wird in ZEBRA banks [GOE92] gespeichert und auf Platte oder Tape geschrieben. Die Daten enthalten nicht nur die simulierten Rohdaten, sondern auch Informationen über die erzeugten Spuren. Diese werden benötigt, um das Rekonstruktionsprogramm aber auch das Simulationsprogramm selbst zu testen.

Der GEANT dient dazu, die von irgeneinem Ereignisgenerator mit einem bestimmten 4–Impuls erzeugten Teilchen durch das Detektorsystem zu propagieren und in regelmäßigen Abständen physikalische Informationen zu erzeugen und abzuspeichern. Dies geschieht unter Berücksichtigung der Materialeigenschaften und der damit verbundenen physikali-

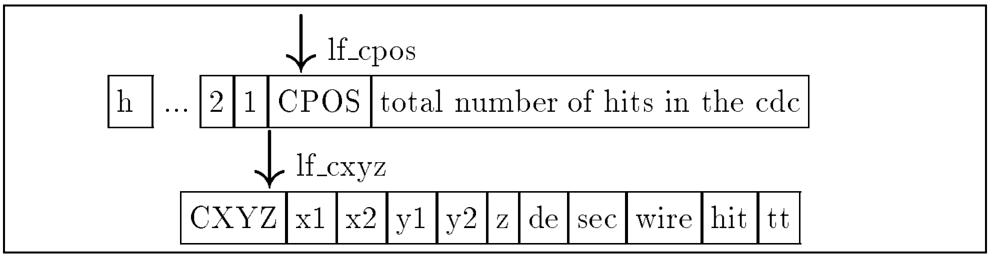

Abbildung 7.2: Struktur der Bank CPOS: x_1, x_2, y_1, y_2, sind die kartesischen Koordinaten in der xy–Ebene eines Kammerhits, die im Koordinaten–Finding aus der Driftzeit berechnet werden, z ist die kartesiche z–Koordinate wie sie über Ladungsteilung gewonnen wird. Q ist die Gesamtladung, tt die Driftzeit, gleichzeitig wird die Sektornummer und die Drahtnummer abgespeichert. Dazu kommt noch eine Zahl, die angibt, um den wievielten Hit auf dem Draht es sich handelt.

schen Prozesse, wie Energieverlust und Vielfachstreuung beispielsweise.

Diese Informationen entsprechen aber noch nicht den Daten, die man im Experiment vom Detektor erhält. Die Digitalisierung der Daten vom GEANT bleibt dem Benutzer überlassen, der für eine realistische Simulation möglicht vielen Eigenschaften des jeweils vorliegenden Detektors berücksichtigen muß. Die Routine GUDIGI, insbesondere die Routine CDCDIGI für die CDC, sorgt für die Digitalisierung der simulierten Daten, Erzeugung von Rohdaten wie im Experiment und Füllen der entsprechenden ZEBRA–Banks. Um die gleiche Analysesoftware für experimentelle und simulierte Daten anwenden zu können, wurde die gleiche Datenstruktur wie im Experiment verwendet. Hier wird das FORTRAN–Unterprogramm–Packet ZEBRA verwendet, das eine dynamische Datenverwaltung mit Hilfe von Zeigern und Listen oder Bäumen ermöglicht. Die in der Analyse zur Verfügung stehenden Daten haben damit die in Abbildung 7.1 dargestellte Struktur.

Für die Simulation von Ereignissen im 4π–Detektor stehen verschiedene Ereignisgeneratoren zur Verfügung:

- MCRANCONE; eine bestimmte Anzahl von Testteilchen kann mit einem bestimmten Impuls in einen Konus emittiert werden.
- MCFRED: Goldhaber Fragmentation
- COSMIC: Cosmic Rays
- LANDAU FIREBALL

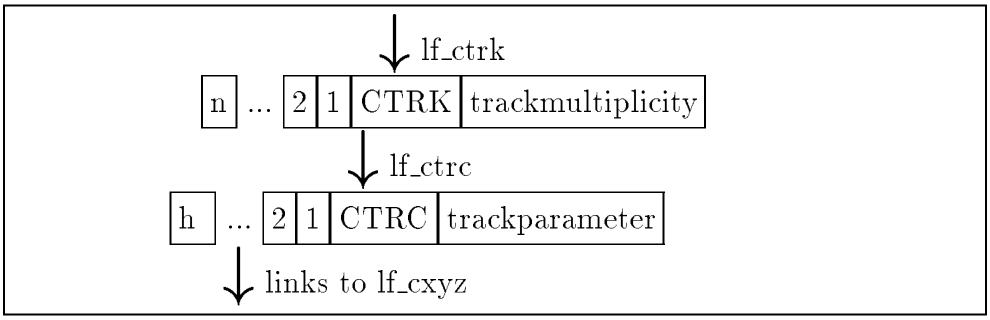

Abbildung 7.3: Struktur der Track–Bank CTRK: Einmal wird die Anzahl der Hits pro Track abgespeichert und ein Errorflag. Dann kommen die fünf Helixparameter und die dazugehörige Kovarianzmatrix, von der nur die fünfzehn oberen Elemente abgespeichert werden, da die Matrix symmetrisch ist.

- FREESCO

Für die Untersuchungen im Rahmen dieser Arbeit wurden nur MCRANCONE und FREESCO [FAI86] verwendet. Mit MCRANCONE wurden die numerischen Eigenschaften des Track– und Vertexfits einmal getestet und dann die Doppelspurauflösung der adaptiven Houghtransformation (AHT) bestimmt. Der FREESCO Ereignisgenerator kam haupsächlich für die Untersuchung der Effizienz der AHT zum Einsatz.

FREESCO (FRagmentation by Explosion/Evaporation Simulation COde) ist ein 1986 von G. Fai und J. Randrup entwickeltes Programm, das den (statistischen) Zerfall von hochenergetischer Kernmaterie behandelt, wie sie bei Schwerionenstößen mit Projektilenergien oberhalb einiger zehn MeV/u auftritt [FAI86]. Das zugrundeliegende Modell stützt sich auf den verfügbaren mikrokanonischen Phasenraum und ist anwendbar bis zu einigen GeV/u Strahlenergie, wo die Erzeugung von anderen Teilchen als Pionen wichtig wird. Ziel ist unter anderem die Beschreibung der Häufigkeit, Energieverteilung und kollektiven Eigenschaften der emittierten Fragmente. In der experimentellen Schwerionenphysik wird das Programm beim Design von Detektoren sowie zur Bestimmung des kollektiven Flusses aus den experimentellen Daten benutzt.

Da die Simulation detaillierter dynamischer Prozesse viel zu aufwendig ist, wird ein Modell benutzt, das von einfachen geometrischen Überlegungen ausgeht. Die Nukleonen von Projektil und Target werden entsprechend dem "Fireball"–Modell in 3 unabhängige Gruppen (=Quellen) eingeteilt:

1. Projektil-Zuschauer ("projectile spectators")
2. Target-Zuschauer ("target spectators")
3. "Participants"

Jede Quelle wird beschrieben durch:

- die Baryonenzahl
- die Ladung
- den 4er Impuls

Bei Quellen mit hoher Anregungsenegie wird das vollständige Zerplatzen der Quellen, bei Quellen mit niedriger Energie wird Abregung durch Abdampfen von Fragmenten simuliert. Als Grenze zwischen diesen beiden wird

$$E_{Grenze} = A \cdot \frac{\text{Bindungsenergie}}{\text{Nukleon}}$$

benutzt. Der Zerfall erfolgt in Pionen, Nukleonen, Deuteronen, α-Teilchen und schwerere Fragmente, wobei die Kerne in einen entsprechend den tabellierten Daten definierten Endzustand übergehen. Konkurrierender, bzw. anschließender elektromagnetischer Zerfall wird nicht behandelt.

Der Zerfall von Quellen hoher Anregungsenergie wird statistisch unter Berücksichtigung des mikrokanonischen Ensembles beschrieben. Dabei wird nicht von einem Gleichgewichtszustand ausgegangen.

Kapitel 8

Test mit Simulationen

Andererseits dürfen wir die Effizienz nicht einfach ignorieren.

– Jon Bentley

8.1 Effizienz und Zeitaufwand

Um die Performance eines Programms zu messen, braucht man ein quantitatives Maß. Dieses Maß wird im Bereich der Spurerkennung (Pattern Recognition) in der Physik als *Effizienz* bezeichnet. Es ist auf den ersten Blick überraschend, daß in vielen Veröffentlichungen auf diesem Gebiet die erhaltene Effizienz *nicht* diskutiert wird. Andererseits haben die Autoren aber sicher eine solche Größe, um ihre Methoden zu testen.

Ein Grund dafür ist gewiß, daß es keine universelle Definition einer Effizienz gibt und daher Befürchtungen vor ungerechtfertigen Vergleichen und Mißinterpretationen bestehen. Um dies zu illustrieren, seien m Ereignisse mit einer mittleren Anzahl von n Tracks pro Ereignis betrachtet. Es gibt nun verschiedenen Möglichkeiten, eine Effizienz zu definieren [GRO87]:

- Die mittlere Wahrscheinlichkeit, einen einzelnen Track zu finden, ist die Zahl der korrekt gefundenen Track dividiert durch die totale Anzahl der Tracks ($n \cdot m$). Diese Zahl fällt normalerweise ziemlich langsam mit größer werdendem n, wenn also die Ereignisse immer konfuser werden.

- Die Zahl der Ereignisse, bei denen *alle* Tracks korrekt gefunden

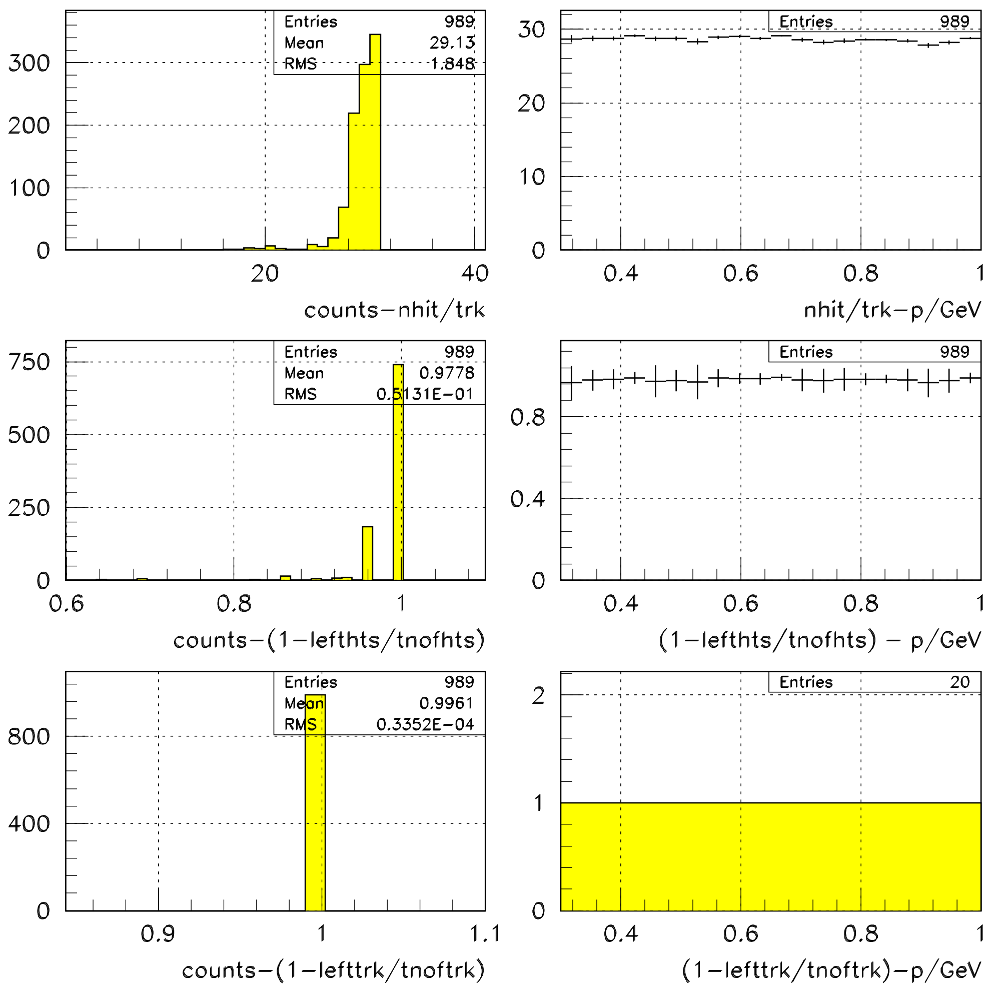

Abbildung 8.1: Effizienz der Spurrekonstruktion für eine Spur pro Ereignis: Die erste Zeile zeigt, aus wievielen Hits ein reskonstruierter Track besteht. Maximal sind 30 Hits möglich. Die Teilchenspuren werden über einen weiten Impulsbereich nahezu vollständig rekonstruiert. Zweite Zeile: Point–Removal–Effizienz und ihre Abhängigkeit vom Teilchenimpuls. Letzte Zeile: Track–Removal–Effizienz und ihre Abhängigkeit vom Teilchenimpuls.

---

wurden. Diese Größe stellt ein sehr empfindliches Maß dar und fällt normalerweise sehr schnell mit n.

- Der Anteil an Ereignissen, in dem wenigsten ein bestimmter Teil, beispielsweise 90%, der Tracks korrekt gefunden wurden.

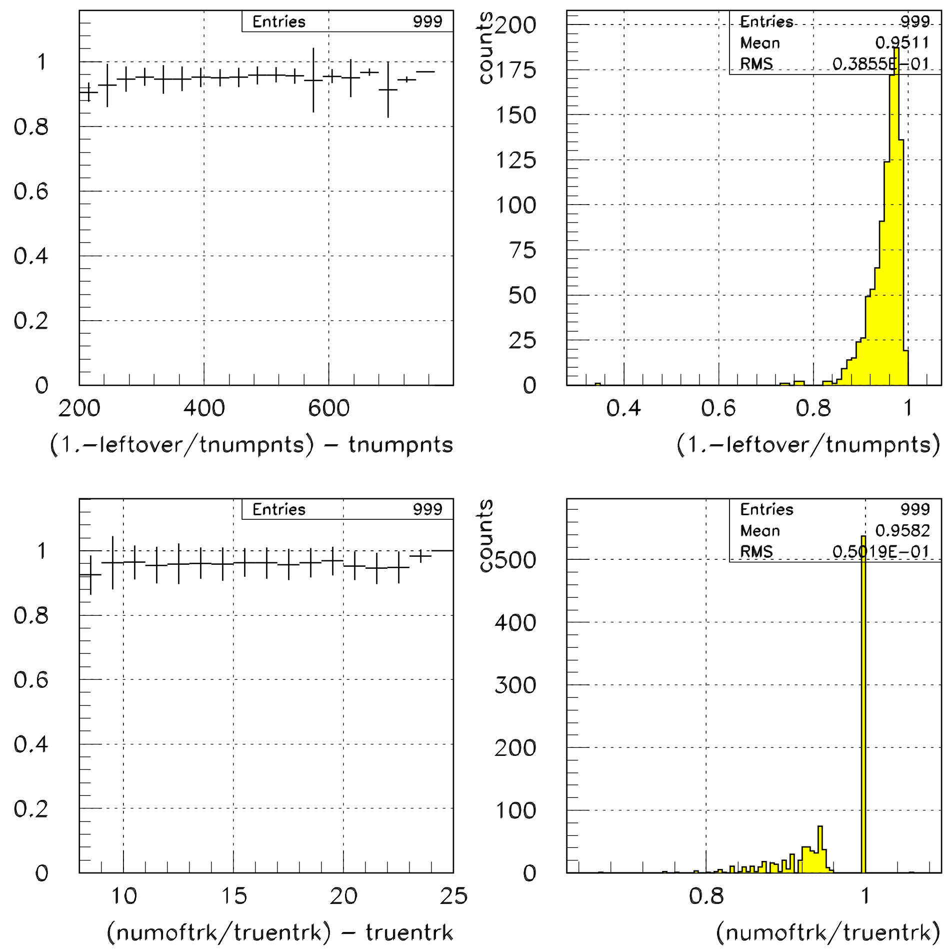

Abbildung 8.2: Effizienz der Spurrekonstruktion für etwa 1000 mit FREESCO simulierte Ca+Ca–Ereignisse mit 1 GeV/u: Erste Zeile: Point–Removal–Effizienz. Letzte Zeile: Track–Removal–Effizienz. Beide Effizienzen über 95%.

- Dann kann man jede der obigen Metriken nehmen, aber Tracks, die notorisch Schwierigkeiten bereiten, unberücksichtigt lassen, wie zum Beispiel Tracks mit einem Impuls, der kleiner als ein bestimmter Betrag ist, oder Tracks in einer besonders schwierigen Detektorregion.

Alle diese Vorschläge sind aber weit davon entfernt, ein zufriedenstellendes Maß für die Effizienz zu bieten. Um von irgendeinem Nutzen zu sein,