Ein hoch effizientes Geometriemodell für die Spurrekonstruktion im Silizium-Vertex-Detektor bei CDF 2 und Suche nach korrelierter Charm-Produktion

Zur Erlangung des akademischen Grades eines
DOKTORS DER NATURWISSENSCHAFTEN
von der Fakultät für Physik der
Universität Karlsruhe (TH)

genehmigte

DISSERTATION

von

Dipl. Phys. Kurt Rinnert
aus Ludwigshafen

Tag der mündlichen Prüfung: 04.02.2005

Referent: Prof. Dr. M. Feindt, Institut für Experimentelle Kernphysik

Korreferent: Prof. Dr. G. Quast, Institut für Experimentelle Kernphysik

Deutsche Zusammenfassung

Einführung

Im Jahr 2004 wurden die drei theoretischen Physiker David J. Gross, H. David Politzer und Frank Wilczek für ihre bahnbrechenden Arbeiten auf dem Gebiet der Quantenfeldtheorie mit dem Nobelpreis für Physik ausgezeichnet. Sie konnten zeigen, dass die Quantenchromodynamik (QCD) als nicht abelsche Eichtheorie eine Eigenschaft besitzt, die als asymptotische Freiheit bezeichnet wird. Damit war die Eignung der QCD zu Beschreibung der Starken Wechselwirkungen zwischen Quarks erwiesen. Dieses Resultat gab der Interpretation der Quarks als physikalisch reale Teilchen eine überzeugende Grundlage. Seitdem konnten viele Aspekte der Dynamik der Starken Wechselwirkung mit Hilfe der QCD berechnet werden. Allerdings entziehen sich viele Eigenschaften der Quark-Quark-Wechselwirkung bis heute einer Berechnung aus grundlegenden Prinzipien. Dies gilt insbesondere für die Fragmentierung schwerer Quarks in Hadronen und deren Produktion in Hadronenkollisionen. Gerade diese Prozesse sind aber von größter Bedeutung für die Verbindung von Theorie und Experiment, da sich freie Quarks der Beobachtung entziehen. Es ist daher wichtig, Methoden zu entwickeln, die die Eigenschaften der Produktion schwerer Quarks in Hadronenkollisionen experimentell zugänglich machen. Die vorliegende Arbeit beschäftigt sich mit der Entwicklung dieser experimentellen Methoden und der Suche nach korrelierter Produktion von Charm-Anticharm Quarkpaaren in den vom CDF 2-Experiment genommenen Daten. Zu den behandelten experimentellen Methoden gehört insbesondere eine effiziente Software zur Rekonstruktion geladener Teilchenspuren. Der Begriff Effizienz hat hier zwei Bedeutungen. Zum einen bezeichnet er ein Maß für die Fähigkeit Dinge zu finden, die tatsächlich vorhanden sind, und zum Anderen meint er Effizienz im Sinne von angemessenem Verbrauch von Rechenresourcen. Der Aufwand für die Umsetzung effizienter Rekonstruktions-Software wird bei der Auslegung eines Experiments häufig unterschätzt. Oftmals sind kleine Änderungen in der Konstruktion des Detektors, wie z.B. die Änderung des Anschlussschemas in einer Lage des Silizium-Vertex-Detektors, für erhebliche Verzögerungen auf dem Weg zu einer effizienten Rekonstruktions-Software verantwortlich. Auf der anderen Seite ist es unmöglich, alle zukünftigen Folgen eines – notwendigerweise – komplizierten Designs vorherzusehen, handle es sich um Hardware oder Software. Dies gilt insbesondere für das Spurrekonstruktionssystem bei CDF 2. Es erfordert daher viel Erfahrung und zahllose Entwicklungszyklen, mit einer Rekonstruktions-Software aufwarten zu können, die es erlaubt hunderte von Millionen von Ereignissen zu analysieren. Folglich wurden große Anstrengungen unternommen, um effiziente Spurrekonstruktions- und Spur-Anpassungs-Software zu entwickeln. Diese resultierten in einer hoch genauen und effizienten Umsetzung, die die Grundlage aller auf Spuren geladener Teilchen gegründeten Analysen in der CDF 2-Kollaboration ist. Viele der technischen Aspekte dieser Umsetzung werden in der vorliegenden Arbeit diskutiert, wobei ein besonderer Schwerpunkt auf die entscheidende Bedeutung einer akkuraten Geometriebeschreibung gelegt wird. Erst die Verfügbarkeit dieser mächtigen experimentellen Werkzeuge hat es, erstmals an einem Hadronenbeschleuniger, ermöglicht, Evidenz für die korrelierte Produktion von D^0-$\bar{D}^0$-Paaren zu etablieren.

Quellen Korrelierter Charm-Anticharm Paare

Die experimentelle Signatur für korrelierte Charm-Produktion ist das Auftreten von zwei Hadronen, von denen eines ein Charm Quark (c) und das andere dessen Antiteilchen ($\bar{c}$) enthält, in einem Kollisionsereignis. In der Praxis muss diese Definition weiter eingeschränkt werden, da nicht alle Endzustände, in die die Hadronen zerfallen können, gleichermaßen experimentell zugänglich sind. Die Untersuchungen in dieser Arbeit beschränken sich daher auf die Mesonen D^0, D^+ und D^{*+} die jeweils ein Charm-Quark enthalten. Weiterhin werden nur die Endzustände $D^0 \to K^-\pi^+$, $D^+ \to K^-\pi^+\pi^+$ und $D^{*+} \to D^0(\to K^-\pi^+)\pi_s^+$ untersucht[1]. Auf der Ebene der Quarks sind also $c\bar{c}$-Paare die Quellen solch korrelierter Paare von Mesonen. Die $c\bar{c}$-Paare können entweder direkt in der Proton-Antiproton-Kollision erzeugt werden oder in Zerfällen entstehen. Die Produktionsprozesse führender Ordnung $\mathcal{O}(\alpha_s^2)$ sind $q + \bar{q} \to Q + \bar{Q}$ (Quark-Antiquark-Vernichtung) und $g + g \to Q + \bar{Q}$ (Gluonenfusion). Die entsprechenden Feynmandiagramme zeigt Abb. 1. In Proton-Antiproton-Kollisionen bei $\sqrt{s} = 1.96$ TeV überwiegt der Beitrag der Gluonenfusion. Insbesondere in Produktion von Charm-Quarks können Prozesse der Ordnung $\mathcal{O}(\alpha_s^3)$ einen wesentlichen Beitrag zum Gesamtwirkungsquerschnitt liefern. Dies sind zum einen Prozesse in denen ein weiteres virtuelles oder reelles Gluon zu den Diagrammen in Abb. 1 hinzugefügt wird. Aufgrund der vielen Möglichkeiten ein Gluon anzuhängen gibt es eine große Zahl solcher Diagramme. All diesen Produktionsmechanismen ist gemein, dass die $c\bar{c}$-Paare mit im wesentlichen entgegengesetzt gerichteten Impulsen erzeugt werden. Dies unterscheidet sie von weiteren Prozessen höherer Ordnung. Der Wirkungsquerschnitt des Prozesses $gg \to gg$ ist um einen Faktor hundert größer als der des Prozesses $gg \to Q\bar{Q}$. dadurch ist es möglich, dass $gg \to g(\to Q\bar{Q})g$ Prozesse, die formal von der Ordnung $\mathcal{O}(\alpha_s^3)$ sind, einen ähnlich großen Beitrag liefern, wie die Gluonenfusion führender Ordnung. Zu diesen Prozessen gehören die Flavour-Anregung, bei der ein virtuelles schweres Quark in einem der kollidierenden Hadronen durch Streuung an einem Parton des anderen Hadrons auf die Massenschale gehoben wird, und Gluonen-Splitting Prozesse ($g \to Q\bar{Q}$), in denen kein schweres Quark an der Streuung beteiligt ist. Abbildung 2 zeigt die entspre-

[1] Wenn nicht ausdrücklich anders gekennzeichnet, verstehen sich hier und im folgenden die ladungskonjugierten Zustände als eingeschlossen.

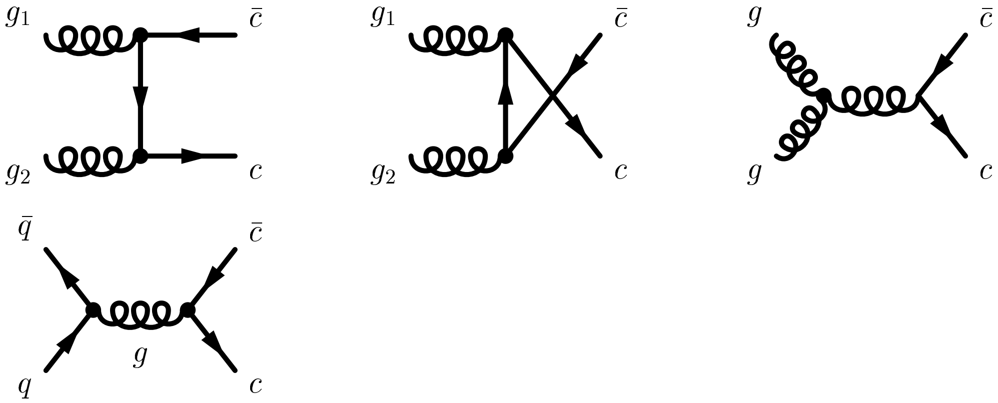


Abbildung 1: Feynmandiagramme zur $c\bar{c}$-Produktion in führender Ordnung $\mathcal{O}(\alpha_s^2)$.

chenden Feynmandiagramme. Die Zuordnung des mittleren Diagramms in Abb. 2 ist nicht ganz eindeutig. Gewöhnlich rechnet man es dem Gluonen-Splitting zu, da kein schweres Quark an der harten Streuung beteiligt ist. Die relativen Impulsrichtungen der so produzierten schweren Quarks unterscheiden von denen im Fall von Quark-Antiquark-Vernichtung und Gluonen-Fusion dadurch, dass es einen erheblichen Anteil von $c\bar{c}$-Paaren mit nahezu gleichgerichteten Impulsen gibt. Dies erlaubt es, aus der Winkelverteilung korrelierter Paare von Mesonen Rückschlüsse auf die Produktionsmechanismen zu ziehen. Zusätzlich zur direkten Produktion in Proton-Antiproton-Kollisionen können korrelierte $c\bar{c}$-Paare auch in Zerfällen schwerer Teilchen entstehen. Insbesondere wenn es sich dabei um langlebige Teilchen handelt, kann dies ebenfalls zu Paaren von Mesonen mit gleichgerichteten Impulsen führen. Diese unterscheiden sich allerdings durch ihre Zerfallslängenverteilung von Paaren, die aus der direkten Produktion stammen. Die meisten Fragen, die hier aufgeworfen werden, können in der vorliegenden Arbeit nicht behandelt werden. Die Bedeutung dieser Fragen unterstreicht aber einmal mehr die Notwendigkeit der Entwicklung experimenteller Methoden, die ihre Beantwortung erst ermöglichen.

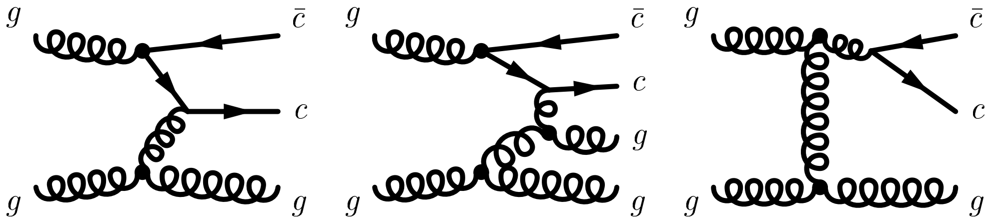


Abbildung 2: Feynmandiagramme zur $c\bar{c}$-Produktion durch Flavour-Anregung und Gluonen-Splitting.

Experimenteller Aufbau

Die in dieser Arbeit untersuchten Daten wurden vom CDF 2 Experiment am Tevatron Beschleuniger genommen. Das Tevatron ist ein symmetrischer Proton-Antiproton-Beschleuniger mit einer Schwerpunktsenergie von $\sqrt{s} = 1.96$ TeV. Es befindet sich am Fermi National Accelerator Laboratory (Fermilab) in den USA. Der Aufbau des CDF 2 Detektors weist die für Detektoren an Beschleunigern typische Zylindersymmetrie auf. Abbildung 3 zeigt einen Querschnitt durch den Detektor. Nah am Wechselwirkungspunkt befindet sich ein Silizium-Vertex-Detektor, der von einer Spurkammer umgeben ist. Dieses System zur Rekonstruktion geladener Teilchenspuren befindet sich innerhalb eines supraleitenden Solenoid-Magneten, der ein nahezu homogenes Feld der Stärke 1.4 T erzeugt. Dieses Magnetfeld krümmt die Bahnen geladener Teilchen und erlaubt so die Messung ihres Transversalimpulses. Direkt innerhalb des Magneten ist ein System zur Flugzeitmessung montiert, welches, in Kombination mit der Impulsmessung, der Teilchenidentifikation dient. Außerhalb des Magneten befinden sich hadronische und elektromagnetische Kalorimeter. Ganz außen schließlich befinden Detektoren zum Nachweis von Myonen, der einzigen Teilchen, die mit großer Wahrscheinlichkeit das gesamte Detektormaterial durchqueren. Die extrem hohe Kollisionsrate am Tevatron

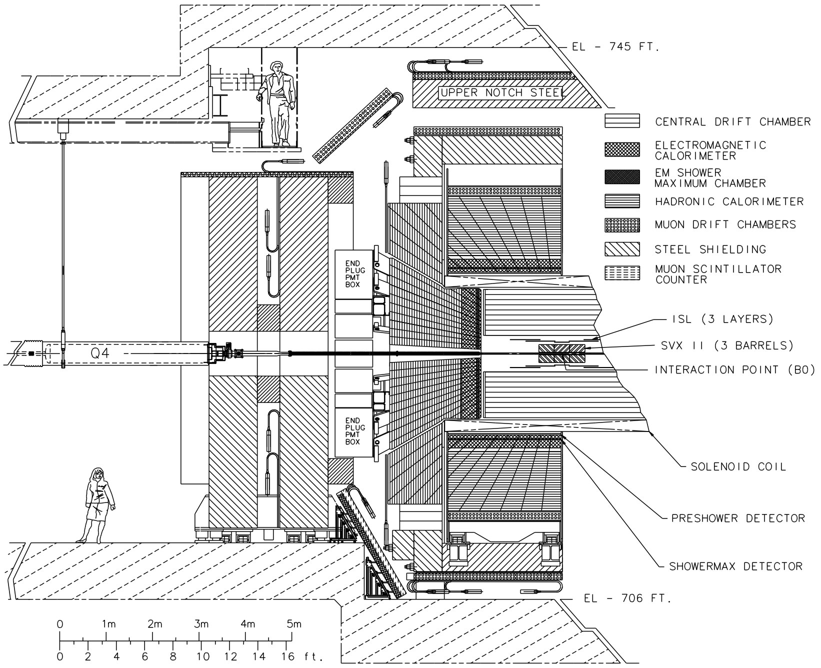


Abbildung 3: Der CDF 2 Detektor.

macht es unmöglich, alle Ereignisse aufzunehmen. Daher verfügt das CDF 2 Experiment über ein aufwändiges System von Ereignisfiltern (Trigger), die nur die physikalisch interessanten Ereignisse auswählen. Der bemerkenswerteste, und für diese Arbeit wichtigste unter diesen Filtern ist der so genannte Zweispur-Trigger. Mittels einer ausgeklügelten Mustererkennung im Silizium-Vertex-Detektor wählt dieser Filter Ereignisse aus, die mit hoher Wahrscheinlichkeit langlebige Teilchen enthalten und reichert damit Zerfälle langlebiger Teilchen in vollständig hadronische Endzustände an. Obwohl der Filter ursprünglich für die Anreicherung von hadronisch zerfallenden B-Mesonen vorgesehen war, sind dies insbesondere auch die D-Mesonen Zerfälle, die in dieser Arbeit untersucht werden. Bis zum November 2004 hat das CDF 2-Experiment allein mit diesem Ereignisfilter 270.103.586 Ereignisse – dies entspricht $\sim 481\mathrm{pb}^{-1}$ integrierter Luminosität – aufgenommen. Nur mit äußerst stabilen und effizienten Rekonstruktionsalgorithmen ist es möglich, aus diesem riesigen Datensatz klare Signale hadronischer Zerfälle mit kleinen Verzweigungsverhältnissen zu gewinnen.

Die Rekonstruktion von Spuren Geladener Teilchen

Die effiziente und genaue Rekonstruktion der Spuren geladener Teilchen ist eine wesentliche Grundlage vieler Analysen in der CDF 2-Kollaboration, und insbesondere der in dieser Arbeit vorgestellten. Aus diesem Grund, und da der Autor wesentliche Beiträge zu deren Umsetzung geleistet hat, wird die Spurrekonstruktion in dieser Arbeit in einiger Ausführlichkeit behandelt. In dieser Zusammenfassung kann nur eine kleine Auswahl der wesentlichen Aspekte diskutiert werden. Die Rekonstruktion geladener Teilchenspuren findet in zwei Phasen statt. Zunächst müssen die zahlreichen Messungen in der Spurkammer und im Silizium-Vertex-Detektor Spurhypothesen zugeordnet werden. Da die Dichte der Messungen um so geringer ist, je weiter entfernt vom primären Wechselwirkungspunkt sie genommen werden, beginnt dieser Prozess in der Spurkammer. Dort gefundene Spuren geladener Teilchen werden dann in den Silizium-Vertex-Detektor extrapoliert, um dort weitere Messungen aufzunehmen. Dies verbessert insbesondere die Ortsauflösung nahe dem Wechselwirkungspunkt. Die Spurkammer deckt allerdings nur einen Bruchteil des Raumwinkels ab. Der Silizium-Vertex-Detektor weist eine höhere Abdeckung auf. Aus diesem Grund sucht ein spezieller, von der Spurkammer unabhängiger, Algorithmus nach Spuren im Silizium-Vertex-Detektor. Dies erlaubt das Auffinden von Spuren in Raumwinkelbereichen, die nicht von der Spurkammer abgedeckt werden, sowie solchen, deren Transversalimpuls nicht ausreicht die Spurkammer vollständig zu durchqueren. Aufgrund der hohen Teilchendichte nahe dem Wechselwirkungspunkt, weisen die so gefundenen Spuren allerdings erhebliche Unreinheiten auf. Diesem Umstand kann nur mit zusätzlicher Information abgeholfen werden. Dies wird erreicht, indem die im Silizium-Vertex-Detektor gefundenen Spuren wiederrum in die Spurkammer extrapoliert werden, um dort weitere Messpunkte aufzunehmen. Dies ist natürlich nur möglich, wenn die Spuren in den Akzeptanzbereich der Spurkammer fallen. Wenn dies der Fall ist, erhöht sich die Qualität der Spuren dadurch allerdings beträchtlich. Die praktische Umsetzung dieser Vorgehensweise wird durch die extrem hohe Teilchendichte in Hadronenkollisionen erheblich erschwert. Um diese hohe Kombinatorik in angemessener Zeit abarbeiten zu können, ist eine hoch effiziente Umsetzung der Extrapolationsroutinen unerlässlich. Diese Aufgabe übernimmt ein auf extrem schnelle Rechenleistung ausgerichteter Kalman-Fitter, der auch in der zweiten Phase der Spurrekonstruktion von großer Bedeutung ist. In dieser zweiten Phase werden die Spurparameter ohne Änderung der aufgenommenen Messungen neu bestimmt. Auf den ersten Blick erscheint dies sinnlos. Es gibt allerdings zahlreiche Gründe, die diese zweite Phase notwendig machen. Der Wichtigste ist in Materialeffekten begründet. Geladene Teilchen wechselwirken mit dem Detektormaterial, welches sie durchqueren. Dies ist zum einen die notwendige Voraussetzung um überhaupt eine Messung vornehmen zu können. Andererseits zieht es auch eher unerwünschte Effekte nach sich. So verlieren die Teilchen bei dieser Wechselwirkung Energie und ihre Flugbahn wird durch Vielfachstreuung abgelenkt. Entscheidend ist nun, dass die Stärke dieser Effekte, neben den Eigenschaften des durchquerten Materials, von der Masse der Teilchen abhängt. Wenn also z.B. der Zerfall $D^0 \to K^-\pi^+$ rekonstruiert werden soll, ist es notwendig die negativ geladene Spur mit einer Kaon-Hypothese neu anzupassen. Die Änderung der Teilchenhypothese von π auf K ist dabei der triviale Teil. Um die Masse des D-Mesons korrekt zu rekonstruieren, ist es notwendig, auch alle Materialeigenschaften richtig zu behandeln. Um dies mit akzeptablen Zeitverhalten zu erreichen muss ein erheblicher technischer Aufwand betrieben werden. Aus diesem Grund ist Behandlung dieses Themas ein Schwerpunkt dieser Arbeit.

Methodik der Korrelationsanalyse

Ausgehend von der hohen Qualität der geladenen Teilchenspuren ist es möglich ein klares Signal für den Zerfall $D^0 \to K^-\pi^+$ aus dem großen Untergrund, der nach wie vor in den vom Zweispur-Trigger gesammelten Datensatz vorhanden ist, zu extrahieren. das Vorhandensein von Korrelationen lässt sich dann aus den relativen Zählraten eines Histogramms extrahieren, in dem die Massenspektren der $\bar{D}^0$ und D^0 gegeneinander aufgetragen sind. Die hierfür verwendete Methode setzt voraus, dass das Massenspektrum symmetrisch ist und sich in einen zentralen Signalbereich und zwei außenliegende Untergrundbereiche einteilen lässt. Wie man in Abb. 4 sehen kann, ist

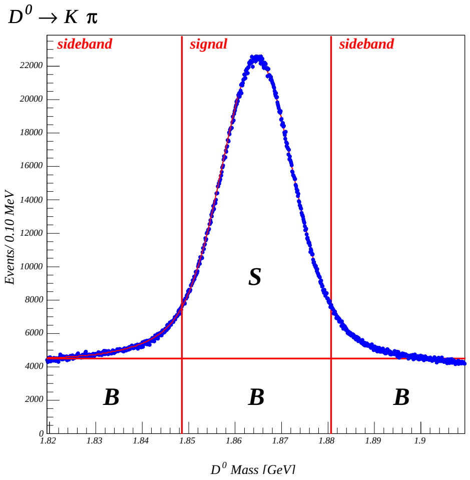
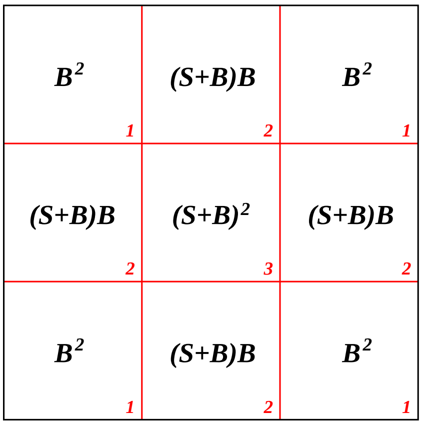


Abbildung 4: Das $D^0 \to K^-\pi^+$ Signal mit der Aufteilung in Signal- und Untergrundregion sowie den entsprechenden Wahrscheinlichkeiten.



B^2 1	(S+B)B 2	B^2 1
(S+B)B 2	$(S+B)^2$ 3	(S+B)B 2
B^2 1	(S+B)B 2	B^2 1

Abbildung 5: Illustration der Korrelationsanalyse. Die Ausdrücke in den Feldern bezeichnen die relativen Wahrscheinlichkeiten für den Fall unkorrelierter Produktion.

dies für das rekonstruierte D^0 Massenspektrum ist in guter Näherung der Fall. Unter diesen Voraussetzungen, kann man einfache Ausdrücke für die Wahrscheinlichkeiten angeben, einen Massenwert entweder in der Signaleregion $(S + B)$, oder einer der Untergrundregionen (B) zu finden. Im Falle unkorrelierter Produktion führt dieser Ansatz direkt auf die Wahrscheinlichkeiten (bzw. relativen Zählraten) im zweidimensionalen Massenspektrendiagramm, wie sie in Abb. 5 eingetragen sind. Damit ist es möglich unter der Annahme unkorrelierter Produktion aus den acht äußeren Bins eine Vorhersage für den zentralen Bin abzuleiten. Wenn die tatsächliche Zählrate signifikant über der vorhergesagten liegt, stellt dies eine Evidenz für korrelierte D^0-$\bar{D}^0$-Produktion dar. Die dargestellte Methode der Korrelationsanalyse wurde mit Hilfe eine Toy-Monte-Carlo-Studie erfolgreich auf ihre Anwendbarkeit überprüft. Die Anwendung auf das $\bar{D}^0$-D^0-Massenspektrum liefert eine überzeugende Evidenz von 5.9 σ für korrelierte D^0-$\bar{D}^0$-Produktion.

Schlussbemerkung

Im Run 2, der zweiten Operationsphase des Tevatron Beschleunigers, hat das CDF 2-Experiment eine große Menge an verwertbaren Daten genommen und erfolgreich reprozessiert. Dies ist ein großer Erfolg, der unter Anderem den zahlreichen Verbesserungen der Detektor-Hardware zu verdanken ist. Die Möglichkeit, diese riesige Datenmanege in effizienter Form auszuwerten, verdankt die CDF 2-Kollaboration aber nicht zuletzt den in dieser Arbeit vorgestellten Rekonstruktionsalgorithmen. Nur mit den hier vorgestellten Methoden der Spurrekonstruktion ist es möglich, Signale für Zerfälle von D-Mesonen in vollständig hadronische Endzustände mit kleinem Verzweigungsverhältnis zu extrahieren. Zum ersten Mal an einem Hadronenbeschleuniger ist es mit Hilfe der hier entwickelten experimentellen Methoden gelungen, die korrelierte Produktion von D-Mesonen nachzuweisen.

A highly efficient Geometry Model for Track Reconstruction in the Silicon Vertex Detector at CDF 2
and
Search for Correlated Charm Production

Kurt Rinnert

Ph.D. Thesis
Faculty for Physics
University of Karlsruhe (TH)

Supervisor: Prof. Dr. M. Feindt, Institut für Experimentelle Kernphysik, University of Karlsruhe (TH)

Contents

List of Figures

List of Tables

Introduction

In the year 2004, at the time of this writing, the Nobel Prize in physics was awarded to David J. Gross, H. David Politzer and Frank Wilczek for their ground breaking work on a non-abelian Quantum Field theory called Quantum Chromodynamics (QCD). They showed that QCD displays the feature of asymptotic freedom, rendering it suitable for describing the strong interactions of quarks and gluons. With this result the interpretation of quarks as physically real entities became much more convincing. Since then it was possible to compute many properties of quark dynamics using QCD. On the other hand many important properties can still not be computed from first principles. Among these are fragmentation of heavy quarks into hadrons and their production in hadron-hadron collisions. But these aspects are of utmost importance when linking experimental observations to theoretical predictions since free quarks are not observed experimentally.

This thesis is concerned with the essential ingredients for the analysis of the production of heavy quark pairs in hadron-hadron collisions as they are recorded by the CDF 2 experiment. To establish such a signal in a hadron collider environment is a highly non-trivial task. It requires a sophisticated detector hardware as well as a highly efficient and stable reconstruction software, capable of coping with the large amount of data collected by a hadron collider experiment. Here, the meaning of efficiency is twofold: it denotes a measure of the ability to find what is there as well as computing efficiency in terms of CPU time and memory consumption. A theoretically perfect algorithm is worth nothing if it is not implemented in such a way that it reliably runs on a real-world computer. The importance of reconstruction software and the complications arising during its implementation are frequently underestimated parts of the experimental design. All too often, a slight change in the design of the detector hardware, e.g. a change of the bonding scheme in one layer of a silicon vertex detector, creates unforseen obstacles on the way to an efficient reconstruction code. On the other hand it is certainly impossible to predict all implications of any non-trivial design, may it be hardware or software. This is especially true in the case of systems for charged particle track reconstruction at hadron collider experiments like CDF 2. It takes much experience and many development cycles to finally come up with reconstruction code that makes it feasible to run an analysis on hundreds of millions of events. Consequently, much effort had to be put in the development of track reconstruction and track fitting algorithms. This resulted in a highly accurate and efficient implementation which is the basis of all track-based analyses in the CDF 2 collaboration. Many aspects of this software implementation will be discussed in detail in chapter 3, emphasising the crucial role of an accurate model of the detector geometry. Only with these powerful tools at hand, it was possible to establish evidence for correlated D^0-$\bar{D}^0$ production for the first time at a hadron collider experiment.

In chapter 1 the implications of hadronisation models on the experimental observables is discussed in more detail. It will be shown that the angular distribution of the meson momenta is a useful variable for distinguishing between various production mechanisms once evidence for correlated Charm production is established. This, and the other possible sources of correlated Charm, like decays of exotic particles, will not be subject of any analysis presented in this thesis. The significance of the raised questions, however, emphasises once more the importance of the algorithms used for extracting clean signals of hadronic decay modes with low branching ratios from the large background present in hadron collider experiments.

The hardware aspects of the experimental setup will be discussed in chapter 2. For the first time at a hadron collider experiment CDF 2 has commissioned a displaced track trigger. Until November 2004 CDF 2 has collected 270,103,586 events, corresponding to $\sim 481\mathrm{pb}^{-1}$ of data, with this trigger alone. Not relying on lepton signatures, the displaced track trigger accumulates hadronic decays of long-lived particles. This dataset is the basis of many analyses in the CDF 2 collaboration and especially of the correlation analysis presented in chapter 4.

Clear high statistics signals of the three charmed meson decays $D^0 \to K^-\pi^+$, $D^+ \to K^-\pi^+\pi^+$ and $D^{*+} \to D^0(\to K^-\pi^+)\pi_s^+$ are established in chapter 4. The mass estimates resulting from fits to the reconstructed mass spectra are in very good agreement with the known meson masses. This impressively demonstrates the accuracy of the geometry model underlying the track fit algorithm. After establishing these signals, a correlation analysis method is introduced and validated using a toy Monte Carlo study. This method is then applied to the D^0 and $\bar{D}^0$ mass spectra, yielding a convincing evidence for correlated D^0-$\bar{D}^0$ production with a significance of 5.9 σ.

Chapter 1

Sources of Correlated Open Charm

Introduction: definition of correlated open charm; restrictions imposed by experiment — Prompt Production of Charmed Hadrons: overview; perturbative ansatz and leading order contributions; higher order contributions; additional higher order contributions — Open Charm from Decays: B decays; charmonium and exotic state decays

1.1 Introduction

Definition of Correlated Open Charm.—An event is said to contain open Charm when there is at least one charmed meson or baryon present in the event. Note that this is not the same as requiring a c or $\bar{c}$ quark. Charmonium, i.e. bound states with $c\bar{c}$ quark content like the J/ψ, are not open charm since the overall charm content of these hadrons is zero. They may, however, decay into charmed mesons thereby creating open Charm. Such decays are one possible source of *correlated* open Charm, see section 1.3. Correlated open Charm is then defined by the presence of two charmed hadrons originating from the same interaction. Correlated open Charm production is not at all an unexpected phenomenon. There are several known sources contributing to it. These will be discussed in the following sections. It is, however, a phenomenon that is hard to establish experimentally in a hadron collider environment.

Restrictions Imposed by Experiment.—While the above definition includes all charmed hadrons and all possible decay modes, only a subset of them is experimentally accessible. In this analysis we furthermore restrict ourselves to fully charged final states that can be reconstructed exclusively. For charged current weak decays this implies fully hadronic final states. For practical reasons that will become evident in chapter 3 we concentrate on low multiplicity final states of the most abundant mesons. Table 1.1 lists the mesons and decay modes used in the analysis presented in this thesis[1].

[1] Throughout this thesis charge conjugate states are always implied unless otherwise stated.

Besides the reconstruction efficiency the relatively low branching ratios of these final states are the major statistics limiting factors[2]. Given this list the signature for correlated open Charm is the presence of any combination of two mesons that can be traced back to a common production vertex. There are many interesting questions raised in the following sections. Most of them will not be investigated further in this thesis. They do, however, clearly show the importance of the experimental tools described in this document. Without these tools, there would be no hope of ever resolving a single one of the questions in the framework of the CDF 2 experiment.

Meson	Decay Mode	Branching Ratio[1]
D^0	$D^0 \to K^-\pi^+$	$(3.80 \pm 0.09) \times 10^{-2}$
D^{*+}	$D^{*+} \to D^0\,\pi^+$	$(67.7 \pm 0.5) \times 10^{-2}$
D^+	$D^+ \to K^-\pi^+\pi^+$	$(9.1 \pm 0.6) \times 10^{-2}$

Table 1.1: Charmed meson decay modes and their branching ratios.

1.2 Prompt Production of Charmed Hadrons

Overview.—The first source of correlated open charm to be considered is the production of charmed mesons directly in the proton anti-proton collision. This is also called prompt production as opposed to the production in decays of long lived particles. Theoretically the interactions of quarks as coloured particles is described in the framework of QCD. But it is not yet possible to calculate the whole dynamics of hadron production in hadron-hadron collisions starting from this *Quantum Field Theory*. Especially two aspects are hard to describe theoretically, namely the internal dynamics of the incident particles and the fragmentation of the partons (quarks and gluons) produced in the collision into hadrons. Thus it is necessary to employ models in order to get a handle on these aspects. The incident particle dynamics are described by *Parton Distribution Functions*. These functions model the parton content and the momentum distribution among the partons. They can be measured in deep inelastic scattering experiments. Fragmentation models are not as accessible experimentally. One commonly used fragmentation model is the *Lund String Model*[2]. In this picture, a string represents the colour confinement field between two quarks squeezed into a tubular region between the quarks, giving rise to a linear confinement potential. Nevertheless, perturbative QCD can successfully be applied to short distance processes in which heavy quarks are produced from the incident particle partons. This will be the

[2] For the $D^{*+} \to D^0\pi^+$ decay the D^0 branching ratios have to be folded in.

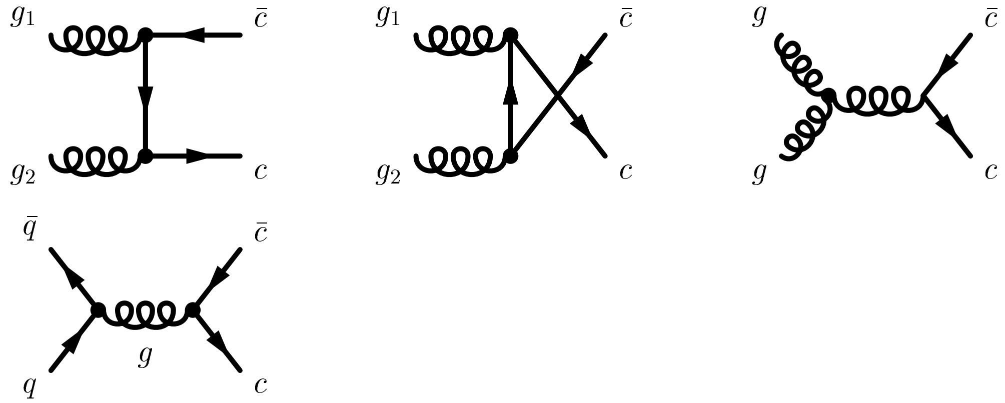


Figure 1.1: Leading order diagrams for heavy quark anti-quark pair production. Gluon-gluon fusion (above) and quark-anti-quark annihilation (below).

starting point for the discussion of heavy quark production in the remainder of this section.

Perturbative Ansatz and Leading Order Contributions.—The perturbative ansatz assumes that the overall cross section can be factored in such a way that the short distance cross section is no longer sensitive to momentum scales below the heavy quark mass[3]. Corrections are then suppressed by powers of the heavy quark mass. This ansatz allows to expand the short distance cross section in powers of $\alpha_s(\mu^2)$ at a given scale μ. The lowest order terms in this series are of $\mathcal{O}(\alpha_s^2)$. At this order there are contributions from quark-anti-quark annihilation and gluon-gluon fusion,

$$\begin{aligned} q + \bar{q} &\rightarrow Q + \bar{Q} \\ g + g &\rightarrow Q + \bar{Q} \end{aligned}$$

At *Tevatron* energies the main contribution comes from gluon-gluon fusion. The corresponding diagrams are shown in fig. 1.1.

Higher Order Contributions.—In the case of the charm quark the mass suppression is not as strong a in the case of the bottom quark. Thus the production is more sensitive to higher order corrections and it is necessary to include them. The diagrams contributing at order $\mathcal{O}(\alpha_s^3)$ are obtained by adding real or virtual gluons to the lowest order diagrams. There is a vast number of such diagrams due to the many possible places at which gluons can be attached. Some of them are shown in fig. 1.2.

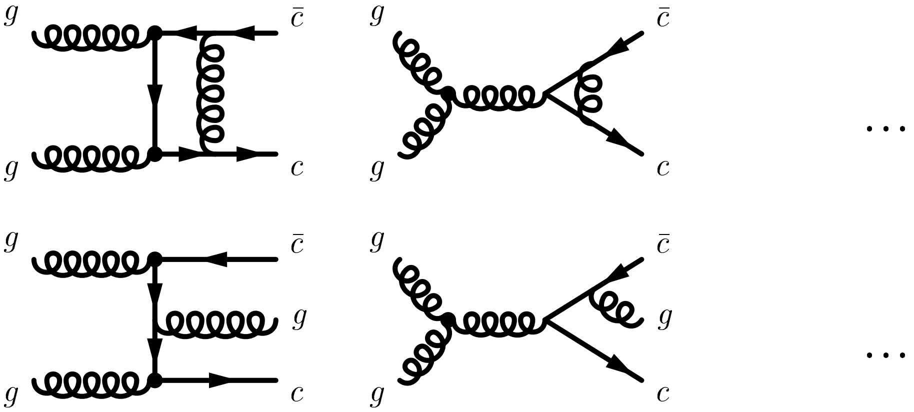


Figure 1.2: Some higher order diagrams for heavy quark anti-quark pair production obtained by adding virtual (above) or real (below) gluons.

Additional Higher Order Contributions.—Further considerations suggest that additional higher order diagrams might contribute significantly to the charm anti-charm productions cross section. The lowest order cross section for the process $gg \to gg$ is about a hundred times larger than that of the process $gg \to Q\bar{Q}$. Thus the gluon splitting process $gg \to g(\to Q\bar{Q})g$ can numerically compete with the lowest order gluon fusion, although it is formally of $\mathcal{O}(\alpha_s^3)$[3]. Additionally, non-perturbative fragmentation effects can alter the properties of the emerging hadrons, thereby affecting the relative experimental accessibility of the various production processes[4].

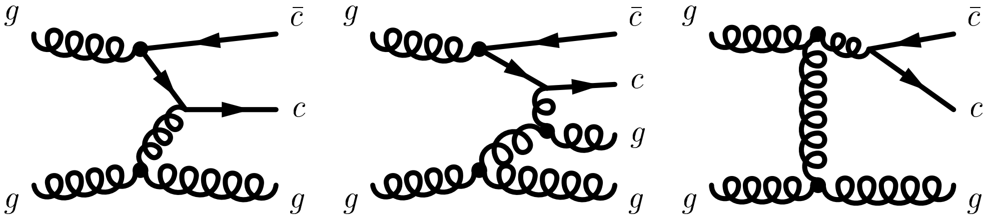


Figure 1.3: Diagrams for additional higher order production processes.

Some examples of additional higher order diagrams are shown in fig. 1.3. Although all of them involve $g \to Q\bar{Q}$ vertices, they are usually subdivided into different classes[4].

- *flavour excitation* processes, when a heavy flavour from the parton distribution of one incident particle is put on mass shell by scattering against a parton of the other incident particle, fig. 1.3 (left).

- *gluon splitting* processes, when no heavy flavour enters the hard scattering, and $g \to Q\bar{Q}$ branching occurs in the initial or final state shower, fig. 1.3 (right).

The classification of the process in fig. 1.3 (centre) is a little ambiguous. It is usually classified as gluon splitting since it does not contain heavy flavours in the hard scattering.

From the experimental point of view it is interesting that some distributions will look quite different for these processes compared to gluon-gluon fusion: in the latter process the heavy quarks are produced back to back, leading to a large angular difference of the momenta of the observable hadrons. On the other hand, heavy quark pairs produced via gluon splitting can lead to nearly collinear hadron momenta, a clear experimental signature once correlated open charm production is established.

1.3 Open Charm from Decays

B Decays.—Bottom quarks dominantly decay into Charm quarks via the emission of a virtual W boson as shown in fig. 1.4. The transition $b \to W^- + u$ is suppressed by $|V_{ub}|/|V_{cb}|$. Consequently there is almost always one D meson among the decay products.

Whether there is another D meson and its quark content depend on the way the W boson branches. There are only to possibilities that produce a second D meson:

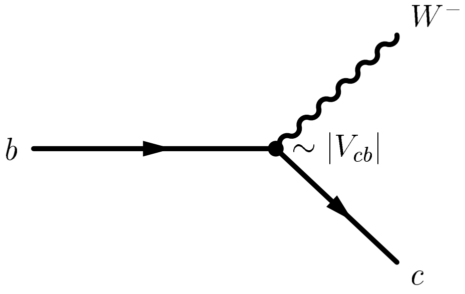


Figure 1.4: The $b \to W^- + c$ vertex.

either $W^- \to \bar{c} + s$ or $W^- \to \bar{c} + d$. The latter is Cabbibo suppressed by $|V_{cd}|/|V_{cs}|$. The diagram shown in fig. 1.5 covers all the possibilities, namely $B^+ \to \bar{D}^{(*)0} D_{(s)}^{(*)+}$, $B^0 \to D^{(*)-} D_{(s)}^{(*)+}$ and $B_s^0 \to D_s^{(*)-} D_{(s)}^{(*)+}$. The indices in parentheses are optional. The experimental signature of correlated open Charm from B decays is different than the one for prompt production due to the long lifetime of the B mesons. Establishing a signal for correlated open Charm from B decays enables a measurement of the relative branching ratios of the various decay modes listed above.

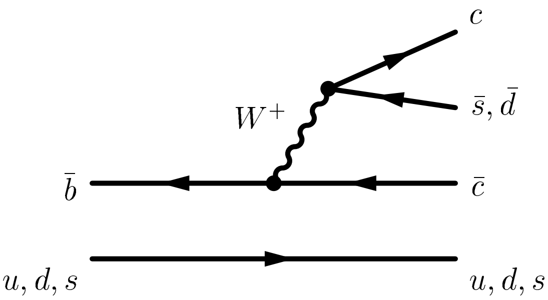


Figure 1.5: Diagrams for B decaying weakly into $D\bar{D}$.

Charmonium and Exotic State Decays.—Other possible sources of correlated open Charm are Charmonium and exotic state decays. Charmonium decays are interesting because there is still not much known about the $c\bar{c}$ states that have a significant branching ratio into $D\bar{D}$, namely $\psi(3770)$ and $\psi(4040)$[1].

Even more interesting is the investigation of exotic state decays into $D\bar{D}$. Recently the Belle collaboration reported the observation of a previously unknown narrow resonance at $m \sim 3872$ GeV[5] decaying into the final state $J/\psi(\to \mu^+\mu^-)\pi^+\pi^-$, now called $X(3872)$. This unexpected discovery triggered many interesting research activities in both, theory and experiment. After the discovery had been confirmed by the CDF 2 [6] collaboration the experimental focus shifted towards the determination of the properties of the $X(3872)$. The CDF 2 collaboration is currently working on a measurement of the spin and parity quantum numbers[7]. It is still possible that charged isospin partners of the $X(3872)$ with a mass below the $D^0 D^{*+}$ threshold exist. These partners could decay into $D^0 \bar{D}^0 \pi^{\pm}$. Thus establishing a signal for correlated $D^0 \bar{D}^0$ production might help to shed some light on the properties of the X.

Chapter 2

Experimental Setup

The Tevatron: overview; the accelerator chain; performance — The CDF 2 Detector: overview; the coordinate system; the tracking system; particle identification; calorimeters; the muon system — The CDF 2 Trigger System: overview; the Two Track Trigger

2.1 The Tevatron

Overview.—The *Tevatron* is a symmetric proton and anti-proton collider ring with a circumference of 2π km, located at the Fermi National Accelerator Laboratory (Fermilab) in Batavia/Illinois, USA. In Run 1, the first phase of operation from 1985 to 1996, the *Tevatron* was running at a centre-of-mass energy of $\sqrt{s} = 1.8$ TeV. Among the highlights of physics results from Run 1 are the the first experimental evidence for the top quark[8] and a high accuracy measurement of its mass[9]. In the ongoing Run 2, the second phase of *Tevatron* operation started in 2001, the two *Tevatron* experiments CDF 2 and DØ are pursuing physics goals such as measuring the frequency of $B_s^0 \bar{B}_s^0$ oscillations, Top-Quark physics, Higgs searches and analyses of rare physical processes. To this end the *Tevatron* was upgraded to achieve higher instantaneous luminosity and a centre-of-mass energy of $\sqrt{s} = 1.96$ TeV.

The accelerator chain.—Fig. 2.1 shows a schematic view of the Fermilab accelerator chain for Run 2. In the first step of acceleration negatively charged hydrogen ions are produced in a Cockcroft-Walton pre-accelerator. The ions leave the Cockcroft-Walton with an energy of 750 keV and are then injected into the *Linac*, a linear accelerator 150 m in length. The *Linac* accelerates the ions to an energy of approximately 400 MeV. The protons produced by stripping the electrons off the hydrogen ions are then fed into the *Booster*, a 150 m diameter synchrotron. When leaving the *Booster*, the protons have an energy of about 8 GeV. Before injection into the *Tevatron* they undergo a final pre-acceleration in the *Main Injector* which gives them an energy of 120 GeV.

The *Main Injector* proton beam is also used to produce the anti-protons by focusing it onto a fixed nickel target. After separating the protons from the numerous different particles emerging from this collision, they are focused and stored in the *Accumulator Ring*. Once a sufficiently large number of anti-protons is stored, they are fed back into the *Main Injector* where they are accelerated to 120 GeV before injection into the *Tevatron*. In the *Tevatron*, the proton and anti-proton beam get their final energy of 0.98 TeV, yielding the centre-of-mass energy of $\sqrt{s} = 1.96$ TeV.

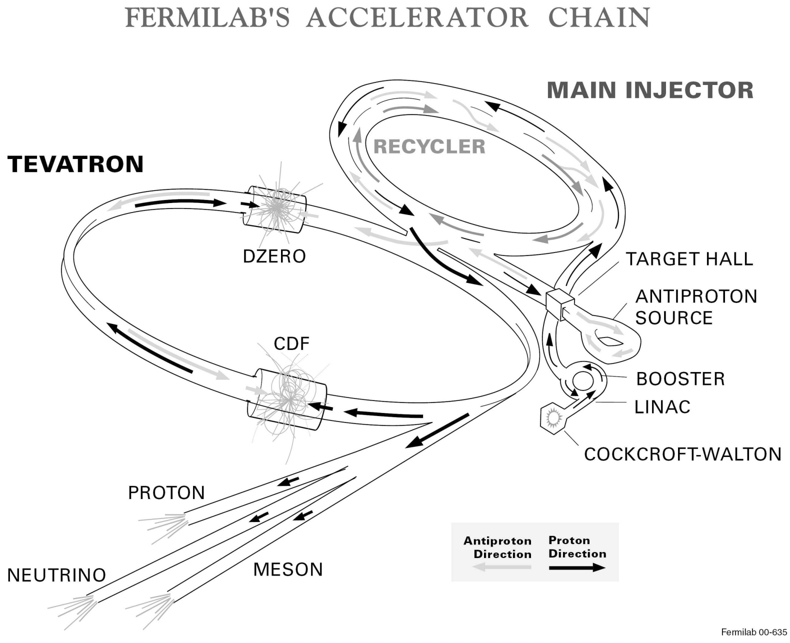


Figure 2.1: The Fermilab accelerator complex for Run 2.

Performance.—Already in Run 1 the anti-proton production was the major limiting factor of the *Tevatron* efficiency. In order to improve the situation for Run 2 the *Recycler* was introduced. The idea is to re-use the remaining anti-protons after a *Tevatron* store. About 75% of the anti-protons are expected to survive a store. These are decelerated down to the energy of 8 GeV in the *Main Injector* and then stored in the *Recycler* for re-use in the next *Tevatron* fill. Unfortunately, in the beginning of Run 2, the *Recycler* could not be commissioned as planned and anti-protons were vanishing at a higher rate as expected. Thus the Run 2 design luminosity of $\mathcal{L} = 10^{32}\,\mathrm{cm}^{-2}\mathrm{s}^{-1}$ was not reached immediately. While this has delayed some of the physics goals of *Tevatron* Run 2, the Fermilab Accelerator Division has meanwhile identified and solved the major problems and the *Tevatron* is now working close to Run 2 design specifications. Figures 2.2 and 2.3 show the *Tevatron* Run 2 integrated luminosity and peak luminosity, respectively[1].

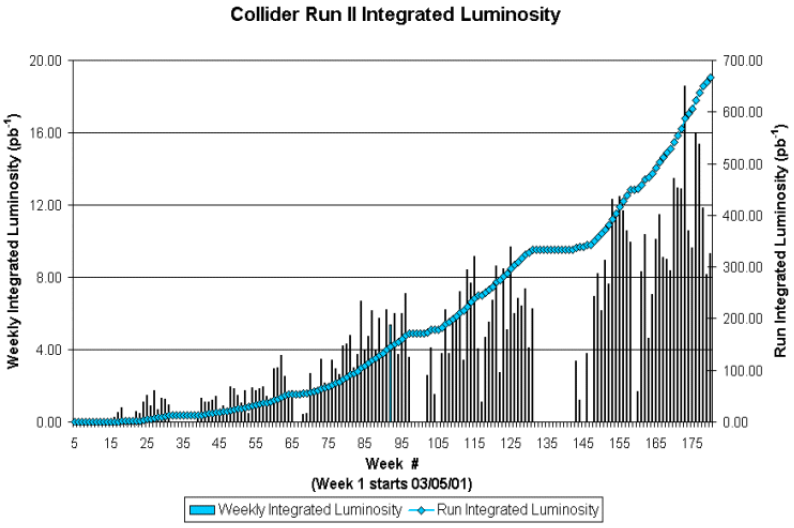


Figure 2.2: Integrated *Tevatron* Run 2 luminosity.

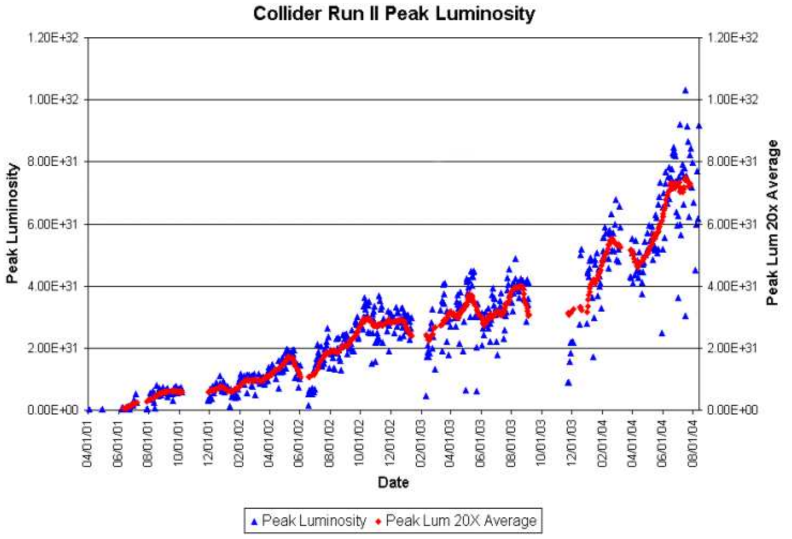


Figure 2.3: *Tevatron* Run 2 peak luminosity.

[1] As of October 19th 2004.

2.2 The CDF 2 Detector

Overview.—The CDF 2 detector is a general purpose collider detector[10]. It features a vertexing and tracking system, particle identification, a superconducting solenoid generating a 1.4 T magnetic field, calorimetry and muon chambers. The components are arranged in the cylindrical symmetry typical to collider detectors. Fig. 2.4 shows a side view of the CDF 2 detector.

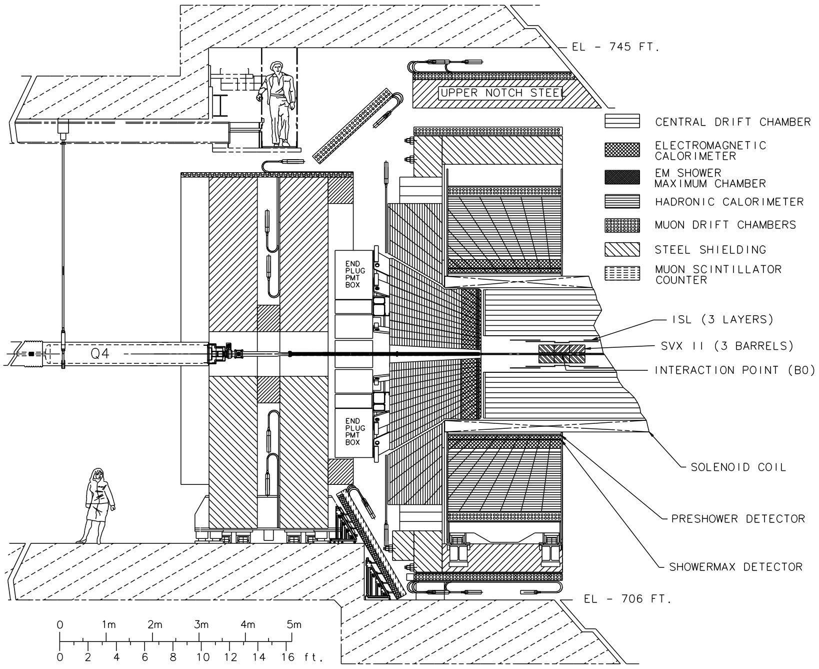


Figure 2.4: The CDF 2 detector.

The Coordinate System.—It will prove useful to define the coordinate system used in CDF 2 since we will frequently refer to it. We use a right-handed coordinate system in which the positive z direction is defined by the direction of the proton beam. Geographically, the proton beam points east at the location of the CDF 2 detector. The polar angle θ is measured from the positive z direction and the azimuthal angle ϕ is measured from the plane defined by the *Tevatron* ring. Another frequently used variable is the pseudo-rapidity η which is defined as $\eta = -\ln(\tan(\theta/2))$. The pseudo-rapidity has the unique property that in hadron-hadron collisions the particle density is almost constant in equal intervals of η. It can be seen from the definition that this means that the vast majority of particles emerging from a collision go into the forward and backward directions and hence can not be detected. Throughout this thesis we will adopt the commonly used jargon that “forward” refers to the forward *and* backward directions, i.e. θ close to 0 or π.

The Tracking System.—Precise and efficient reconstruction of charged particle tracks is the core ingredient of many CDF 2 analysis and especially important to the work presented in thesis. Hence we will describe it in a little more detail.

The tracking system consists of two major components: the *Central Outer Tracker* (COT) and a silicon vertex detector. The COT is a cylindrical drift chamber 304 cm in length along the z axis, covering the radial region $44\,\text{cm} < r < 132\,\text{cm}$. This corresponds to a pseudo-rapidity coverage of $|\eta| < 1$. The COT has eight super-layers with twelve measurements in each super-layer. Four out of the eight super-layers are axial layers, measuring only track parameters in the r-ϕ plane. The remaining four super-layers add z information by virtue of a stereo angle of $\pm 3°$. Due to pure geometrical reasons the track density COT is much lower than in the silicon detector. In combination with the large lever arm of 88 cm and the high number of measurements per track this enables high purity charged particle tracking with excellent momentum resolution. However, the z parameter and pointing resolution at the collision point is poor compared to that of the silicon detector.

The silicon tracking system consists of three subsystems with different features, namely the SVX II the *Intermediate Silicon Layers* (ISL) and the so-called *Layer 00* (L00). The silicon detector system allows for standalone charged particle tracking independently of the COT in the forward regions. The geometry and pseudo-rapidity coverage of the silicon detector is illustrated in fig. 2.5.

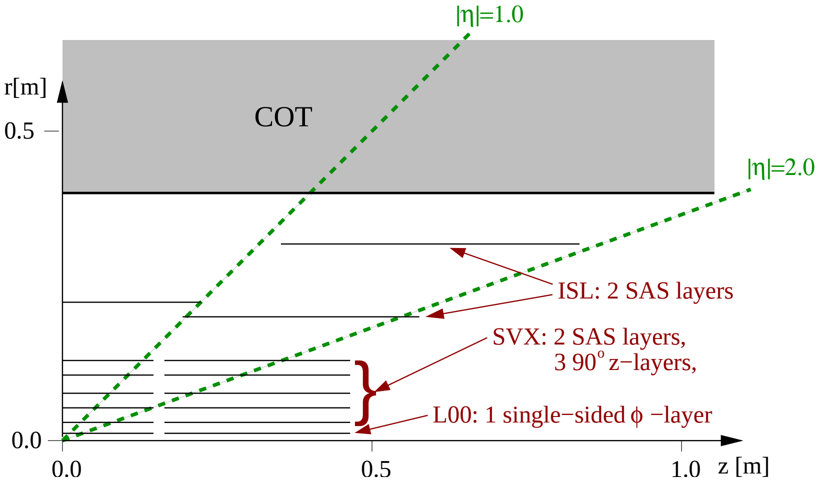


Figure 2.5: Geometry of the CDF 2 silicon detector.

The SVX II is organised in three barrels with five layers of double-sided silicon micro-strip sensors arranged in twelve wedges. One side of the silicon sensors measure r-ϕ parameters with strips parallel to the z axis. The strips on the other side are tilted by a stereo angle and allow z parameter measurements. There are two different types of sensors with respect to the stereo angle: 90° stereo and *Shallow Angle Stereo* (SAS) sensors with an angle of $\pm 1.2°$. While the z resolution of the latter is poor compared to the 90° sensors, the number of ambiguous combinations with axial strips is significantly lower. This is illustrated in fig. 2.6. In the 90° stereo layers the situation is further complicated by readout channel multiplexing. In order to limit the total number of readout channels, up to three strips on different sensors are combined in one channel. In these cases the decision which of the strips corresponds to the readout channel can only be made when a track hypothesis is already available. Contrary to the situation in the 90° stereo layers, the lower ambiguity in the SAS layers makes it possible to combine strips from both sides to a three dimensional measurement independently of a track hypothesis. This does not only provide an independent three dimensional measurement. In such a combination the contributing hits also confirm each other. The silicon pattern recognition algorithms benefit a lot from this, as will become evident in section 3.2. The SVX II has two SAS and three 90° stereo layers.
The ISL is located between the SVX II and the COT. Its main purpose is to provide measurement points close to the COT when pursuing drift chamber tracks into the silicon detector. Both ISL layers are SAS layers. Due to this, and its large η coverage up to $|\eta| = 2$ it also plays a crucial role in silicon standalone tracking.

The innermost subsystem of the silicon detector is L00. It is composed of two

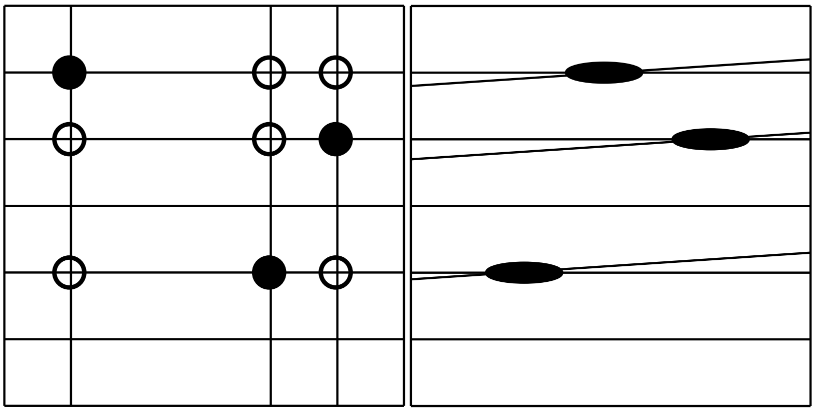


Figure 2.6: Ambiguities in the combination of axial and stereo strips in 90° and 1.2° stereo layers. Correct combinations are indicated by filled blobs, possible but wrong combinations by empty circles.

layers of single-sided micro-strip silicon sensors, mounted directly onto the beryllium beampipe. The purpose of L00 is to improve the the r-ϕ resolution close to the interaction point. It bears its name because it was proposed beyond baseline[11] and the name “layer 0” – reflecting the index counting of the C++ programming language – was already assigned to the innermost SVX II layer. After L00 was added to the detector description it technically got assigned the index ’0’. To avoid confusion by different numbering schemes we will use the offline software numbering scheme from table 2.1 throughout this thesis.

The mechanical dimensions of the silicon sensors in the various detector components differ considerably. Most importantly the strip pitch varies in the different layers of SVX II and ISL as can be seen in table 2.1. How this affects the resolution of the position measurements will be discussed in more detail in section 3.2.

			pitch [μm]			
			axial		stereo	
detector	layer	stereo angle [°]	strip	readout	strip	readout
L00	0	n/a	25.0	50.0	n/a	
SVX II	1	90.0	60.0	60.0	141.0	141.0
	2	90.0	62.0	62.0	125.5	125.5
	3	+1.2	60.0	60.0	60.0	60.0
	4	90.0	60.0	60.0	141.0	141.0
	5	-1.2	65.0	65.0	65.0	65.0
ISL	6	+1.2	55.0	110.0	73.0	146.0
	7	+1.2	55.0	110.0	73.0	146.0

Table 2.1: Stereo angle, strip and readout pitch of the different silicon detector components

Particle Identification.—Besides the muon chambers, the only detector component with the sole purpose of identifying particles is the *Time of Flight* detector (TOF). The TOF is mounted just outside the COT inside the solenoid as shown in fig. 2.7.

For a given particle, the combination of the momentum measurement from the COT and the time of flight measurement allows to compute the particles mass. Thus one can distinguish particles of different mass, especially protons, K-mesons and π-mesons. This only works for charged particles since neutrals can not be detected in the drift chamber. The distinction of K and π-mesons is of utmost importance for many B physics analysis, especially $B_s^0\bar{B}_s^0$ oscillations. In Run 1 the only way to distinguish

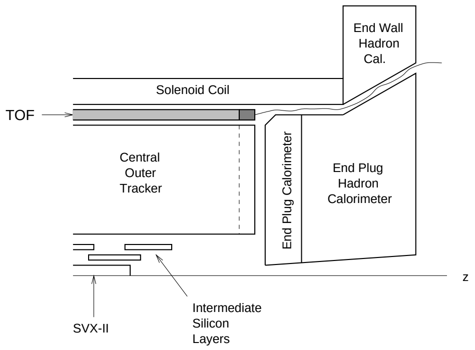


Figure 2.7: Location of the TOF system in the CDF 2 detector.

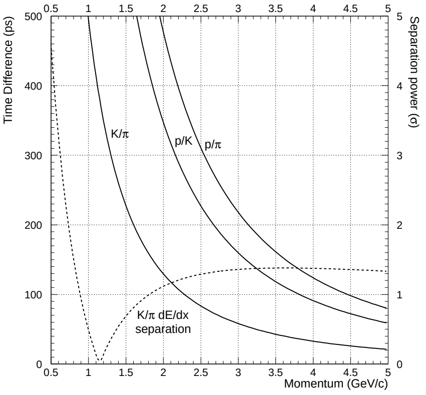


Figure 2.8: Time of flight differences as a function of particle type and momentum. The dashed line indicates K/π separation power of COT dE/dx.

K-mesons from π-mesons was their different energy loss in the drift chamber. Both methods have a different resolution dependence on the particle momentum. The new TOF detector is especially powerful in the momentum region between one and two GeV where the drift chamber dE/dx does not give much information. This is illustrated in fig. 2.8. Other means of particle identification involve the electromagnetic calorimeter in combination with the COT for electrons, the muon chambers and dE/dx in the silicon vertex detector. The best overall separation is achieved by combining the information from all detector components properly, e.g. using a neural network.

Calorimeters.—The CDF 2 detector features several calorimeters: central electromagnetic and hadron calorimeters, end-wall hadron calorimeters and end-plug electromagnetic and hadron calorimeters. The calorimetry covers the whole azimuth range and the pseudo-rapidity region $|\eta| < 3.64$. The calorimeters are designed to measure the energy of hadronic jets photons and electrons. They are mounted outside the solenoid, as can be seen in fig. 2.4.

The Muon System.—The muon system is the outermost part of the CDF 2 detector. It consists of scintillators drift cells and steel absorbers. Usually only muons reach the muon chambers since all other particles are stopped inside the calorimeter or the steel absorbers. In order to reach the muon chambers a muon must have a momentum of $\sim$ 1.5 GeV. This setup allows for very pure decisions on whether there was one ore more muons produced in a collision, rendering the muon system an indispensable tool for physics analysis involving $J/\psi \to \mu^+\mu^-$ or semi-leptonic meson decays and the corresponding triggers.

2.3 The CDF 2 Trigger System

Overview.—The collision rate at the *Tevatron* is much higher than the rate at which data can be stored on mass storage. Thus the trigger plays an important role in selecting the interesting events from a huge background. There is a large number of trigger paths implemented in the CDF 2 trigger system, reflecting the broad physics programme of Run 2. It is beyond the scope of this document to describe them all in detail. We will restrict ourselves to a general overview and a more detailed description of the *Two Track Trigger*; the latter being especially important for analysis presented in this thesis.

The CDF 2 trigger is a three level system with an overall rejection of 120,000:1. This reduces the collision rate to an event output rate of ~ 50 Hz. Given the a typical event record size of 200-300 kB this results in ~ 12 MB/s written to mass storage for offline analysis. Fig. 2.9 shows a schematic view of the data-flow in the trigger.

The first (L1) and second (L2) trigger levels are implemented in hardware and

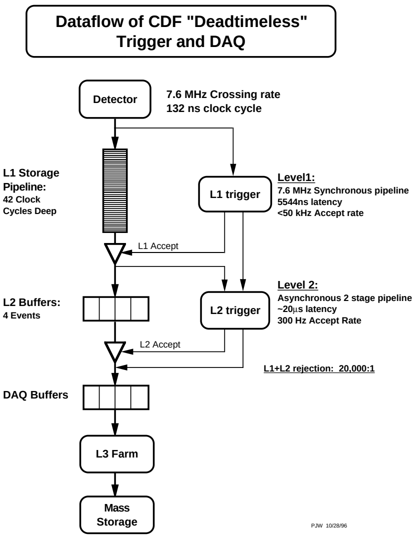


Figure 2.9: Data-flow in the CDF 2 trigger and data acquisition system.

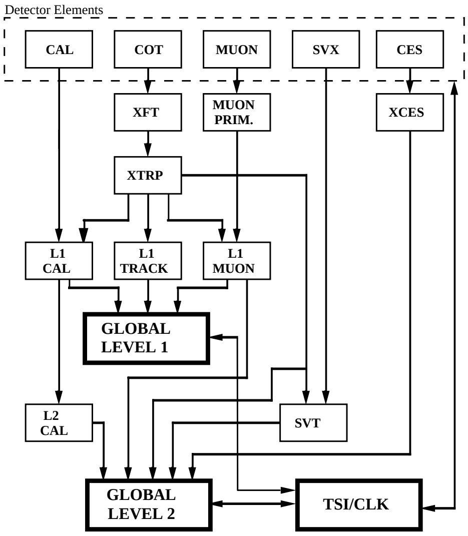


Figure 2.10: The first two trigger levels and the involved detector elements.

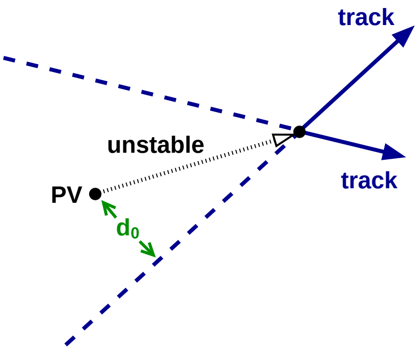


Figure 2.11: Definition of the track impact parameter.

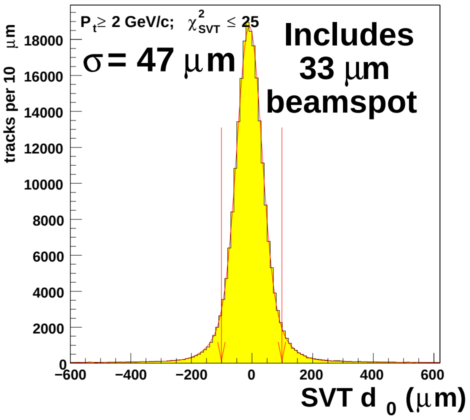


Figure 2.12: SVT impact parameter resolution.

make use of several detector components, namely the calorimeters the COT and the muon system. This is illustrated in fig. 2.10. It is worth noting that this trigger system works nearly dead-time-less. This is achieved by storing events accepted by L1 in buffers allowing L2 to make its decisions asynchronously. The third trigger level (L3) is implemented in software running on a PC farm. The software environment in L3 is the same as in offline analysis. This allows to base the L3 decisions on event variables reconstructed with offline quality.

The Two Track Trigger.—The name *Two Track Trigger* (TTT) denotes a trigger path designed to trigger on decays of long lived particles into hadronic final states. There is a strong correlation between the lifetime of the decaying particle and the impact parameters of the decay product tracks. The track impact parameter d_0 is defined as the distance to the primary interaction point at the point of closest approach in the x-y-plane. This is illustrated in fig. 2.11. In order to make use of this correlation it is necessary to first reconstruct tracks in the r-ϕ-plane. Since the trigger has to deal with a high event rate of $\sim$ 5 MHz, the track reconstruction is implemented in hardware. First, the *eXtremely Fast Track Finder* (XFT)[12] finds tracks in the COT in L1. The resulting list of XFT tracks is then passed to the *Silicon Vertex Tracker* (SVT)[13] which is part of L2. The SVT adds silicon hits to the XFT tracks by employing a sophisticated pattern matching algorithm. The resulting impact parameter resolution is close to the offline resolution as shown in fig. 2.12. In order to pass the TTT, an event is required to have the following properties.

- at least two oppositely charged XFT tracks with $p_t > 2\,\mathrm{GeV/c}$ and $\Delta\phi^{1,2} < 135°$ in L1

- at least two SVT tracks with $p_t > 2\,\mathrm{GeV/c}$, $100\,\mu\mathrm{m} < |d_0| < 1\,\mathrm{mm}$ and $\chi^2_{\mathrm{SVT}} < 25$ in L2
- match of the SVT tracks to COT tracks and confirmation of p_t and impact parameter requirements with the additional requirements of $2^\circ < \Delta\phi^{1,2} < 90^\circ$ and the decay length in the r-ϕ-plane projected on p_t ($\mathrm{L_{xy}}$) greater than 200 μm in L3.

While the TTT was especially designed to trigger on hadronic B meson decays such as $B \to \pi^+\pi^-$ it also enriches samples of hadronically decaying charmed mesons. Actually, this led to the first publication of the CDF 2 collaboration on the mass difference m(D_s^+) - m(D^+)[14]. All results presented in this thesis are based on the hadronic dataset collected by the TTT.

Chapter 3

The Reconstruction of Charged Particle Tracks

Introduction — Track Finding: track parameterisation; tracking in the COT; silicon hit reconstruction; outside-in silicon tracking; silicon stand-alone tracking; inside-out tracking; summary — Track Fitting: overview; the kalman-fitter — Detector Modelling and Material Descriptions: introduction; complete geometry description; active detector component model; passive material model; summary

3.1 Introduction

In this chapter the term “reconstruction” refers to the task performed by various algorithms operating on data that has been collected by the CDF 2 experiment and stored on digital mass storage. The algorithms are implemented in software using the C++ programming language[15]. Without sufficiently stable and efficient offline software, physics results could not be obtained in a reasonable time. Here “efficiency” refers to both, the measure of the ability to find what is there, and a good computing performance in terms of memory and time consumption. These two efficiency requirements often conflict and are hard to meet at the same time. This is especially true for the combinatorial problem of charged particle track finding, an essential prerequisite of any exclusive analysis. Furthermore, high accuracy and purity of the reconstruction algorithms can also conflict with good computing efficiency. It is non-trivial, but nevertheless possible, to optimise the software with respect to all these requirements. Some aspects of such an optimisation will be illustrated by discussing the implementation of a highly accurate and yet fast detector model in section 3.4.

3.2 Track Finding

Track Parameterisation.—Due to the magnetic field charged particle trajectories are bent to helices. The helix orientation is such that the r-ϕ-projection is a circle and the axis is parallel to the z-axis. For a given orientation five additional parameters are needed to fully specify a helix. Two different choices of parameters are used in CDF 2 tracking: perigee parameters and local parameters. For both choices the three parameters specifying the circle in r-ϕ are illustrated in fig. 3.1. The perigee is the point of closest approach to the origin in the r-ϕ-plane. This renders the perigee parameterisation the most suitable for physics analysis. The five perigee parameters are:

- The curvature $C = \frac{1}{2R}$ where R is the radius of the circle in the r-ϕ-plane.
- The impact parameter d_0. Its magnitude is the two dimensional distance to the origin. The sign is chosen such that $\vec{p}_t$ $\vec{d_0}$ and $\vec{e}_z$ form a right handed system. In the case illustrated in fig. 3.1 d_0 is negative.
- The direction of $\vec{p}_t$ at the perigee ϕ_0.
- The z-position of the perigee z_0.

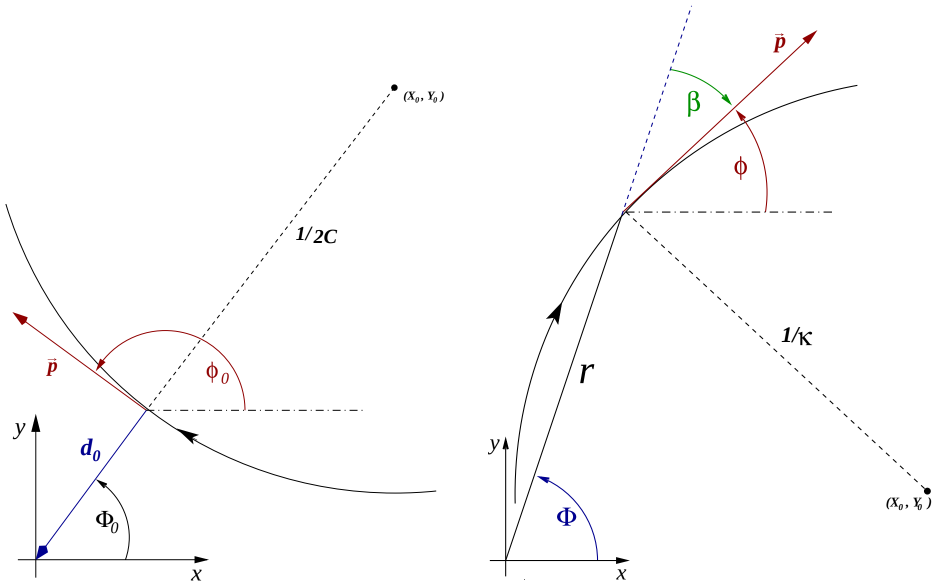


Figure 3.1: The parameterisation of charged particle tracks in the r-ϕ-plane. Perigee parameters (left) and local parameters (right)

- A measure of the momentum z-component $\cot\theta = dz/ds$, where s is the path length in r-ϕ.

Parameterising tracks close to the primary interaction point is optimal for analysis. But it has disadvantages in track finding and fitting, especially when using a progressive fit method. The reason for this is that detector components in the tracking volume do not measure perigee parameters. They all measure local positions relative to a known reference frame. This is the distance to a wire in the COT and the distance to the centre of a silicon sensor in the SVX II and ISL. Choosing helix parameters that reflect this simplifies and speeds up fitter implementations. The fast Kalman-fitter implementation used in most silicon track finding algorithms and for refitting uses local parameters. They are called "local" because they are a function of r as illustrated in 3.1. The five local parameters are:

- The curvature $\kappa = 1/R$ where R is the radius of the circle in the r-ϕ-plane.
- The azimuth angle Φ.
- The angular difference of the momentum direction an the azimuth, β.
- The z-position z
- The polar angle at the point of parameterisation θ.

These parameters are only used internally. When track reconstruction or refitting is finished they are converted into perigee parameters for analysis. Two important terms can now be defined more precisely: Track finding, or pattern recognition, means finding sets of detector measurements (hits) that are compatible with the hypothesis that they form a helix. Track fitting is the determination of the helix parameters and the associated error matrix at a given space point. Most of the time this space point will be the perigee.

Tracking in the COT.—Charged particle track finding at CDF 2 starts in the COT. This has pure geometrical reasons. Since the number of tracks is constant in ϕ the track density in $r\phi$ is lower at larger radii. Due to the resulting lower hit density the assignment of hits to a track hypothesis is much more pure than in the inner detector. The tracking in the COT happens in several steps:

1. r-ϕ pattern recognition in the axial layers
2. two-dimensional track fit in r-ϕ
3. purification of axial hit content

4. r-z pattern recognition in the stereo layers
5. two-dimensional track fit in r-z
6. purification of stereo hit content
7. full three-dimensional track fit

There are two algorithms implementing these steps, namely the segment linking and the histogram linking algorithm[16]. The segment linking algorithm finds independent track segments in the COT super-layers. The segments are then linked to form a track. The second algorithm works in a less obvious way. In the r-ϕ phase of pattern recognition hits belonging to a track are identified by clustering in curvature space. This works as follows. For a given pair of hits on can compute the curvature $1/2R$ of a circle in r-ϕ using the beam-spot as the third point. The curvatures are filled in a histogram. Real tracks lead to peaks in this histogram, thereby allowing to identify the correct hits. Obviously, this approach introduces a bias toward the beam-line. The histogram linking thus tends to be less efficient for tracks with large impact parameters, see fig. 3.2. On the other hand the histogram algorithm performs better in high occupancy events[17]. The efficiencies of both algorithms on single μ and $t\bar{t}$ Monte Carlo are summarised in table 3.1.

Algorithm	single μ eff. [%]	$t\bar{t}$ eff. [%]
Segment Linking	97 ± 1.4	85.4 ± 1.1
Histogram Linking	98 ± 1.4	95.8 ± 1.1

Table 3.1: Efficiency of the COT tracking algorithms on single μ and $t\bar{t}$ Monte Carlo.

Efficiency is defined as the ratio of found over findable tracks. A track is considered findable under the following conditions:

- The track originates inside the beampipe
- It passes through all COT layers, i.e $|z| < 150$ cm at $r = 140$ cm.
- It has more than 20 hits in total.
- The transverse momentum is greater than 0.5 GeV

The momentum requirement of $p_t > 0.5$ GeV is rather loose. As can be seen in fig. 3.3 the efficiency drops rapidly for tracks with $p_t < 1.0$ GeV. These low momentum tracks

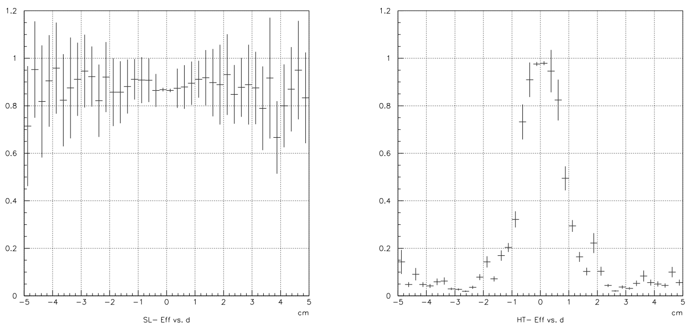


Figure 3.2: COT tracking efficiency vs. impact parameter on $t\bar{t}$ Monte Carlo. Segment linking (left) and histogram linking (right).

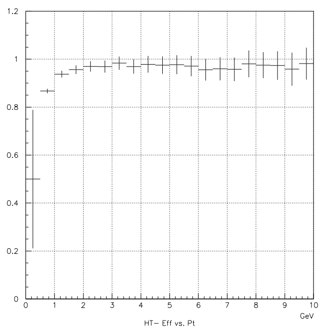


Figure 3.3: Histogram linking efficiency vs. p_t for $t\bar{t}$ Monte Carlo.

contribute most to the inefficiency. For tracks with $p_t > 1.0$ GeV the combined COT tracking efficiency is well above 98% on $t\bar{t}$ Monte Carlo. Note that $t\bar{t}$ events are the worst case scenario with respect to occupancy. Since, depending on the circumstances, the two algorithms differ in their efficiency they are run both. Each is provided with the full set of COT hits. This results in many duplicate tracks, i.e. tracks found by both algorithms. The duplicates are identified and removed in the last step of COT pattern recognition. After duplicate removal the duplicate fraction is below 0.9%. The gain in overall track finding efficiency is well worth this additional investment of CPU time. In the L3 trigger timing is more important and only XFT tracks originating from the primary interaction point have to be confirmed. Thus only the histogram linking algorithm is run in L3. While the space point resolution is dominated by measurements in the silicon detector the COT provides a much better momentum resolution. With $\sigma_{p_t}/p_t^2 \sim 0.15\,\%\ \mathrm{GeV}^{-1}$ the momentum resolution supersedes the design specification[10].

Silicon Hit Reconstruction.—Before the silicon pattern recognition the silicon hits themselves have to be reconstructed. To this end the silicon readout channels are scanned for strips with a charge above a certain noise threshold. Consecutive strips between two charge minima are considered to belong to the same particle interaction. Such strips are said to form a cluster. The hit position, i.e. the assumed intersection point of the particle track with the sensor, is then determined by the charge-weighted mean of the strip positions.

The resolution of this local position measurement depends on the the strip pitch and the number of strips in the cluster. This number shows a large variation. However, only three classes are considered in the determination of the resolution: clusters with one strip, two strips or more than two strips. The fraction of hits in the classes in the different layers of the silicon detector is shown in table 3.2. The resulting resolutions

		fraction of hits [%]					
		axial			stereo		
detector	layer	1 strip	2 strips	>2 strips	1 strip	2 strips	>2 strips
SVX II	1	7.7	22.8	69.5	17.0	25.0	58.0
	2	8.8	25.7	65.5	13.7	23.7	62.5
	3	11.6	29.6	58.8	4.0	19.6	76.4
	4	7.5	25.3	67.2	13.9	24.5	61.6
	5	12.9	24.5	61.6	12.9	30.9	56.2
ISL	6	9.2	23.1	67.7	12.5	25.3	62.2
	7	14.5	21.6	63.9	9.1	18.1	72.8

Table 3.2: Number of strips per cluster in the different silicon detector layers.

of the local position measurements are listed in table 3.3. Due to the small errors of the position measurements in the silicon detector, the position measurement close to the primary interaction point is dominated by the innermost hits on the tracks. This imposes a high purity requirement on the tracking algorithms: a wrong hit on a track hurts much more than a missed hit, especially in the innermost layers where the hit density is high. Many implementation details of the tracking algorithms are owed to this.

Together with the known position and orientation of the silicon sensor the locally one-dimensional measurements provide a two-dimensional measurement in global coordinates. This is why they are known as *2D hits*. The 2D hits are used in two different ways. Firstly, they are used directly as two-dimensional measurements in the silicon pattern recognition algorithms. Secondly, they can be combined with a second 2D hit on the other side of the sensor. The latter yields a locally two-dimensional and globally three-dimensional measurement. Hence the name *3D hits*. As already mentioned in section 2.2 these combination are most useful in the SAS layers. Especially in the *silicon stand-alone* tracking described below the SAS 3D hits play a crucial role.

Outside-in Silicon Tracking.—In the silicon detector the hit density is much higher than in the COT. This makes it desirable to employ all available information that can help in the pattern recognition. The very pure and nearly duplicate free list of tracks produced by the COT tracking algorithms provide such information. Consequently, the first step in silicon pattern recognition is to pursue COT tracks into the inner detector and add silicon hits. This procedure is called *outside-in tracking* (OI). As in the case of the COT tracking the OI tracking is performed in separate $r\phi$ and rz

		hit resolution [μm]					
		1 strip		2 strips		>2 strips	
detector	layer	axial	stereo	axial	stereo	axial	stereo
SVX II	1	13	31	11	26	23	54
	2	13	27	11	23	24	48
	3	13	13	11	11	23	23
	4	13	31	11	26	23	54
	5	14	14	12	12	25	25
ISL	6	24	24	21	21	43	43
	7	24	24	21	21	43	43

Table 3.3: Spatial resolution of silicon hits depending on the layer and the number of strips in the cluster.

phases. And again there are two algorithms implementing this procedure: one based on a generic progressive fitter[18] and one based on an extremely fast Kalman-Fitter implementation described in section 3.3[19]. The two algorithms were developed independently and evaluated against each other in the process. This resulted in two highly efficient and pure implementations. Both algorithms are run in offline production. Unlike in the case of COT tracking they do not run both on the full hit set. The second algorithm only uses silicon hits not used by the first one, thereby reducing the cost in terms of CPU time. The Kalman-fitter based algorithm is run second. Since it is slightly more efficient it can sometimes recover tracks missed by the other algorithm.

The working principle and efficiency of the two algorithms are almost identical. Thus it will be sufficient to have a closer look at only one. We choose the Kalman-Fitter based algorithm. In this algorithm there are three stages of pattern recognition. In the first phase only axial 2D hits are considered. In the second SAS stereo hits are added if they can be combined to a 3D hit with an axial hit found in the first phase. In the third and last phase 90° z-hits are added. While a track is considered a valid OI track only if it has picked up at least three axial hits, z-hits are not required. In principle one could do the pattern recognition for all kinds of hits in one go. This actually works quite well with a perfect detector on Monte Carlo. In a realistic scenario with a misaligned detector however, separating the pattern recognition phases for axial SAS and 90° z-hits has proven to be much more robust. The basic algorithm is the same in all three phases: starting from the COT inner wall the track is propagated into the detector until it intersects a silicon sensor. If there is more than one hit compatible with the track the search path is branched accordingly. To account for possible detector inefficiencies there is always one additional branch in which no hit is added in that layer. The branching can lead to many hypothesis. This requires an excellent timing performance of the fitter and the intersection finding code. The procedure is repeated until the innermost layer is reached. But not all branches are pursued that far. In each measurement layer the branches are checked against certain quality criteria like, e.g., minimum fraction of found over expected hits. Branches that do not match the criteria are eliminated. If more than one hypothesis survives until the end the best one has to be chosen. This is done using quality criteria like number of hits and goodness of fit. A more detailed description of the algorithm and its fine tuning is provided in [19].

There are many sensible definitions of efficiency. All have in common that the denominator is defined by the number of findable particle tracks. For the evaluation of the OI tracking a track is considered findable if it was reconstructed in the COT with more than 70 hits the particle left more than six hits in the silicon and its transverse momentum is greater than 0.5 GeV. This still leaves us with many possibilities for the numerator. Obviously, a track counting for the numerator has to be found at all. A track is considered found if it can be uniquely matched to a Monte Carlo particle via its silicon hit content. Using this minimal definition the OI efficiency is as high as 98.2% in the worst case scenario of $t\bar{t}$ Monte Carlo with merged-in minimum bias events. One can impose more stringent requirements for the numerator like requiring some or all track parameters to be well reconstructed. An elaborate study of the OI efficiency using various numerator definitions and Monte Carlo samples can be found in [19].

While the transverse momentum resolution is dominated by the COT the accuracy of the position estimate close to the primary interaction point is dominated by the measurements in the silicon. The tracks found by the OI tracking algorithms combine these two advantages. They inherit the excellent momentum resolution from the COT track and provide an impact parameter and z-position resolution of $\delta d_0 \sim 10\,\mu$m and $\delta z_0 \sim 50\,\mu$m, respectively.

Silicon Stand-alone Tracking.—The OI tracking can not exploit the full potential of the silicon detector. It inherits the η-dependence of the reconstruction efficiency from the COT tracking: tracks are only found up to $|\eta| \sim 1$. The purpose of the *silicon stand-alone tracking* (SiSa) is to recover tracks in the forward region $1 < |\eta| < 2$ and tracks in the central region that have too low momentum to traverse the entire COT.

The silicon hit search in the OI tracking is based on well established COT tracks that are unlikely to be fakes. In the case of SiSa tracking we are not in such a comfortable position. The helix hypothesises seeding the hit search have to be formed from combinations of silicon hits alone. This introduces a much higher risk of creating fake tracks. Furthermore, the number of possible track seeds can become very large, creating a combinatorial problem. It is hence necessary to use all a priori available information to reject fake seeds as soon as possible. This requirement determines the design of the reconstruction algorithm.

In order to form a track seed, five helix parameters have to be computed from a combination of silicon hits. Three axial measurements are needed to determine the r-ϕ-parameters C d_0 and ϕ_0. Fixing $\cot(\theta)$ and z_0 requires two additional z measurements. In principle one could build every combination of two z-hits and three axial hits that constitutes a reasonable helix, i.e. a helix describing a track originating near the primary interaction point. Next one would try to attach more hits to all the seeds and finally perform a global optimisation. This, however, is simply not feasible. Already the number of reasonable combinations of three axial hits is far too high to carry out this procedure in a reasonable amount of time. In addition, a completely independent consideration of z-hits and axial hits leads to many fake seeds. Consequently, the SiSa algorithm uses a different approach. The combinatorial problem is reduced by using only two axial hits in different layers and the beam position in the x-y-plane to determine the seed r-ϕ parameters. It should be noted that the beam position does not enter as a measurement. It is only used to form track seeds. The fact that the two hits in an SAS 3D hit confirm each other is used to reduce the fake rate: each of the two axial seed hits is required to be a constituent of an SAS 3D hit. Due to the ambiguities in the 90° layers they are not suitable for this purpose. Thus only two SVX II layers and the ISL can be used for seed construction. This can reduce the track finding efficiency in a non-perfect detector. But given the high risk to produce fake tracks, purity has precedence here. To further reduce the fake rate the track seeds are required to be compatible with the z-position of one of the primary interactions in the event. This reduces the number of seeds significantly and thereby reduces the time spent in the SiSa tracking considerably. Obviously, the z-vertices have to be found before the silicon tracking is run. This is achieved by a histograming algorithm based on the COT z_0 parameters and the seeds themselves[20].

Once the list of track candidates is established it is sorted with respect to two quality criteria. The seeds built from 3D hits in the SVX II are preferred over seeds with one or both hits in the ISL. The better z resolution in the SVX II allows for narrower search roads, thereby reducing combinatorics and increasing purity. The second criterion is the transverse momentum, high momentum tracks being preferred over low momentum tracks. The tracks seeds are processed in the resulting order of preference. The hit search algorithm is very similar to the OI algorithm with the COT tracks replaced by the track seeds. There are additional tweaks that deal with the impurities in the seed list and subtle systematic effects. Once a track candidate is accepted, *all* hits on the track are considered correctly used and all seeds sharing a hit with the track are eliminated. In order to minimise the chance of losing real tracks, the seed processing is performed several times. In the first round very strong quality requirements are imposed on the tracks, producing a high purity sample. Then, round by round, the remaining seeds are processed with weaker and weaker quality requirements. This procedure ensures good purity and efficiency within a reasonable processing time. A detailed description of the SiSa algorithm and its performance under various circumstances can be found in [19].

Inside-out Tracking.—While the SiSa tracking recovers forward and low momentum tracks in the silicon, there are yet unused COT hits from particles that did not traverse the entire COT for one reason or the other. It is the purpose of the *inside-out tracking* (IO) to make use of this information.

The idea is to reverse the concept of OI tracking. The fast Kalman-fitter comes into play again here. It is capable to deliver the best estimate of local track parameters at every space point in a very efficient way. Using the fitter, SiSa tracks are extrapolated outward to the inner wall of the COT. The parameter estimate at this position is then used as a seed for a outward hit search in the COT. After the COT hits compatible with the seed are identified they are fitted without the silicon hits, yielding a COT-only parameter estimate at the COT inner wall. This "COT track" together with the silicon hits of the SiSa seed is then treated like an OI track: using the Kalman-fitter once more, an inward fit is performed in order to determine the perigee parameters

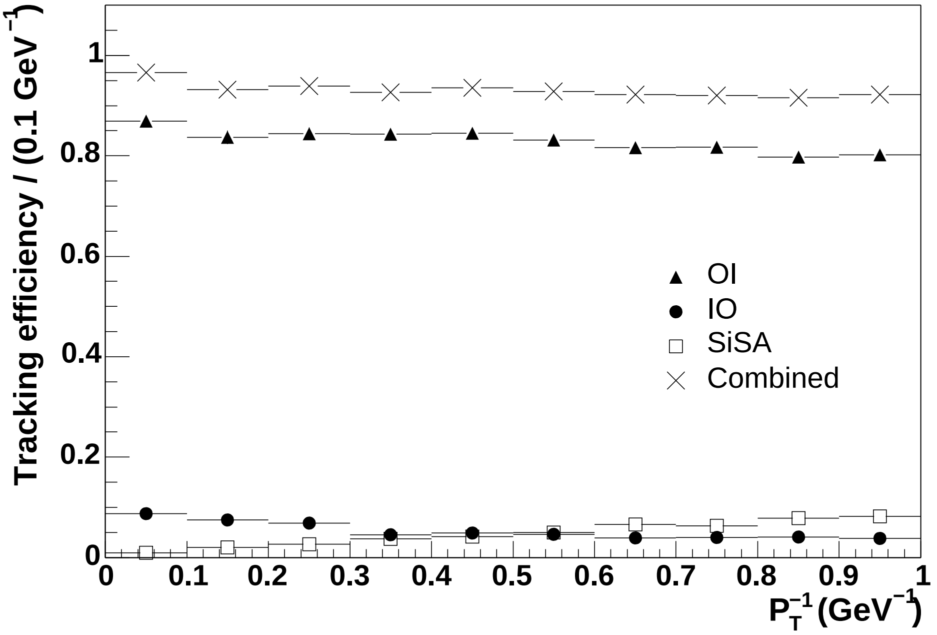


Figure 3.4: Efficiency of the silicon tracking algorithms versus p_t^{-1}.

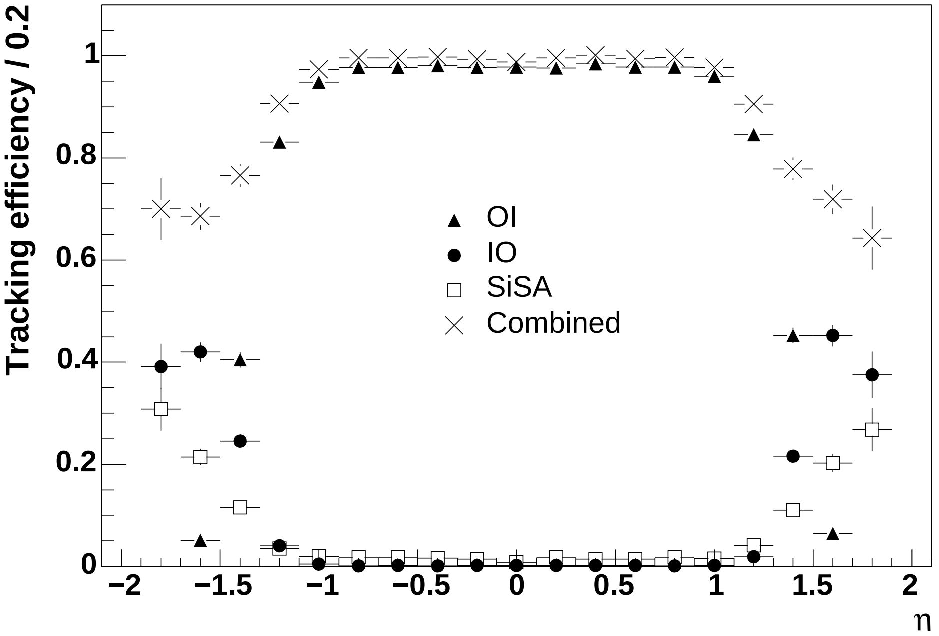


Figure 3.5: Efficiency of the silicon tracking algorithms versus η.

and their covariance matrix.

The IO algorithm greatly improves the forward and low momentum tracks in two ways. Firstly, the p_t resolution is improved considerably by adding COT hits to the SiSa tracks. Secondly, the fake rate is decreased by discarding SiSa tracks which can not be confirmed in the IO tracking[21].

Summary.—By running the four tracking algorithms in the order COT-OI-SiSa-IO CDF 2 tracking makes best use of all available information while at the same time maintaining a high purity. A fast Kalman-fitter implementation plays a crucial role in connecting the COT to the silicon measurements and in the silicon pattern recognition itself. The overall tracking efficiency as a function of p_t and η is summarised in figures 3.4. and 3.5. The two parameters most relevant for physics analysis, C and d_0, are reconstructed with a resolution of $\sigma_{p_t}/p_t^2 \sim 0.15\,\%\ \mathrm{GeV}^{-1}$ and $\delta d_0 \sim 10\,\mu\mathrm{m}$, respectively.

3.3 Track Fitting

Overview.—Employing statistical methods in order to obtain the best estimate of track parameters and their covariance matrix for a given set of measurement points is called *track fitting.* At CDF 2 the statistical method used in track fitting is always an equivalent of χ^2 minimisation method. It is important that the track fit provides a measure of fit quality in addition to the parameter estimate since it is otherwise impossible to base any decision on the fit result.

In the remainder of this chapter we are specifically concerned with track fitting in the CDF 2 silicon detector. The measurements entering the track fits are either position measurements in the SVX II and ISL or the result of a COT track fit, as in the case of OI and IO tracks.

Contrary to what the above definition might suggest, track fitting is also performed *during* the pattern recognition. In this scenario the set of measurements is not yet complete and the track fit quality criterion helps to decide whether a hit should be added to a track. Especially in the SiSa tracking a very large number of hypothesis has to be tested. This makes a very fast fitter implementation a necessity.

The second scenario of track fitting is track refitting, i.e. the re-determination of track parameters and their covariance matrix without changing the hit content of the track. There are several reasons why it can become necessary to refit tracks. The most common reason is a change in particle hypothesis. For a given momentum, the energy a particle loses due to its interaction with the detector material depends on its mass. The track fit has to take this energy loss into account and hence the fit result depends on the mass. The mass is not directly measured by the tracking detectors. Since the majority of particles produced in the collision are pions the default mass hypothesis is m_π. But if one reconstructs e.g. $D^0 \to K^- \pi^+$ it is necessary to refit the negatively charged track with a kaon hypothesis in order to get the D^0 mass right. Another reason that makes it necessary to refit track is a change in alignment or detector understanding like an improved understanding of the silicon hit resolution. While significant improvements of the detector understanding become rare over time, the alignment is always subject to change. The relative position and orientation of the silicon sensors and the location of the silicon detector with respect to CDF 2 coordinate system change with time. In most cases these changes are very small. But the high resolution of the local position measurements in the silicon detector make it necessary to use the best available alignment for track refitting. For a track-based analysis this means that all tracks should be refitted, even for the default m_π mass hypothesis. An additional, more unpleasant, reason for refitting tracks is the discovery of a bug in the fitter implementation. Again, all this makes it desirable to perform the track fits in as short a time as possible.

Given the manifold scenarios of track fitting, a versatile track fitter is needed that performs equally well and fast in all of them. The Kalman-fitter described in the following section is ideally suited for this purpose.

The Kalman-Fitter.—A Kalman-fitter is a progressive fitter based on the principle of a Kalman-filter. A progressive fitter is a fitter that does not explicitly refer to previously applied measurements or corrections when a new measurement or correction is incorporated into the fit. All measurements and corrections are applied in a step-by-step manner. In each step the starting point is the current status of the fit which implicitly contains all the information from previous steps. Progressive fitters can be implemented much more efficiently than non-progressive fitters. This is very similar to the underlying principle of the Kalman-filter. The original purpose of the Kalman-filter was to provide a signal filtering method in electrical engineering. It was first described by R. E. Kalman in 1960[22].

The basic idea is the following. Suppose we want to measure a time dependent quantity $x(t)$. For each time t there is a true value $x_{\mathrm{true}}(t)$. What ever the means of measuring x are, they can never be perfectly accurate. Consequently each measurement $x(t)$ has an associated error $\sigma(t)$. The error quantifies our lack of information about the true value $x_{\mathrm{true}}(t)$. Now suppose a certain time span Δt passes before the next measurement. Just *before* the next measurement is performed we know less about $x_{\mathrm{true}}(t + \Delta t)$ than we knew about $x_{\mathrm{true}}(t)$ *after* the measurement at time t. This is because x depends on time and we had to make a projection into the future. The procedure used for this projection together with the underlying assumptions is called the *transport model*. Our non-perfect knowledge of $x_{\mathrm{true}}(t)$ and uncertainties in the transport model both contribute to the increased uncertainty at the time $t+\Delta t$. After the new measurement we have to update our overall estimate using the projection and the new measurement. Provided the errors are reasonable, the best way to combine these two informations is an error-weighted mean. This combination always has a smaller uncertainty than the single estimates entering it. This simply expresses the fact that it contains more information. The whole procedure is nicely illustrated in [19]. Note that it is not necessary to keep all the single measurements to carry out this procedure. All information collected at the time t is implicitly contained in $x(t)$ and $\sigma(t)$. It can be proven that this procedure is completely equivalent to a least square fit to all measurements. To perform the latter, however, it would be necessary to record all measurements.

The Kalman-fitter for particle tracks works very similar. Nevertheless there are some important differences. First of all there is more than one parameter to be determined. Instead we have a vector $\vec{x}$ of the five track parameters. Since there are considerable correlations between the parameters, the uncertainties have to be described by a five dimensional covariance matrix. Next the measurements are not ordered by time but rather by their distance from the z-axis, r, following a particles path from the beam outward. But these two differences are not fundamental. After all they just represent a change in the dimensionality of parameter space and in the choice of the ordering parameter. There is one fundamental difference, however. A Kalman-filter requires the transport model to be linear. The transport procedure in

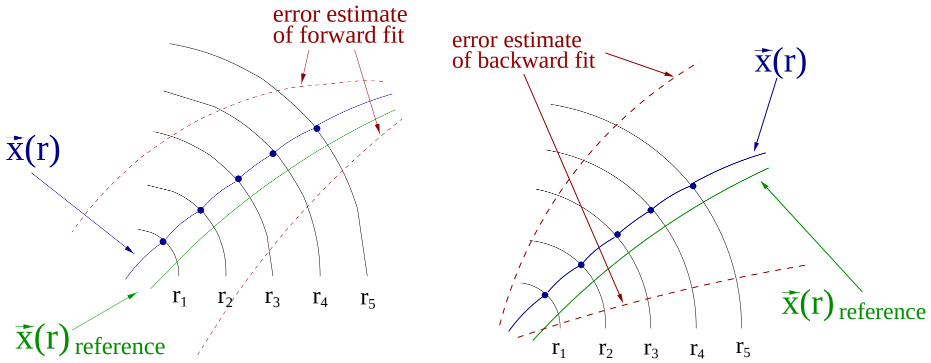


Figure 3.6: Illustration of forward vs. backward Kalman-fit.

track fitting involves the computation of the local parameters at a radius r_2 from the given parameters at another radius r_1. For curved trajectories this is a nonlinear transformation. Thus the assumption of linearity is violated in charged particle track fitting. In order to apply the Kalman-filter principle the transport model has to be linearised. This is achieved via a Taylor series expansion about a reference point in parameter space: instead of the track parameters themselves the deviation from a reference trajectory is fitted. The deviation $\Delta\vec{x}(r) = \vec{x}(r) - \vec{x}(r)_{\text{reference}}$ is the distance of the fitted track and the reference trajectory in parameter space. Terms of order $\mathcal{O}(\Delta\vec{x}(r)^2)$ or higher are dropped from the expansion. The transport model for the deviation is then linear and the Kalman-filter principle can be applied. This approach is based on the assumption that the deviation is always sufficiently small to allow for a linear approximation. For very low momentum tracks with large energy loss or multiple scattering this assumption does not hold sometimes. To account for this the reference trajectory is reset to the fit result after each fit step, ensuring $\Delta\vec{x}(r) = \vec{0}$. This is only done in track refitting and in the final fit after the pattern recognition. resetting the reference also in pattern recognition would require the recomputation of all subsequent material intersections in each search branch – a far too time consuming procedure, even in the highly efficient geometry framework described in the following sections.

Given that all the measurements are available right from the start one could ask what is the point in applying a Kalman-filter based fit algorithm. The answer is there are certain tasks that can be performed much more efficiently compared to a any non-progressive fitter, namely the ones where a decision has to be made. This occurs most frequently in the hit search stage of the silicon pattern recognition. For each intersection of a track with a silicon sensor there can be several hits compatible with the track hypothesis. It has to be decided which hits are added to the track. A non-progressive fitter would have to perform a fit to all hits, i.e. the hit currently evaluated *and* all previously added hits, for each of the hits possibly compatible with the track. A Kalman-fitter based algorithm on the other hand does not have to reconsider the previously added hits. It just incorporates the new measurement into the fit by computing a weighted mean.

A Kalman-fitter is also the more natural choice when the track parameter estimate has to computed at an arbitrary space point along the track. The result of the Kalman-fit only contains all information after all measurements and corrections have been incorporated, i.e. after the last fit step. To which space point the last fit step corresponds depends on the direction of the fit and whether all measurement are used. If progressive fit is performed outward from the beampipe to the COT inner wall it is called a *forward fit*. It is forward in the sense that it follows the flight direction of a particle coming from the primary interaction point. In a forward fit the last fit step corresponds to the COT inner wall where it yields the best parameter estimate. A forward fit is most suitable if a silicon track is to be extrapolated outward into the COT like in the IO tracking. If the direction of the fit is reversed it is called a *backward fit*. In this case the best parameter estimate is obtained inside the beampipe. The two cases and how the parameter uncertainties evolve in each of them are illustrated in fig. 3.6. If the backward fit is not carried out completely one obtains the best parameter estimate at the space point where it stopped. This is useful when reconstructing long lived particles like K_S. A K_S can decay in the middle of the silicon detector. In this case the measurements and material corrections inward from the K_S decay vertex should be excluded from the track fit of the decay products[1]. The best parameter estimate at an arbitrary space point can be obtained by applying the following procedure. First the fit is performed in both directions and the results at each fit step are recorded. Then the best parameter estimate at a given space point is obtained by simply computing the weighted mean of the forward and backward fit results of the corresponding fit step[2].

A full-featured Kalman-filter based track fitter is implemented in the `TrackingKal` package of CDF 2 offline software[23]. This implementation aims for computing efficiency wherever possible. All calculations for the parameter transport model, parameter conversion, matrix inversions etc. were carried out analytically[19]. The results were then implemented in C++, taking care to minimise any computational overhead in the process. This extremely fast implementation is used in silicon pattern recognition algorithms and for refitting tracks in all track-based analyses at CDF 2.

[1] If the candidate represents a real K_S all measurements inward from the decay vertex can only be wrong since it is impossible that the outward bound decay products have left hits inward from the vertex.

[2] In this case the reference trajectory must not be reset. It has to be the same for both fit directions, limiting the method to “well behaved” tracks.

3.4 Detector Modelling and Material Descriptions

Introduction.—An accurate and efficient detector model is a necessary precondition of a fast fitter implementation. There are two major aspects of the detector geometry that have to be represented in the reconstruction software: the exact position of all active components and the properties of the passive material and its distribution. Physically each active detector component is always also passive material – unfortunately we can not build immaterial measurement devices. From the software representation point of view this is not relevant, however. The two sides of the silicon sensors where the measurements occur can be modelled as infinitely thin surfaces. The fitter just has to know their exact location and orientation in order to relate the local measurements to the CDF 2 coordinate system, yielding a measurement of one or more track parameters. This model of the active components has to be organised in a way that allows to find intersections of helices and sensor surfaces as quickly as possible. Otherwise the silicon pattern recognition would be slowed down considerably, no matter how fast a fitter implementation is employed. The material aspects of the sensors as well as the passive detector material can then be modelled independently. It is even possible to completely decouple the description of the passive material properties from the physical detector components. This allows to handle the passive material effects in a very efficient yet highly accurate way.

On the other hand it is necessary to maintain a detailed model of all detector components for simulation purposes. In order to ensure consistency the other, more specialised, models must be derived from this fully detailed description in such a way that changes to the geometry and material properties are propagated automatically. Independently maintained detector models for specific purposes have to be avoided. A geometry model obeying all these requirements is described in the following sections. A more detailed discussion of the technical aspects of the passive material model is given in appendices A and C.

Complete Geometry Description.— The complete CDF 2 geometry description is organised in a tree-like hierarchical structure of so-called detector nodes and volumes[24]. The detector is subdivided in its major components like the muon system, calorimetry and the silicon vertex detector. These are the nodes. The nodes are completely independent of each other. This allows detector experts to model one component without having to consider properties of other components. Internally the detector nodes are organised in volumes. The volumes correspond to smaller components like half ladders in the silicon detector. There are two kinds of volumes: logical and physical volumes. A logical volume is a branching point or leaf in the geometry tree. It only knows about its position, its orientation and the child and parent volumes in the geometry tree. For each logical volume there is an associated physical volume. The physical volume knows about its exact dimensions and material properties. The rationale behind all this is the fact that many detector components have exactly the same properties. For example the very same half ladder appears 72 times with different orientations and positions in a given layer of the SVX II. The geometry description is implemented in C++. Consequently the node and volume concepts directly translate to C++ classes. CDF 2 still uses GEANT 3[25] for detector simulation. Since GEANT 3 is implemented in FORTRAN the simulation can not access the geometry description directly. The GEANT 3 geometry is generated automatically from the C++ geometry, thereby ensuring consistency.

It is a major achievement of CDF 2 that one geometry description is used consistently in simulation and reconstruction. However, the design of this geometry description emphasises simulation over reconstruction. Here we are especially concerned with the description of the silicon detector as it is implemented in the `SiliconGeometry` package of CDF 2 offline software[26]. In track finding and track fitting we frequently face the problem of path finding: for a given helix we have to find the intersections with the silicon sensors in order to attach the measurements to the fit. A track is usually pursued through the detector inside-out or outside-in, i.e. along increasing or decreasing distance to the z-axis. The cylindrical symmetry of the real detector can help a lot in performing this task. But the geometry description does not explicitely express this symmetry. It does not 'know' about the layer structure of the silicon detector. There are interfaces that hide this fact from the user. But it does not help to brush the problem under the carpet syntactically. If not all available information is exploited, path finding will always be slow. This problem is addressed by the active component model described in the next section.

Active Detector Component Model.— In this section we will discuss a meta structure imposed upon the generic geometry description. This structure was designed to solve the path finding problem mentioned in the previous section. It is especially tailored to the needs of the fast Kalman-fitter described in section 3.3. The basic idea is to make the geometry description aware of the symmetries of the real detector, especially the layer and ϕ-wedge structure.

The meta structure is implemented by the `KalDetector` class in the `SiliconGeometry` package. Just like the generic detector description it features the concept of a geometry tree. Consequently there is a corresponding class called `KalDetectorNode`. Despite its name the semantics of this class is rather that of a combination of logical and physical volumes than a that of a node in the generic description[3]. Each `KalDetectorNode` represents a half ladder. It knows about its orientation and lo-

[3] In hindsight we would have chosen the name more wisely. This illustrates another common problem in software development. It is hard to find the proper names for classes and even local variables. One should put considerable effort in thinking about names. Doing so usually yields some important insights. This is one of the – unfortunately many – places were we failed to do so.

cation and all geometry properties. The geometry properties include the half ladder dimensions as well as the orientation of the silicon strips and the stereo angle, where applicable. A KalDetectorNode is a KalAlignmentNode by inheritance. This concept was introduced to allow for alignment studies using the fast Kalman-fitter. The KalAlignmentNodes are organised in a tree. The tree is managed by the class KalAlignmentTree. Not all KalAlignmentNodes have to be a KalDetectorNode. This allows to represent aspects like the barrel structure in the geometry description. In fact it allows to group arbitrary sets of KalAlignmentNodes (and hence KalDetectorNodes) and hook them to one space point, possibly involving rotations and translations of the whole group. The only restriction being that each node has exactly one parent node. The KalAlignmentTree takes care that the locations and orientations of all nodes are updated when a new node is introduced into the geometry tree or the position and/or orientation of an existing node is modified. The most simple geometry tree is one master node at the origin and all KalDetectorNodes representing half ladders directly attached to it. This is the default configuration. It is also virtually the only configuration currently in use. The positions and orientations of the half ladders are initialised from the generic geometry description, making sure everything is consistent. This, and the fact that the tree structure is very similar to the one implemented in the generic geometry description, is the reason why this implementation is also known as the *KAL proxy*.

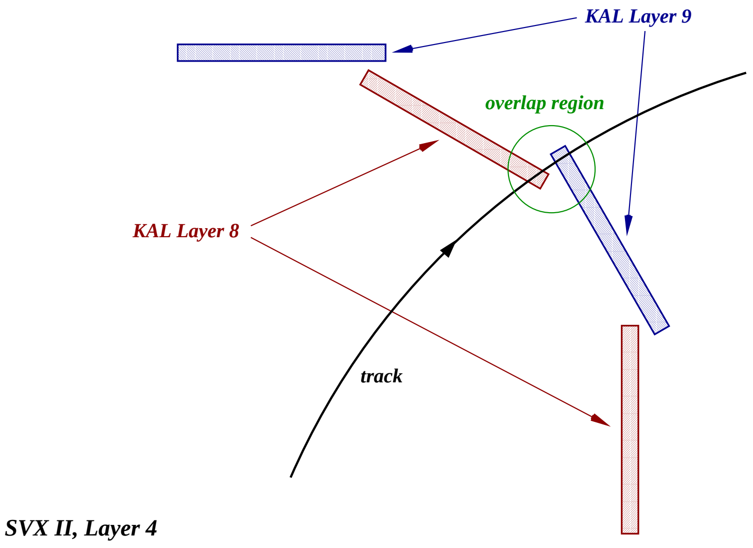


Figure 3.7: Illustration of the layer structure provided by the KAL proxy. Overlap regions are handled naturally.

The `KalDetector` knows more, however. It is not ignorant about the detectors layer and ϕ-wedge structure. After the tree of half ladders is initialised an additional structure is superimposed by invoking *`KalDetector::structureGeometry()`*. The layer structure is built first. All half ladders are sorted with respect to their minimum distance to the z-axis. The resulting sorted list is then scanned starting with the half ladders with the smallest minimum radius. The half ladders are added to a sublist constituting the current layer until a significantly different radius is encountered. In this case a new layer is started. The procedure is repeated until the end of the sorted half ladder list is reached. Note that the half ladders belonging to a layer are not required to have exactly the same minimal radius. This is because exact equality is generally not a good concept when dealing with computer numerics. The half ladders in one layer are rotated about the z-axis. This leads to small deviations from the nominal radii due to the limited accuracy of implementation of the trigonometric functions. To account for this the radii in one layer are allowed to differ at the level of 0.05%. This is large enough to account for rounding errors and sufficiently small to ensure the correct layer structure. There is one caveat, however. In the real detector the half ladders are not at their nominal positions. This misalignment can lead to radial shifts large enough to make the above procedure fail. Using an even more sloppy definition of equality does not help since it leads to mixed-up layers. Thus the layer structure has to be built before the alignment is applied.

The resulting layer structure makes it easy to access all half ladders belonging to a given layer. In addition, the layers in the KAL proxy reflect the real detector structure and not just an abstract numbering scheme. The generic geometry description only knows about the layer concept introduced in section 2.2. This has unpleasant consequences. In this generic numbering scheme the half ladders in one SVX II layer occur at two different radii and overlap in regions around $\phi = \pi/6 \cdot (n + 1/2),\ n \in \mathbb{Z}$. This is illustrated in fig. 3.4. This leads to problems in a typical situation during pattern recognition. When the track finding algorithm looks for compatible hits at all half ladders in a generic layer things can get complicated. There can be more than one compatible hit but it is not clear what this means. Two compatible hits can be mutually exclusive or they can actually belong both to the track, depending on whether the situation occurs due to the ϕ overlap or not. In ISL layer 6 the problem is even worse. Layer 6 consists of three barrels with two different mean radii which overlap in z, leading to up to four-fold overlap in ϕ. In the generic geometry, awkward workarounds are necessary to deal with this problem. In the KAL proxy geometry, on the other hand, all overlap regions are handled naturally.

After the layer structure has been built the layers are subdivided in ϕ-wedges by a very similar procedure. Finally the half ladders in the ϕ-wedges are organised into z-ranges. The layer, ϕ-wedge and z-range structures are all sorted lists and can be searched very efficiently using a binary search algorithm. The `KalDetector` class provides convenient and efficient interfaces to find all half ladders reaching inside a volume defined by r, ϕ, z and the uncertainties $\delta\phi$ and δz. The meta structure and interfaces provided by the KAL proxy emphasise the reconstruction use case of the geometry description. It is most suitable for track finding and track fitting.

Passive Material Model.— So far we were only concerned with active detector components, i.e. silicon sensors. But the silicon tracking volume at CDF 2 also contains a considerable amount of passive material. The correct treatment of this material is crucial for the correct determination of track parameters and their errors by the track fitting algorithms, especially for low momentum tracks[27]. Consequently, this is also true for the resolution and accuracy of track momenta, and hence of reconstructed particle masses. Furthermore, there are timing constraints since all tracks have to be refitted for each particle hypothesis in a track based analysis. The task then is to handle the passive material effects as *accurate and fast* as possible.

The main effects from passive material are energy loss (dE/dx) and multiple scattering. Energy loss of minimum ionising particles in solids can be described by the Bethe-Bloch formula[28]:

$$dE/dx = C_{dE/dx} \cdot \frac{1}{\beta^2} \left(\ln \left(\frac{2 m_e c^2 \beta^2 \gamma^2}{I_0} \right) - \beta^2 \right) \tag{1}$$

Were β, γ and m_e have their usual meaning, dx is the pathlength in the material and $C_{dE/dx}$ and I_0 are material properties. The energy loss directly affects the curvature of a charged particles trajectory in the magnetic field. Due to correlations, this also has an impact on the other track parameters when they are propagated between different radii.

Multiple scattering is governed by the integrated radiation length in the material:

$$dRL = dx/X_0, \; [X_0] = \frac{\text{cm}}{\text{radiation length}} \tag{2}$$

Where dx is again the pathlength in the material and X_0 is a material property. Since its mean effect vanishes multiple scattering does not directly affect track parameters but it increases the uncertainty of the parameters.

Inspection of equations (1) and (2) yields the following list of ingredients needed to handle the passive material effects:

- material properties
  - $C_{dE/dx}$, the energy loss constant
  - I_0, the mean excitation potential
  - X_0, specific radiation length

- a particle hypothesis (a mass is needed to compute β and γ from the momentum p)
- dx, the pathlength in the detector components traversed by the trajectory

Obtaining the material properties is trivial once the problem of finding the traversed detector components is solved. The silicon tracking volume contains a large number of different detector components in a variety of shapes and orientations. Finding intersection points and path lengths under these conditions is a highly non-trivial task. Only in a small number of cases the intersections of helices with geometrical volumes can be calculated analytically, namely for planes and cylinders parallel to the helix axis and for planes perpendicular to it. In all other cases one has to retreat to numerical solutions. But these are inherently slow, violating one of our requirements.

Several solutions to this problem were proposed and implemented in CDF 2 offline software. They range from a complete numerical solution that uses GEANT in reconstruction to a solution that only considers the components that can be handled analytically and accounts for the missing material by introducing 'phantom material'. In a sense using GEANT is the perfect world: it provides full consistency with the complete material description and is highly accurate. It is, however, far too slow to be used in everyday analyses. A detailed description of the various solutions and a comparison of their performance can be found in appendix B. The only fully satisfactory solution is provided by the SiliMap concept introduced in this section. The SiliMap solution aims for the high accuracy provided by GEANT while simultaneously providing fast pathfinding.

The basic idea is to trade time for memory: GEANT has to step through a given detector component each time it is hit by a track. Each time it does so the same material properties are retrieved by this time consuming process. The timing can be improved considerably by performing the GEANT stepping only once and storing the result.

If this was done on the level of detector components as they appear in the detailed geometry, nothing would be gained concerning the pathfinding problem. Fortunately, the information on the component's shape etc. is not relevant for the material integration. This enables us to organise the storage in a way that makes pathfinding trivial: a binned three dimensional map. Figure 3.8 illustrates this concept. The silicon tracking volume, i.e. $r < r_{\mathrm{COTWall}} = 40.6$ cm, $|z| < 100$ cm, is divided in bins in r, ϕ an z. While the ϕ and z bins in each r bin (map layer) are equidistant, the layer boundaries are chosen suitable for the needs of the pattern recognition and the fitter. This organisation makes pathfinding easy: once the map layer is known, the bin indices can be directly computed from the ϕ and z coordinates.

The binning resolution must be sufficiently high to reflect the often rapid changes in material properties. On the other hand, decreasing the bin size rapidly increases

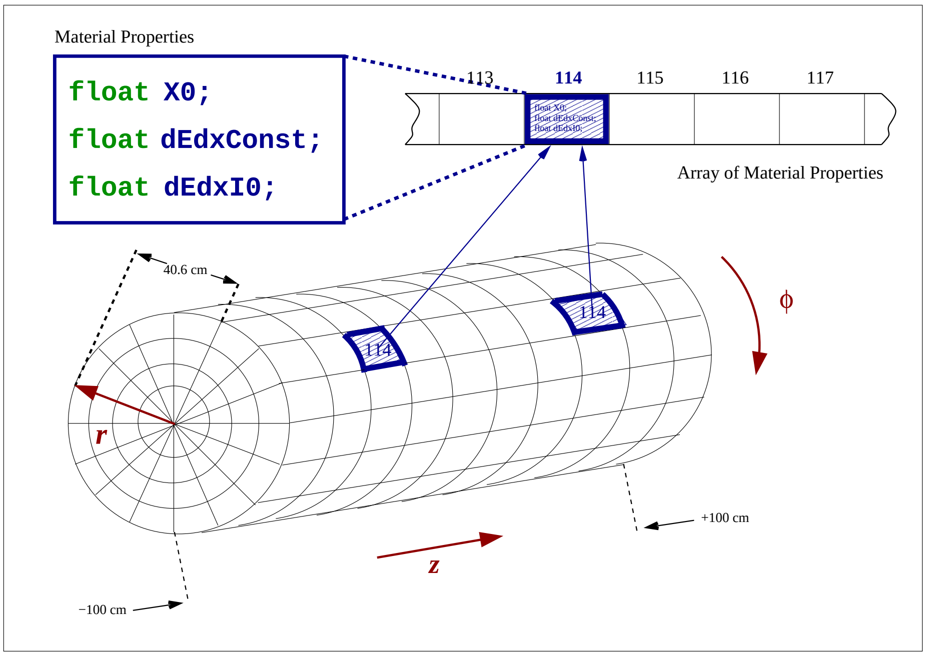


Figure 3.8: The concept of SiliMap is to produce a binned map of the passive material properties using GEANT. Each map bin holds the index of a description of the average material properties at the position of the bin. Due to the high symmetry of the silicon detector several bins have the same average material properties. In the real map the bin size is constant in $r\phi$ rather than ϕ.

the memory consumption. This makes it necessary to exploit the symmetry of the detector. Each bin is scanned at different positions with GEANT to obtain the average material properties in its volume, i.e. the scanning resolution is even higher than the binning resolution. The spatial resolution achieved by this procedure is illustrated by using SiliMap to integrate the radiation lengths along trajectories perpendicular to the z-axis. The resulting plot is shown in fig. 3.9. It nicely reflects the features of the silicon detector geometry.

The high granularity of the binning millions of trajectories which have to stepped through the detector by GEANT. To avoid the unnecessary invocation of this very time consuming process the results have to be stored persistently. This raises the question how the map should be distributed to users. These technical issues and and how they are resolved by the SiliMap implementation are discussed in detail in the appendices A and C.

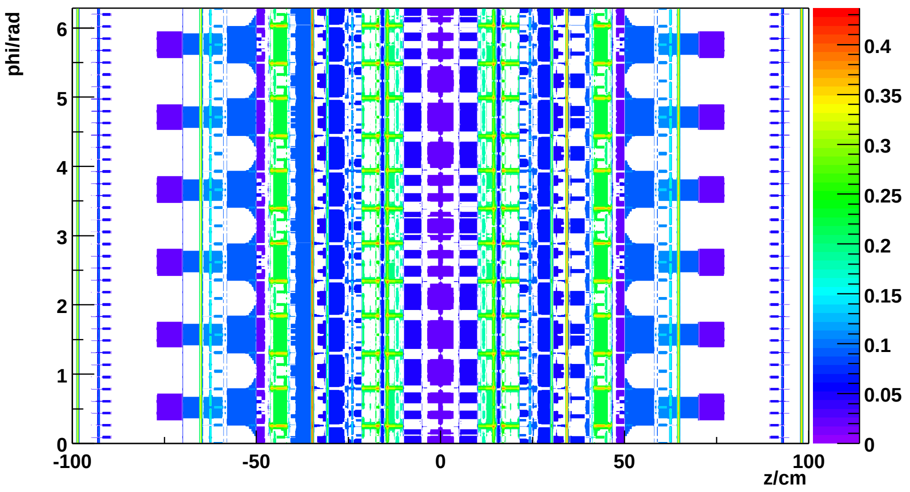


Figure 3.9: Integrated radiation lengths in the CDF 2 silicon detector according to SiliMap. The plot shows the integrated radiation lengths along trajectories perpendicular to the z-axis.

Summary.— There are numerous requirements that have to be met by a good geometry description. The most obvious was actually not discussed so far, namely the consistency of the geometry description and the real detector. We did not discuss it because it is not a software implementation issue. Nevertheless the CDF 2 collaboration has put considerable effort in ensuring this consistency and the software model is meanwhile an almost perfect description of the real detector.

Given that, we have to ensure consistency of the simulation and the reconstruction model and provide suitable and efficient interfaces for all use cases. This is achieved by using a complete geometry description as the basis of a number of derived proxies and interfaces. The KAL geometry proxy in conjunction with the SiliMap model of the passive material provide a highly accurate and efficiently interfaced geometry model for track finding and fitting.

Chapter 4

Correlation Analysis

Outline of the Analysis — Reconstruction of charmed Mesons: track selection criteria; decay reconstruction; D meson mass spectra — Correlation Analysis: concept of the analysis; toy Monte Carlo study — Evidence for correlated Charm Production; evidence for correlated $D^0\bar{D}^0$ production; angular distributions

4.1 Outline of the Analysis

The goal of the analysis is to establish evidence for the correlated production of charmed meson pairs for the first time at a hadron collider. To this end it is first necessary to extract the D meson signals from the huge dataset collected by the Two Track Trigger. Three D meson decay modes are considered, namely $D^0 \to K^-\pi^+$, $D^+ \to K^-\pi^+\pi^+$ and $D^{*+} \to K^-\pi^+\pi_s^+$. The signals are established by combining tracks and fitting them to a common vertex. Combinations with invariant masses inside a certain window around the known meson masses[1] are then considered candidates. Possible pairs of correlated mesons are constructed by combining the four-momenta of two mesons, again by applying a vertex fit. This procedure was implemented in a CDF 2 offline analysis job[29][30] using the C++ programming language. The analysis libraries and executable were compiled and linked against release 5.3.4 of CDF 2 offline software[31]. The analysis was run on the Two Track Trigger dataset as of November 28th 2004. This dataset has a total size of 13,265.407 GB, corresponding to 270,103,586 events and $\sim 481\text{pb}^{-1}$ of integrated luminosity. All four-momenta and additionally required data resulting from this procedure were written to disk for later analysis using the ROOT data analysis framework[32]. The final analysis was done using ROOT 4.00/04 compiled with gcc 3.2 on a SuSE Linux 8.1 system running Linux kernel 2.4.27 on an AMD Athlon XP 2000+ CPU. The search for correlated production was performed in this final step of the analysis. In order to proof the concept of the correlation analysis method a toy Monte Carlo study was conducted. The various steps of the analysis will be described in detail in the following sections.

4.2 Reconstruction of charmed Mesons

Track Selection Criteria.— The analysis at hand is completely track based. No information from detector components other than the tracking system are used. This makes it necessary to impose some quality requirements on the tracks entering the analysis. Quality here more or less directly translates to purity, i.e. fake tracks not corresponding to a real particle are to be avoided. In addition, the particle trajectories should be well behaved. Tracks with large kinks from multiple scattering are usually not well reconstructed and have a higher probability of containing wrong hits.

Several variables are correlated with track quality. First of all there is the tracking algorithm. OI tracks which are required to traverse the entire COT tend to be of higher quality than SiSa tracks that are based on silicon measurements only. Also COT tracks without silicon information are of lower quality in the sense that they do not provide a high resolution position measurement near the primary interaction point. In general high momentum tracks can be reconstructed with higher purity. But we can not afford to impose stringent momentum requirements on all tracks since we want to reconstruct $D^{*+} \to K^-\pi^+\pi_s^+$ which contain a soft pion (π_s) in the final state. The most important remaining variable is the number of hits on the track. The higher the number of hits compatible with a single track hypothesis the lower is the probability that this hypothesis is a fake track. These considerations lead to the following requirements for tracks entering the analysis:

- OI tracks with at least 15 axial and 10 stereo COT hits
- IO tracks with at least 8 axial and 3 stereo COT hits
- at minimum three axial silicon hits
- transverse momentum greater than 500 MeV

Silicon-standalone tracks are omitted because those that do not give rise to IO tracks are in general of low quality. All tracks fulfilling these requirements are refitted with K and π particle hypothesis using the fast Kalman-fitter and material model described in chapter 3. As recommended by the CDF 2 B-physics group, the magnetic field as well as the COT track covariance matrix and curvature were corrected following the procedures documented in [33] and [34]. Since L00 was not well understood at the time the data used in this analysis was processed the tracks in the dataset contain no L00 hits. In order to further improve the position resolution close to the primary interaction point a dedicated L00 pattern recognition was performed prior to the track refits by using the `L00AddAndRefit` interface[35][36][37]. No L00 hits are required, however. All successfully refitted tracks meeting the above requirements enter the next stage of reconstruction described below.

Decay Reconstruction.— The D mesons are reconstructed by building all possible track combinations compatible with decay products for a given channel and then performing a vertex fit. The implementation of this procedure makes heavy use of a toolkit for B physics analysis available in CDF 2 offline software[38]. This toolkit introduces concepts like stable and decaying particles as well as decay chains. It also allows to transparently apply various algorithms to particles and decay chains, thereby greatly facilitating exclusive analyses.

The most simple channel is the $D^0 \to K^-\pi^+$ two body decay. The refitted tracks are organised in four lists of stable particles, two for the two possible charges of the K mesons and two for the π mesons. All possible combinations of negatively and positively charged tracks are created by a simple nested loop algorithm. The outer loop runs over the K^- list. For each K^- candidate a second loop runs over all π^+ candidates. In this inner loop the decaying particles are constructed by combining the two stable particles. Most of the combinations constructed this way do not correspond to real D^0 mesons. Many of these wrong combinations can be discarded already at this stage because their invariant mass is far off the known D^0 mass. The best estimate of the invariant mass is available only after a full vertex fit was performed. Since vertex fits are expensive in terms of computing time a rough estimate of the invariant mass is computed from the track four-momenta. The invariant mass computed this way has a large uncertainty since the track momenta are taken at the perigee. Thus one has to be careful to cut not too hard on this quantity. Candidates are accepted if the invariant mass computed from the track momenta is inside a 160 MeV window centred at the known D^0 mass. For these candidates the K^- and π^+ tracks are fitted to a common vertex using the CTVMFT vertex fitter[39]. This vertex fit yields the vertex position and the four-momentum of the decaying particle as well as the corresponding covariance matrices. With a better estimate of the invariant mass now available the acceptance window is narrowed down to 100 MeV. Only combinations with successful vertex fits and invariant masses inside this window are accepted as D^0 candidates. Finally the two tracks in the final state are required to form a *trigger pair*. This means they have to resemble two tracks that would pass the Two Track Trigger. The requirements are not exactly the same as for the Two Track Trigger. Only the impact parameter and transverse momentum requirements are applied, i.e. $100\ \mu\text{m} < |d_0| < 1$ mm, $p_t > 2$ GeV and $\sum p_t > 5$ GeV. This trigger pair requirement greatly improves the signal to background ratio and removes some systematics that would otherwise complicate the correlation analysis described in section 4.3.

In the case of the $D^+ \to K^-\pi^+\pi^+$ three body decay the situation is slightly more complicated. The basic algorithm is the same as in the $D^0 \to K^-\pi^+$ case with one inner loop added for the second pion. A new problem arises because the two pions have the same charge. If one naively builds all combinations there will be some containing the same track twice. This can be avoided by arranging the loops appropriately. In order to be able to use some generic algorithms for reconstructing multi body decays a different solution was chosen, however. All reconstructed charged particle tracks in an event have an unique identification number. The unwanted combinations are then filtered out by requiring that no track identification number occurs twice. Otherwise the algorithm is the same as in the case of D^0, including the requirement that two of the three track in the final state form a trigger pair.

The $D^{*+} \to K^-\pi^+\pi_s^+$ also has a three body final state. This suggests that the same procedure as in the D^+ case should be applied. But since D^{*+} decays resonantly via D^0, i.e. $D^{*+} \to D^0(\to K^-\pi^+)\,\pi_s^+$, a different approach was chosen. Instead of combining three tracks, the algorithm combines already reconstructed D^0 candidates with positively charged tracks corresponding to the soft pions. Another difference

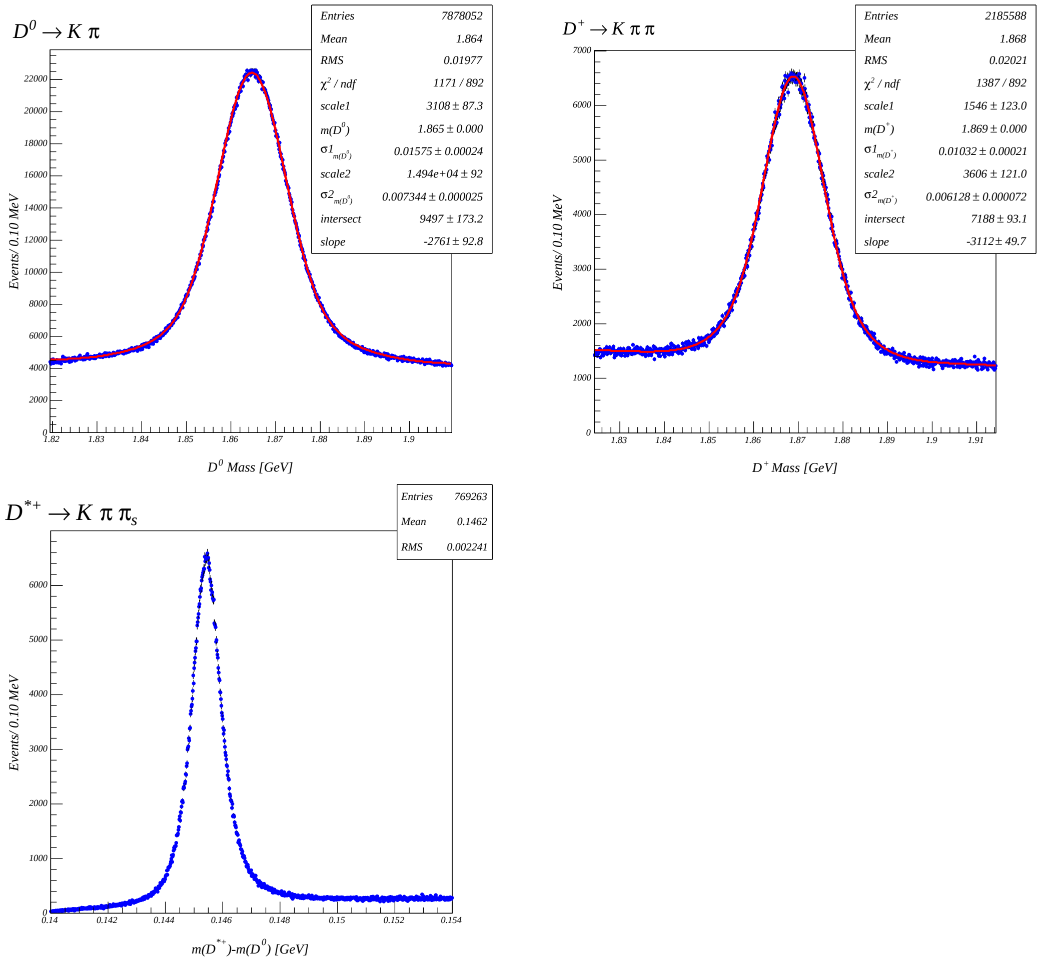


Figure 4.1: Mass distributions of the reconstructed D meson decay channels.

is in the way the invariant mass cut is implemented. The cut is not applied to the invariant mass of the D^{*+} candidates themselves but rather to the mass difference $m(D^{*+}) - m(D^0)$. This difference is known with much higher accuracy than the mass itself because most systematic effects of the reconstruction cancel. D^{*+} are accepted if 140 MeV $< m(D^{*+}) - m(D^0) <$ 155 MeV. The other requirements are the same as for the D^+ candidates, again including the presence of a trigger pair.

D Meson Mass Spectra.— The four-vectors resulting from the reconstruction algorithm described in the previous section are written to disk in a format suitable for later analysis with the ROOT data analysis framework. The D meson signals are established by analysing their mass spectra.

In the case of D^0 and D^+ mesons the invariant mass is plotted in a 90 MeV window centred at the known meson mass. Next a function describing the mass distribution is fitted to the resulting histogram. This function is the sum of a background and a signal model. The background model is a linear function and the signal is a double Gaussian with same mean. For the D^{*+} the mass difference $m(D^{*+})-m(D^0)$ is plotted in the range 140 MeV $< m(D^{*+}) - m(D^0) <$ 154 MeV. No fit is performed for the $m(D^{*+}) - m(D^0)$ mass difference. The results of this analysis are shown in fig. 4.1.

In all three decay channels a clear signal was established. This is not trivial in a hadron collider environment. The possibility to collect such a large number of events with hadronic final states of long-lived particles is mostly owed to the displaced track trigger. But without highly efficient and stable track reconstruction software it would have been impossible to analyse this huge dataset and extract the signals from the large combinatorial background.

The width of the D^0 and D^+ signal is not the natural width but reflects the experimental momentum resolution. It is furthermore remarkable how well the D^0 and D^+ masses are reconstructed. With $m(D^0) = 1864.76\,\text{MeV}$ and $m(D^+) = 1868.97\,\text{MeV}$ both are well in between one σ of the PDG world averages of $m(D^0) = 1864.6\pm0.5\,\text{MeV}$ and $m(D^+) = 1869.4 \pm 0.5\,\text{MeV}$[1]. This clearly proves the functionality of the track reconstruction and the accuracy of the detector model.

4.3 Correlation Analysis

Concept of the Analysis.— The correlation analysis presented here aims to find correlations between the mass spectra of charmed and anti-charmed mesons. It is based

[1] The statistical errors of the reconstructed masses are negligible and a study of the systematic errors was not performed since we do not want to measure the D meson masses. Assigning proper errors to the mass measurements would only improve the statistical agreement.

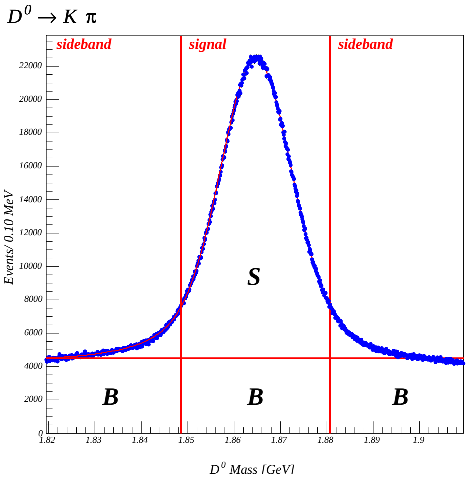
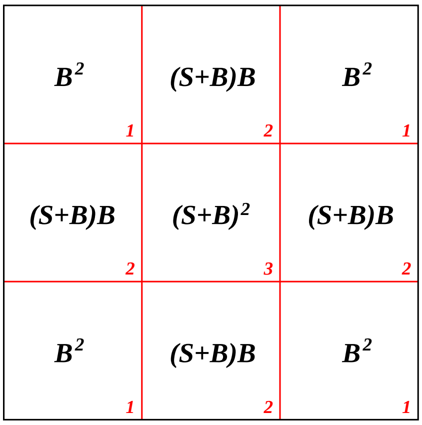


Figure 4.2: Signal and sideband regions in the D^0 mass spectrum. S and B denote the signal and background probabilities, respectively.



B^2 1	$(S+B)B$ 2	B^2 1
$(S+B)B$ 2	$(S+B)^2$ 3	$(S+B)B$ 2
B^2 1	$(S+B)B$ 2	B^2 1

Figure 4.3: Illustration of correlation analysis. The labels in the nine bins denote the probability for a combination to end up in this bin in the case of zero correlation.

on some assumptions. Most importantly the mass spectra are required to be symmetric and the background to be flat. Furthermore the method is only valid when the two mass spectra have the same shape. Given the shape of the spectra presented in the previous section this leaves us with D^0-$\bar{D}^0$ correlations. Similar methods can be developed that allow to include other possible correlations. But these are more complicated and not yet well understood. To examine them all is beyond the scope of this document.

In order to understand the analysis method it is first necessary to introduce some concepts and notations. The mass spectra are divided in three regions of equal width. The central region contains the mass peak and is called the *signal region*. The two regions left and right of the signal region are called the *sideband regions*. We now assume that the sideband regions essentially contain no signal. For the given D^0 and $\bar{D}^0$ mass spectra this is true to a very good approximation. We can now interpret the mass spectra as probability density functions. This means that there are definite probabilities to find a value of m in the signal or sideband regions. With the already introduced assumption of a flat background the background probability is constant over the whole mass range. We denote this probability $3 \cdot B$. Since, to very good approximation, there is no signal in the sidebands, B is the probability to find an entry in one of the sideband regions. The area in the central region of the mass spectrum is larger since it also contains signal events. If we denote the signal probability with S, the total probability to find an entry in the signal region is $S + B$. This is illustrated in fig. 4.2. We can now plot the $\bar{D}^0$ mass versus the D^0 mass spectrum. Possible correlations should express themselves in the count rates of the resulting two-dimensional histogram. The two dimensional plot is divided into nine regions of equal area as in fig. 4.3. The borderlines of the regions are at the very positions that defined the signal and sideband regions of the mass spectra. In absence of any correlations we can now compute the probabilities for an entry ending up in one of the regions and hence predict the relative count rates of the nine regions. For two uncorrelated events with probabilities P_A and P_B the total probability of observing both is simply $P = P_A \cdot P_B$. This yields the following probabilities for the three classes of regions at hand:

1. sideband-sideband regions: $P_1 = B^2$
2. sideband-signal regions: $P_2 = B(S + B)$
3. signal-signal region: $P_3 = (S + B)^2$

Since we are only interested in relative probabilities we can neglect the overall norm and directly identify the count rates with the probabilities. If we denote the number of entries in all regions of type 1 and type 2 with N_1 and N_2 respectively, we can compute S and B. Starting from

$$N_1 = 4 \cdot P_1 = 4 \cdot B^2 \qquad (1)$$

$$N_2 = 4 \cdot P_2 = 4 \cdot B(S + B) \qquad (2)$$

we get

$$B = \frac{1}{2}\sqrt{N_1} \qquad (3)$$

$$S = \frac{N_2 - N_1}{2\sqrt{N_1}} \qquad (4)$$

This allows us to predict the count rate in the signal-signal bin:

$$N_{3,\text{pred}} = (S + B)^2 = \frac{N_2^2}{4N_1} \qquad (5)$$

If the observed count rate N_3 is significantly higher than this prediction we have established evidence for correlated D^0-$\bar{D}^0$ production. In order to quantify the significance we need an estimate of the error on $\Delta \equiv N_3 - N_{3,\text{pred}}$. We assume the count rates to be independent and Poisson distributed. This directly yields their errors

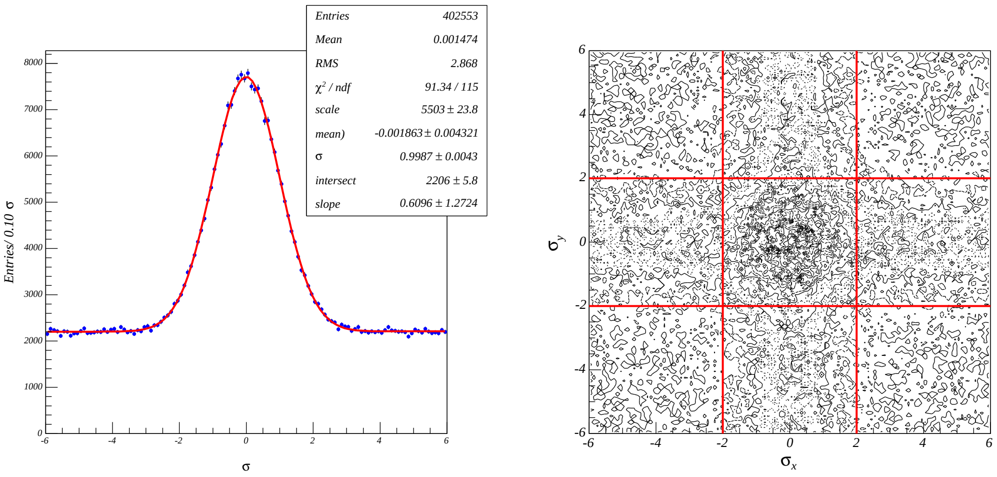


Figure 4.4: Toy Monte Carlo Analysis for $sb = 2/3$ and $rc = 0.2$. The evidence for correlation is 7.8 σ.

$\sigma_{N_i} = \sqrt{N_i}$, $i = 1, 2, 3$. Since there are no correlations between the N_i the error on Δ is obtained through simple error propagation:

$$\sigma_\Delta = \sqrt{\sum_{i=1}^{3} \left(\frac{\partial \Delta}{\partial N_i}\right)^2 \cdot N_i} \tag{6}$$

$$= \sqrt{\frac{N_2^2\,(4N_1N_2 + N_2^2)}{16N_1^3} + N_3} \tag{7}$$

The ratio Δ/σ_Δ then quantifies the significance of the correlation.

Toy Monte Carlo Study.— Before we apply the analysis method to the D^0 mass spectra we verify its functionality by conducting a toy Monte Carlo study. It is a toy Monte Carlo study in the sense that it does neither involve the generation of physics events nor a detector simulation. This is justified by the fact that the correlation analysis method is completely ignorant of the meaning of the involved spectra. Thus the method can be verified if we produce two spectra for which we have the signal to background ratio and their correlation under control.

This is achieved by the following algorithm. Using a pseudo random number generator we produce two spectra, x and y. The shape of both is a constant flat background plus a standard Gaussian with mean zero and variance one in the range $[-6, 6]$. The spectra are produced on an event by event basis. The number of entries to the spectra in each event follows a Poisson distribution with mean two. In each event the algorithm first decides how many entries of x spectrum will be generated. For each x entry the random number generator is used to decide whether it is a signal or a background event. The signal to background ratio is controlled by one variable, sb, in a simple way. The random number generator generates a number in the interval $(0, 1]$. If this number is greater than sb the entry will be a signal entry. This means that the signal to background ratio is $(1 - sb)/sb$. The background entries are generated according to the flat background distribution. The signal event distribution is the standard Gaussian distribution. The rate of correlation is controlled in very much the same way as the signal to background ratio. For each signal x a random number in $(0, 1]$ is compared to the control variable rc. If the random number is less than rc a signal entry in y distribution will be forced. This is repeated until all entries for the x distribution in this event have been generated. Afterwards the y entries are generated in a similar way. However, no correlated entries are created for signal events in y. Consequently the maximum correlation rate at $rc = 1$ is 50%. This procedure introduces an asymmetry between x and y. To compensate for this the roles of x and y in each event are swapped on a random basis by comparing a random number to 0.5. This algorithm was implemented in C++ and run inside the ROOT framework, using the `TRandom3` random number generator.

By generating a large number of events in this fashion we obtain two spectra with arbitrarily high statistics. The signal to background ratio and the rate of correlation are completely under control. If we create two correlated spectra for which the assumptions underlying our analysis method hold, we should find a clear evidence for correlated production. This is indeed the case. If we produce 100,000 events with

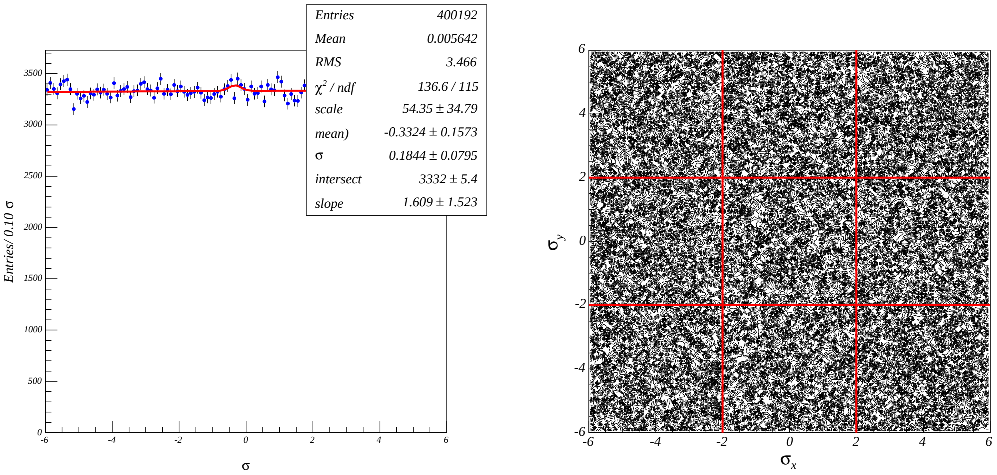


Figure 4.5: Toy Monte Carlo Analysis for $sb = 1.0$ and $rc = 1.0$. The evidence for correlation is 0.06 σ.

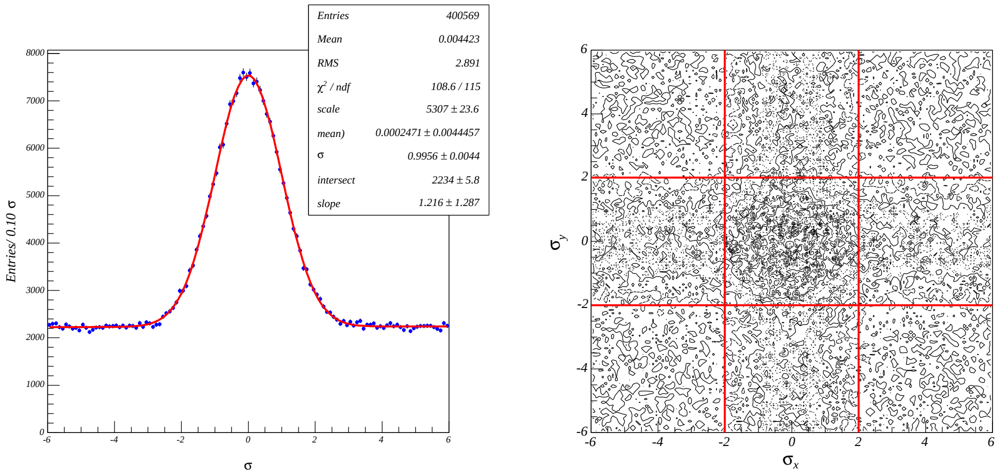


Figure 4.6: Toy Monte Carlo Analysis for $sb = 2/3$ and $rc = 0$. The evidence for correlation is 0.55 σ.

$sb = 2/3$ and $rc = 0.2$, i.e. with a signal to background ratio of 1/2 and a correlation rate of 10%, we obtain a 7.8 σ evidence for correlation. Figure 4.4 shows the corresponding plots.

It remains to be shown that the method does not create fake evidence and breaks down when one or more basic assumptions are violated. Also this is indeed the case. Figure 4.5 shows the results for maximum correlation but no signal. As expected, the evidence for correlation vanishes in this case. In the case of a reasonable signal to background ratio with zero correlation the evidence also vanishes, as can be seen in fig. 4.6. The results of a more granular scan of the (sb,rc) parameter space can be found in appendix D. In summary, the analysis method is suitable for establishing a correlation of two spectra as long as the underlying assumptions hold.

4.4 Evidence for Correlated Charm Production

Evidence for Correlated D^0-$\bar{D}^0$ Production.— After we have established the analysis method we can now apply it to the D^0 mass spectra. In order to minimise possible sources of errors the analysis algorithm was implemented in a C++ class, allowing to use exactly the same implementation for the toy Monte Carlo study and data analysis. The analysis code was run inside the ROOT framework on the compressed data produced by the offline analysis job. Figure 4.7 shows the mass correlation plot. We observe an excess in the signal-signal bin of the D^0-$\bar{D}^0$ mass distribution with a significance of 5.9 σ. For the first time at a hadron collider experiment, this constitutes a clear evidence of correlated D^0-$\bar{D}^0$ production.

Angular Distributions.— It is now most interesting to examine whether the significance of the correlation is related to the angle between the flight directions of the two mesons. Figure 4.8 shows a plot of the azimuth angles of the meson momenta. The most striking features are the three clearly visible bands in the plot. The band on the main diagonal is populated by meson pairs with collinear momenta, i.e. $\Delta\phi \sim 0$. The bands in the upper left and lower right of the plot correspond to meson pairs with a large separation of $|\Delta\phi| \sim \pi$. We thus observe a clear jet structure: the meson pairs are either produced collinearly or back to back. This raises the question whether the correlation is equally strong in both cases. In order to answer this question we divide the data sample in two subsamples, a collinear sample with $|\Delta\phi| < \pi/2$ and a back to back sample with $|\Delta\phi| > \pi/2$. The correlation analysis is then performed for both samples separately. The angular distribution and correlation plots for the two samples are shown in fig. 4.9. We observe a difference in the correlation significance in the two cases. The analysis of the collinear sample yields an excess with 5.3 σ significance. With 2.6 σ the evidence is much weaker in the back to back sample. There several scenarios that can lead to this result. If the production vertex of the two mesons is the primary interaction point this would hint a strong contribution from higher order production mechanisms like flavour excitation and gluon splitting. On the other hand a significant displacement of the meson production vertex from the primary interaction

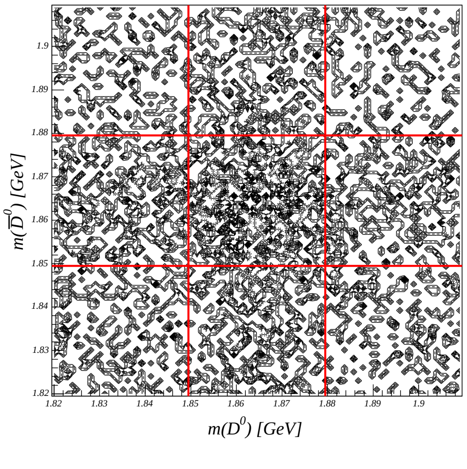


Figure 4.7: $\bar{D}^0$ mass spectrum versus D^0 mass spectrum

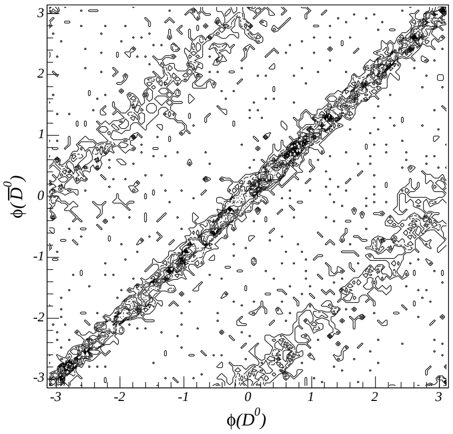


Figure 4.8: Angular distribution of D^0 and $\bar{D}^0$ meson momenta. A clear jet structure is visible.

point suggest that the meson pairs are produced in the decays of long-lived particles.

This leads us to examine the distribution of the proper decay lengths of the meson pair production vertices. To this end we plot L_{xy} versus $\cos(\Delta\phi)$ of the meson momenta. Note that L_{xy} is not the distance of production vertex and primary interaction point. It is the projection of this distance in the x-y plane onto $\vec{p}_t$, i.e.

$$L_{xy} \equiv \frac{(\vec{x}_{\text{secondary}} - \vec{x}_{\text{primary}}) \cdot \vec{p}_t}{|\vec{p}_t|} \qquad (8)$$

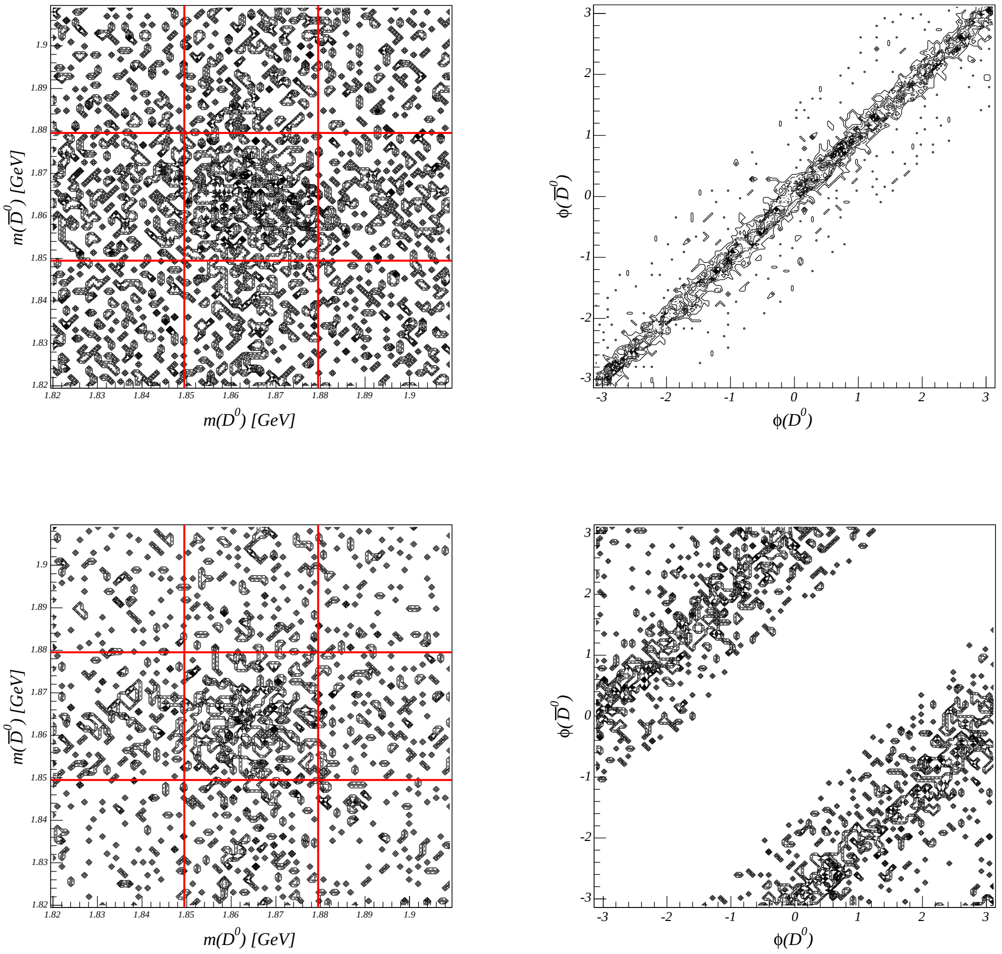


Figure 4.9: Correlation and angular distribution for collinear (upper plots) and back to back (lower plots) meson pairs.

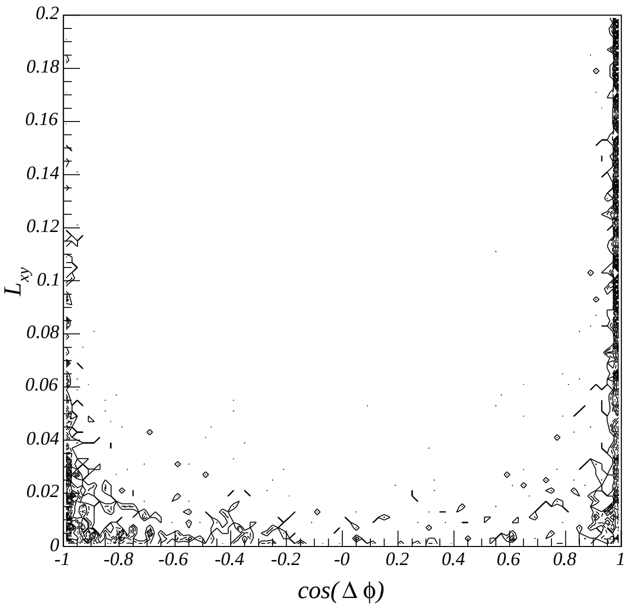


Figure 4.10: L_{xy} versus $\cos(\Delta\phi)$ of the meson pair momenta. The collinear meson pairs tend to originate from a vertex with large decay length.

The plot in fig. 4.10 shows that the collinear meson pairs tend to be produced at large decay lengths. This suggests that they originate from decays of long-lived particles rather than higher order production mechanisms. While this is well worth further investigations these are beyond the scope of this thesis because they require even larger statistics than are available at the time of this writing. Thus we are left with open questions. But one question is answered beyond any doubt: the developed software for charged particle track reconstruction is sufficiently accurate, efficient and stable to allow for large scale analysis of the hadronic dataset recorded by the CDF 2 experiment.

Chapter 5

Conclusions

In Run 2, the second period of operation of the Tevatron, the CDF 2 collaboration has collected and successfully processed a large amount of data. This is a tremendous success, owed to the substantial upgrades of the detector hardware as well as to the efficiency and stability of the reconstruction software and computing facilities.

For the first time at a hadron collider CDF 2 has commissioned a displaced vertex trigger. With this trigger alone CDF 2 has collected $\sim$ 481pb^{-1} of integrated luminosity. Although designed for collecting hadronic decays of B mesons, this trigger has provided the CDF 2 collaboration with a huge sample of charmed mesons decaying into fully hadronic final states. This allows to examine Charm production in hadron collisions using exclusively reconstructed charmed meson decays with very high statistics.

An efficient and stable charged particle track reconstruction is an essential prerequisite of any analysis of fully hadronic decays. Based on an extremely fast implementation of a Kalman-fitter a very pure and efficient tracking algorithm for the CDF 2 silicon detector was implemented. Furthermore, the implementation of an accurate and efficient model of the detector geometry allows to fit tracks with arbitrary particle hypothesises at the correct momentum scale while maintaining high computing efficiency at the same time. All analyses using charged particle tracks in the CDF 2 collaboration heavily rely on the functionality implemented by this software. These algorithms were improved through numerous development cycles and subjected to many tests of their functionality. Without these time consuming efforts it would have been impossible to extract the clear D meson signals that ultimately allowed to establish evidence for correlated D^0-$\bar{D}^0$ production for the first time in a hadron collider environment.

Appendix A

SiliMap Implementation

Class Structure — The Memory Problem — Material Integration

A.1 Class Structure

The implementation of the SiliMap concept involves several classes, distributed over several packages. The core functionality, including I/O, is implemented in the `SiliconGeometry` package[26]. The code responsible for the scanning resides in `TrackingUserMods`[40] and the standard `MaterialIntegrator` interface is implemented in `TrackingSI`[41]. The complete list of classes where they reside and the functionality they provide is given in the table below[1].

Class	**Package**	**Functionality**
`SiliMap`	`SiliconGeometry`	main interface, ASCII I/O
`SiliMapLayer`	`SiliconGeometry`	map constituent, core material integration algorithm
`SiliMapMaterial`	`SiliconGeometry`	layer constituent, holds material properties
`StorableSiliMap`	`SiliconGeometry`	provides `StorableObject` I/O
`SiliMapIntegrator`	`TrackingSI`	generic `MaterialIntegrator` interface
`SiliMapScanMod`	`TrackingUserMods`	scanning algorithm, scan job setup, plain Root tree output

The `SiliMap` main interface is implemented as a singleton. To avoid the implementation of trivial setters not needed for the client API, all classes, especially in I/O, are declared friends of the classes they need to modify. We do not claim that the

[1] The class `SiliMapDBInterface` is not listed here since database I/O is obsolete since quite a while. The class will be removed in future releases of cdfsoft2.

overall design is optimal. For example it might considered inconsistent to implement `StorableObject` I/O in a separate class and making ASCII I/O a `SiliMap` functionality. This and other flaws are owed to either technical requirements imposed by the CDF 2 software framework or simply to the development history. Once a peace of software is in production it becomes hard to make fundamental changes to it. This is not meant to be an excuse. It should just illustrate an all too common problem, not only in high energy physics: when things basically work and resources are limited, everything is left as is. Sometimes things work not due to their design but rather in spite of it. We hope that `SiliMap` is not such a bad case, however. The class structure is documented here to help maintainers and not because it represents an ground-breaking design solution for the problem at hand.

A.2 The Memory Problem

Naively storing the three material properties obtained by the scan for each bin in the map as a `float` is not feasible. This can be seen from the following calculation. Assuming a binning resolution of 2 mm, an average map layer has 1000×500=500,000 bins. With 30 layers and 4 bytes per `float` the total memory would be ~170 MB. We can not afford to enlarge the size of offline executables by this amount just to store a material description. Furthermore, a map this large would considerably slow down program startup when it is read in.

The solution to this problem is threefold:

- exploit detector symmetries
- use a fuzzy definition of equality
- use a sophisticated indexing scheme

Instead of storing three `float`s per bin a per-layer mapping scheme is implemented: only for bins *distinct* in their material properties these are stored. Whenever a bin scan is completed the already stored material properties in the layer a searched for a match. The matching is where the fuzzy equality comes in. The accuracy of equality is defined by `SiliMapMaterial::ACCURACY` which defaults to 1%. This means that bins whose material properties do not deviate by more than 1% are considered equal. If a match is found, a reference to the already stored properties defines the "material" of the bin. Since the ϕ-binning is chosen to reflect the detector symmetry and there are additional translational symmetries, matches are very common. Thus the memory consumption is lowered considerably[2].

[2] One might think that this can be improved upon by only storing 'materials' distinct in the map scope. However the gain has proven to be too small to sacrifice the ability to do a parallel scan.

This, however, is not yet the end: storing the references as pointers to `SiliMapMaterial` still consumes too much memory. On a 32 bit system a pointer is still four bytes, on 64 bit SGI even eight. So even if all the layers had many equal bins we still would end up with >57 MB on a 32 bit system and >114 MB on native 64 bit. Firstly this is too large and secondly it is architecture dependent.

The best approximation of an architecture independent type in C++ is a `char`. In most implementations it corresponds to one byte. Since this is four times smaller than a pointer on a 32 bit system we choose it as the index type. A `SiliMapLayer` is then defined by

- a vector of distinct (in the above sense) `SiliMapMaterials`
- a vector of `char`s that index the 'materials'

One caveat remains, however. In some layers there are more than 256 distinct 'materials'. In these cases one byte is not sufficient to index them all. To work around this two `chars` are used in these layers, which requires some bit logic to compute the indices (using an `int` would consume four bytes and `short` is not well defined).

Using the machinery described above the runtime memory consumption of SiliMap is no more than ~17 MB, architecture independent. This is a factor of ten lower than with the naive approach and sufficiently low to use it in offline reconstruction. Persistently stored as a `StorableObject`, i.e. with Root compression, the file size is only ~1 MB. The time spent in reading the map on program startup is hardly noticeable.

A.3 Material Integration

In general the material integration between two arbitrary points on a given trajectory (always a helix in our case) is done in two steps:

1. determine all SiliMap layers traversed by the part of the trajectory
2. integrate the material in all layers by computing average material properties

Since the first step is not necessary for clients that already know about the layer they want to integrate over (e.g. the KAL fitter) it is not implemented in `SiliMap` itself. The `SiliMapIntegrator` can be used for the general case at the cost of some loss in timing performance.

The core material integration algorithm is common to all SiliMap access scenarios and is implemented in *`SiliMapLayer::integrate()`*. It works as follows. Prior to a call of *`SiliMapLayer::integrate()`* the ϕ and z coordinates of the trajectory entry (at inner layer radius) and exit point (outer layer radius) has to be known. From this

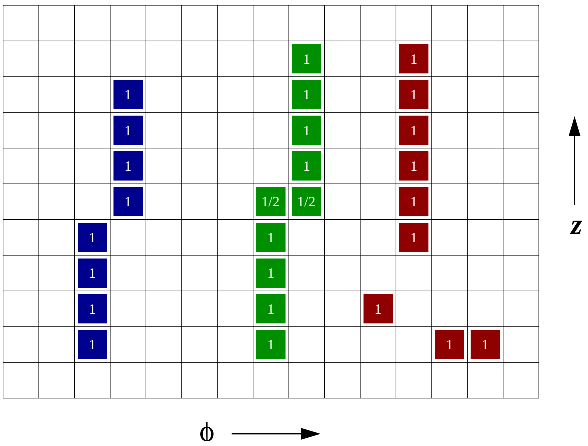


Figure A.1: Illustration of the possible scenarios for the material integration in one SiliMap layer. Non-trivial cases for even (left, blue) and odd (centre, green) number of z bins and the trivial cases (right, red). The numbers indicate the weight given to the bins in the averaging of the material properties.

information *SiliMapLayer::integrate()* will determine the bins traversed by the trajectory in the layer and average their material properties. In the most general case the z and ϕ bin indices will differ for the entry and exit points. The other scenarios are trivial special cases (only one bin, two neighboured ϕ bins and a row of z bins). In the non-trivial case two rows of z bins enter the averaging procedure. One aligned with the entry point ϕ bin and one with the exit point ϕ bin. In case of an odd number of z bins the two neighboured bins in the centre enter the average with half weight. This is illustrated in figure A.1. After the contributing bins and their weights are determined the average material properties are computed and returned in a `SiliMapMaterial` struct. In all cases the path length in the layer is computed accurately – either by the `SiliMapIntegrator` or directly by the client as in the case of KAL fitter. Using the path length and material properties the client can then modify the track parameters and covariance matrix accordingly. This procedure assumes that a trajectory never traverses more than two bins in ϕ in one SiliMap layer. Obviously, this is not true in general. The assumption allows for a better timing performance, however. And it can be justified by the following calculations. Geometrical considerations yield the following equation for the angular difference of the entry and exit points as seen from the origin:

$$\Delta\phi = \arcsin(r_2 \cdot \kappa_C \cdot B/2p_t) - \arcsin(r_1 \cdot \kappa_C \cdot B/2p_t) \qquad (1)$$

Where $r_1(r_2)$ is the inner(outer) radius of the map layer, $\kappa_C = 0.0029979$ is the curvature constant $B = 1.4116$ T is the magnetic field strength and p_t denotes the track transverse momentum. The assumption is thus justified if this difference is smaller than the angle covered by two bins in a given layer. Furthermore, the problem is not severe if the material properties vary slowly in ϕ (or not at all, as in the case of e.g. the COT inner cylinder). The following table lists some scenarios. The numbers for $\Delta\phi$ in the table are rounded results obtained from equation (1). But the arcsin() function actually is linear to a good approximation near the origin.

r_1 [cm]	r_2 [cm]	p_t [GeV]	# ϕ bins	allowed $\Delta\phi$ [°]	$\Delta\phi$ [°]
20	21	0.25	628	1.0	0.5
40	41	0.25	1000	0.72	0.5
20	28	1.0	628	1.0	1.0
20	28	0.5	628	1.0	2.0
20	28	0.25	628	1.0	4.0

The only cases where the two bin assumption does not hold are the last two. Note, however, that these assume a layer of 8 cm thickness! All SiliMap layers with rapidly changing material properties are thinner than 1 cm. For these the assumption holds even for very low momentum tracks.

Appendix B

Comparison of Passive Material Descriptions

The Different Descriptions of the Passive Material: Introduction; The Perfect World – Using GEANT; The Independent Simplified Material Model; The Detailed Structured Subset — Comparison on Single Muon Monte Carlo — Comparison on J/ψ Data — Timing Performance

B.1 The Different Descriptions of the Passive Material

Introduction.— In this section we will present the various combinations of track fitters and material descriptions. Although only two of them are currently used in physics analysises it will be useful to have an overall picture. One reason is that this nicely illustrates some aspects of how long-lived software projects evolve. But most importantly it justifies why the two combinations in use are the ones recommended for analysis. The SiliMap description of the passive material was already described in chapter 3. The other available descriptions will be introduced in the following sections, especially emphasising their consistency and maintainability.

The Perfect World – Using GEANT.— One major achievement of the CDF 2 geometry description is that it allows to use the same geometry in simulation and reconstruction. While the main reason for this design was to ensure consistency, it also allows the usage of simulation software in reconstruction. This directly leads to the 'perfect world' solution of the material integration problem: using GEANT to find intersections and simulate the material effects. This solution is implemented in the `G3X` package[**?**]. It can not be used by the KAL fitter in its current implementation.

This approach has the unquestionable advantage of the highest available accuracy. Unfortunately it has the drawback that GEANT is comparably slow – GEANT's timing performance is to poor to employ it under circumstances with high combinatorics[1]. Nevertheless, the `G3X` package provides a great tool for the analysis of preselected data sets. Moreover, it constitutes an indispensable ingredient to the SiliMap solution.

The Independent Simplified Material Model (SI Simple).— A completely different approach is to ignore the full-featured geometry description and implement a simpler one, specially tailored to simplify the path finding problem. This solution is implemented by the `SimpleSiliconIntegrator` in the `TrackingSI` package[41]. It is only used by the SI fitter. Historically a consistent geometry framework did not yet exist when this solution was implemented. So, strictly speaking, it was not ignored. Be this as it may, the `SimpleSiliconIntegrator` is still used by some fitter clients and tracking algorithms, despite the fact that a much more detailed geometry is available since quite a while.

The major drawback of this approach is the likely divergence of the detailed and the simplified geometry description. And in fact both meanwhile describe rather different detectors. One should keep in mind that this solution always was intended to be temporary. Until the rise of `G3X` it was the only one available for the SI fitter.

The Detailed Structured Subset (KAL Proxy).— The divergence problem is partially avoided by using a subset of the detector components contained in the detailed geometry description and superimpose a structure which facilitates the path finding. Since the KAL fitter relies on such a structure, this was the only option available for this fitter prior to SiliMap. The SI fitter does not use it. The selection criterion for the subset is that the shape and orientation of the component can be handled analytically. I.e. it is possible to analytically compute intersections and path lengths for helices. This boils down to cylinders and box shapes parallel to the z-axis. The superimposed structure arranges the components in layers (r), wedges (ϕ) and ranges (z) which allows for a very efficient lookup of traversed components. An incarnation of this concept is the `KalDetector` proxy used by the tracking algorithms implemented in the `TrackingKal` package and in track refitting[31]. Since the subset covers the majority of detector components this solution performs pretty well.

However, it also has its drawbacks: the selection criterion is somewhat arbitrary. It might reject relevant detector components simply because they can not be handled analytically. In addition, the interface of the detailed geometry was not designed to support the initialisation of proxies. This raises maintainability problems, since the initialisation procedure has to rely on some assumptions on the geometry structure

[1] It should be stressed that GEANT is not slow in the field of simulation. It simply dominates the CPU consumption when used in the context of track fitting.

and naming conventions[2]. In case of conflicts manual intervention is required.

B.2 Comparison on Single Muon Monte Carlo

The main purpose of the MC validation is to verify the consistency of SiliMap and the full detector description. Thus single-μ MC samples are sufficient for this purpose. Several samples were generated for this validation. The results presented here are all obtained from a 10,000 event single-μ^- sample with $p \in [0.5, 2.0]$ GeV, $|\eta| < 1$. The sample was generated with cdfSim from cdfsoft2 5.3.1 using the FakeEvent generator. The 'phantom layers'[3] were off for simulation , SiliMap scanning and `G3XIntegrator` refitting. This actually does not matter in a consistency check. The settings have to be the same whenever GEANT is invoked, however. Standard `ProductionExe` was run for track reconstruction. The evaluation ntuple was created by running `SiFitterTestMod` from the `TrackingUserMods` package. This module can be configured to employ any of the available fitter/material integrator configurations via the `TrackRefitter` interface. The truth informations was retrieved from the OBSP banks. It is necessary to look at only one charge to in order make the systematic effects in curvature reconstruction visible. The results do not look different for the corresponding μ^+ sample. Only OI tracks entered the refit ntuple. The reason for only using OI tracks is that the SI fitter is needed for `G3XIntegrator` refitting. This fitter was never validated for refitting silicon stand-alone or IO tracks and it actually has some problems, especially in the forward regions. Since this comparison is only concerned with material integrators only tracks for which both fitters are known to work properly are used.

The parameter pulls for five p_t^2 bins in the region $p_t \in [0.5, 2.0]$ GeV are shown in figure B.1. To allow for visual comparison all plots have the same range on the pull axis. On this scale only the curvature pull shows visible differences between the fitter configurations. The curvature pull clearly shows that the KAL/Proxy configuration underestimates the detector material. The dE/dx estimated by this configuration is too low which leads to a deviation of the curvature estimate from the OBSP truth. As expected, this deviation grows with falling momentum.

On the other hand the KAL/SiliMap configuration performs as good as the `G3XIntegrator`. This means it shows close to perfect consistency with the full featured

[2] This is not true for active detector components: `CdfHalfLadderSet` provides a well defined interface to them.

[3] The phantom layers are material layers introduced into the geometry description in order to get the right momentum scale. They are tuned to the J/ψ mass. First there was only one such layer placed between the ISL and the COT inner wall. At this time it could not be associated with real detector components. Later the description was refined and now there are several such layers. Recent investigations allowed to associate these layers with real detector components. Thus the term 'phantom layer' is actually no longer appropriate.

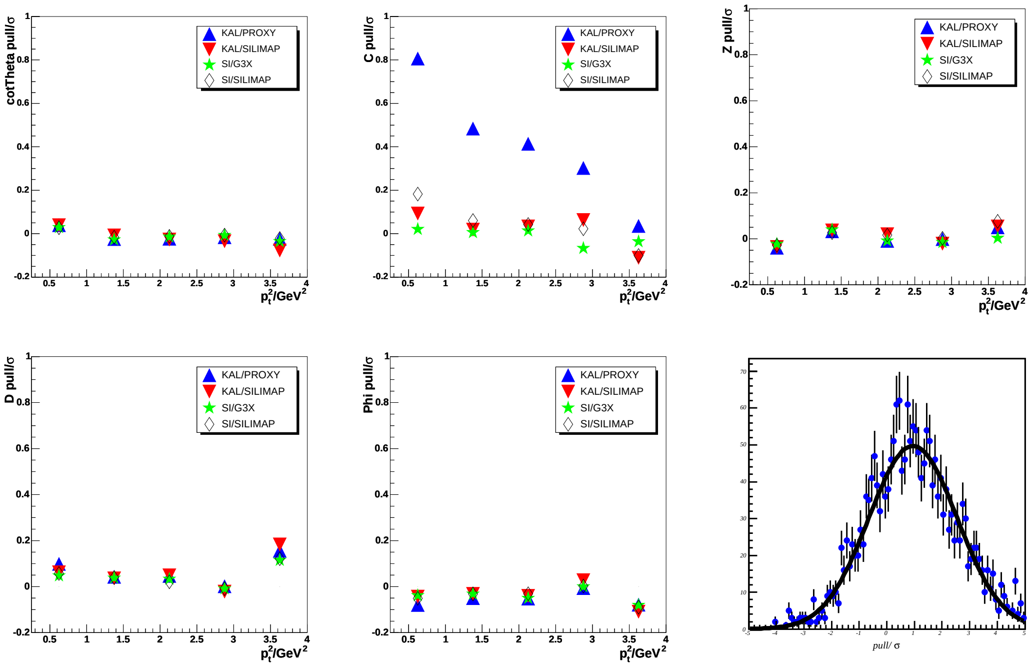


Figure B.1: Track parameter pulls of the various fitter/material integrator configurations. For each p_t^2 bin a Gaussian was fitted to the pull distribution as illustrated in the lower right plot. The histograms show the Gaussian means as obtained from the fits.

geometry since `G3XIntegrator` integrates exactly the same material that was used for simulation in exactly the same way, i.e. using GEANT.

The other track parameters are only slightly affected by the fitter configuration. This can be seen in more detail in the zoomed views shown in figure B.2.

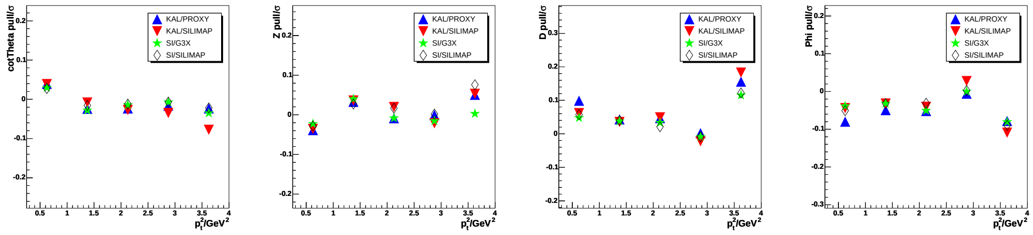


Figure B.2: Pulls of the less affected track parameters $cot(\theta)$, z_0, d_0 and ϕ_0. They were obtained in the same way as in figure B.1.

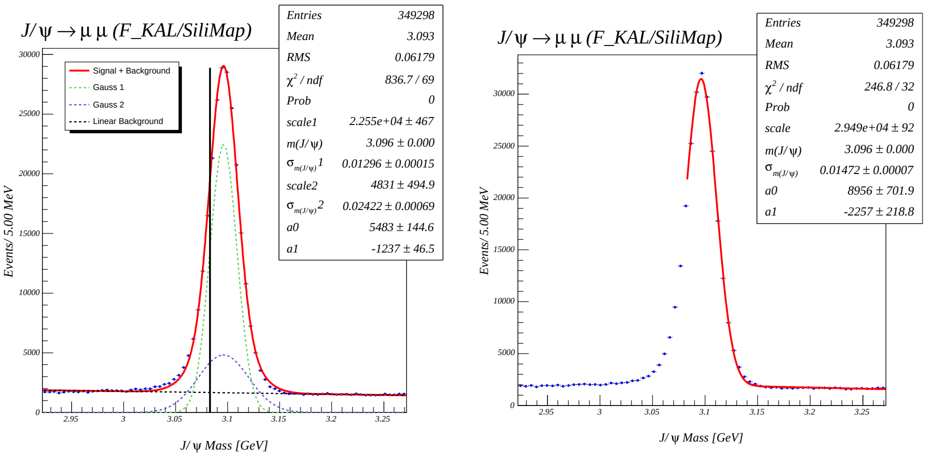


Figure B.3: J/ψ candidate mass distribution and fits used to determine the peak width. In the left plot a double Gaussian with common mean plus linear background was fitted to the distribution. The result determines the range for the single Gaussian fit in the right hand side plot.

B.3 Comparison on J/ψ Data

The crucial test is the validation on data. SiliMap has to perform at least as good as the currently recommended tuned KAL/Proxy configuration[33] to be an alternative. That it actually performs better will be shown below.

The first 10% of the jpmm0c dataset were used for this validation. Statistics are thus not as high as in [33] but sufficient for this purpose. The J/ψ candidates were formed from two oppositely charged muons retrieved from *`CdfMuonView::defMuons()`*. The tracks were required to be OI tracks with $|\eta| < 1$ and $p_t > 1.45$ GeV. At least 20 axial 15 stereo COT hits and 2 axial silicon hits were required. The tracks were re-fitted with the various fitter/material integrator configurations using the `TrackRefitter` interface. In all cases the COT covariance scaling the COT curvature correction and the magnetic field scale from [33] were applied. The KAL/Proxy configuration was used in two different setups: once with the default 'phantom layers' of cdfsoft2 5.3.1 and once withe the 'phantom layer' tunings from [33] (KAL/6905). For SI/G3X and SiliMap the default 5.3.1 geometry settings were applied.

The tracks were fitted to a vertex using the `PartFitVertex` interface to CTVMFT. The resulting mass distribution is shown in figure B.3.

The mass pulls shown in the left plot of figure B.4 were computed using the PDG J/ψ mass and the errors from the CTVMFT fit using formula 1.

$$\text{pull}_m = (m - m_{\text{PDG}} + 0.00082\,\text{GeV})/\sigma_m \tag{1}$$

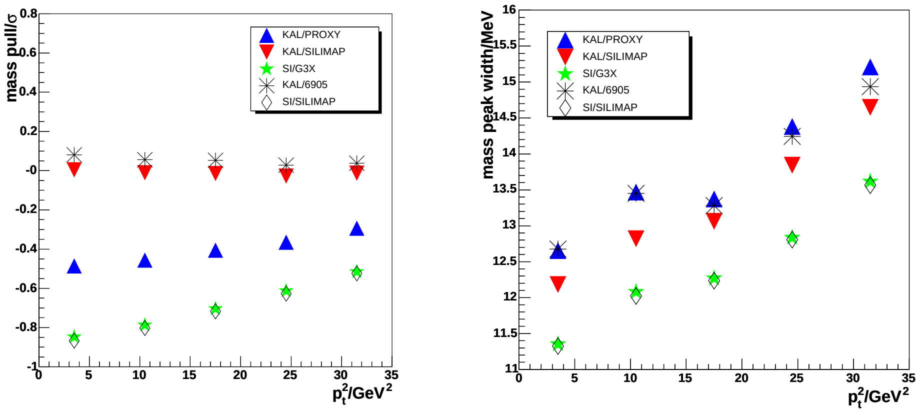


Figure B.4: Pulls (left) and widths (right) of the reconstructed J/ψ mass for all fitter/material integrator configurations.

The 0.00082 GeV are added to account for the radiative tail in the mass Distribution. This is the same procedure as used in [42][43][33]. The KAL/SiliMap configuration performs best concerning the pull. That is it has the smallest slope in the pull vs. p_t^2 distribution and the overall pull is closer to zero than in all the other configurations. While the advantage over the KAL/6905 configuration is only small, the KAL/SiliMap configuration does not require special geometry tuning.

B.4 Timing Performance

Since the aim was to provide an accurate *and fast* material integration the final test is a comparison of the timing performance of the available fitter/material integrator configurations. Figure B.5 shows a comparison of the time per track fit of all available configurations[4]. As expected, the `G3XIntegrator` is the slowest of them all. It is also the broadest distribution because the time spent in the GEANT stepping heavily depends in the number of different volumes traversed by the track. In the full featured detailed geometry description this number shows a large variation.

It was clear from the start that `G3XIntegrator` can never beat the others concerning speed. Figure B.6 provides a closer look at the faster combinations. This not only illustrates the difference in speed between the material integration but also the differences between fitters. The KAL fitter is more than a factor of two faster than the SI fitter[5], as can be seen from the comparison of the two for the SiliMap case. Part of

[4] The CPU type actually does not matter. It just sets the absolute time scale.

[5] By default the SI fitter is compiled with maximum optimisation. This was turned off for this

this might be due the abstract `MaterialIntegrator` interface that does not allow to use layer information available in the fitter (see section A.3). Be this as it may – the KAL fitter is the one recommended by the physics groups for analysis and thus the most important comparison is the one between KAL/Proxy and KAL/SiliMap. The KAL/SiliMap configuration is ∼25% slower than the KAL/Proxy configuration. Given the much more accurate and consistent material description provided by SiliMap this is hardly relevant. Especially when reconstructing exclusive decay chains using vertex fits, the percentage of time spent in the track fit does not change significantly when switching from KAL/Proxy to KAL/SiliMap.

The good timing performance would even allow to use SiliMap in pattern recognition. Especially the KAL tracking algorithms could benefit from that. To do so would, however, require to re-tune numerous parameters of the pattern recognition to find the best working point concerning efficiency and purity. But we do not know a priori whether this will pay off. The only advantage we are sure of is improved maintainability. The tracking group agreed that this is currently not worth the effort.

comparison to make it a fair one. The overall picture does not change significantly with an optimised SI fitter, however.

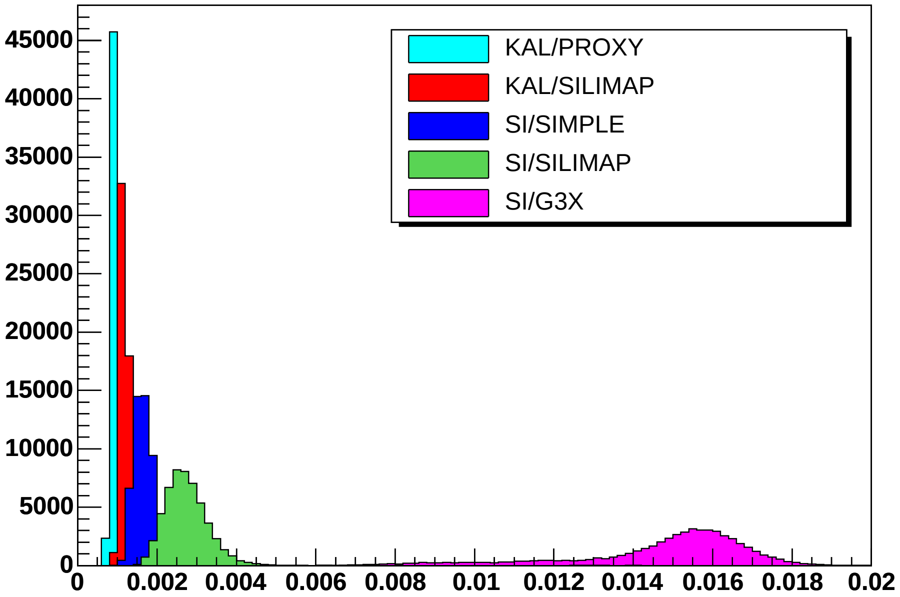


Figure B.5: Time per track fit for all fitter/material integrator configurations measured using `ZMtimer` on an AMD Opteron 240

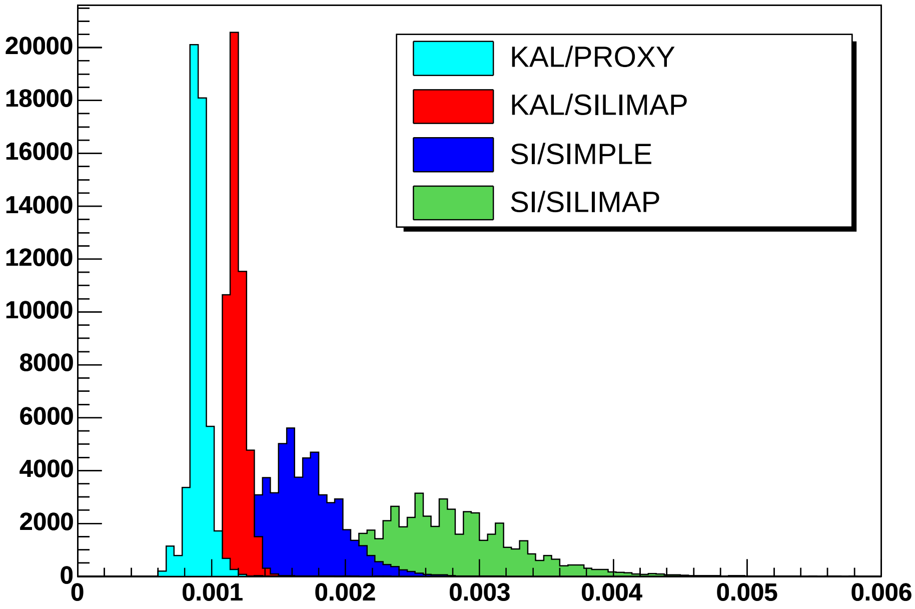


Figure B.6: Time per track fit for the faster fitter/material integrator configurations measured using `ZMtimer` on an AMD Opteron 240

Appendix C

SiliMap Scanning and Distribution

The Scanning Procedure — Setting up a Scan Job — SiliMap Distribution

C.1 The Scanning Procedure

The scanning procedure is implemented by the `SiliMapScanMod` module. It consists of the following three major steps:

1. Determination of the map layer boundaries and binning
2. scanning of all layers defined in step 1.
3. Storing the result persistently

In practice one should employ the parallelised scanning procedure as documented in section C.2. In this case the results of step 1 are stored persistently and serve as input for numerous scan jobs which run layer scans in parallel. The results are then concatenated to a valid SiliMap by an automatically generated script.

Considering the interface of `G3XIntegrator`, the layer boundaries can be chosen arbitrarily (e.g. equidistant). However, it proves useful to chose them in a way suitable for pattern recognition and track fitting. This basically means that the map binning structure, while arbitrary in principle, should reflect the organisation of the silicon detector. In R the active components are organised in well defined layers. In the offline software numbering scheme there are eight layers, numbered from 0 to 7. Layer 0 is also known as L00. Layers 1-5 constitute the SVX II and layers 6 and 7 the ISL. But the radial structure of the real detector is more granular. For instance layer 6 represents the whole central part of the ISL which actually consists of three barrels. The central one has a larger average radius than the outer barrels. In addition, all barrels in the silicon detector are organised in an even number of ϕ-wedges, the

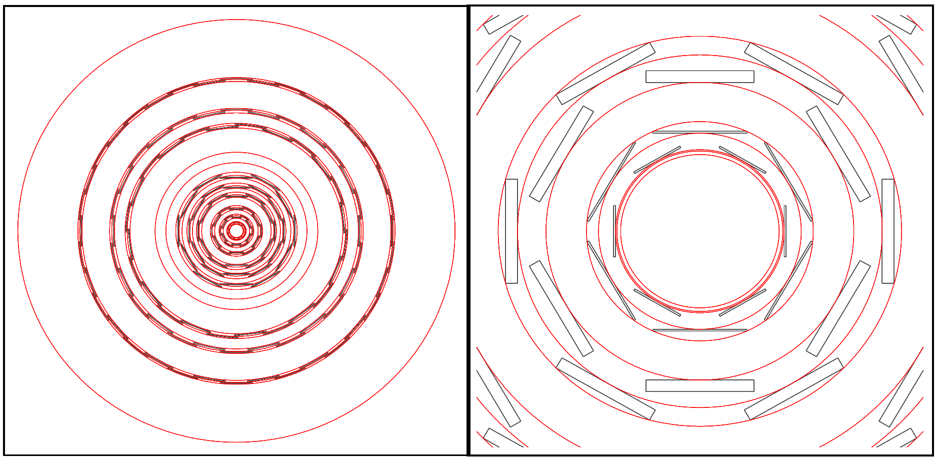


Figure C.1: R-ϕ-view of half ladders and the layer boundaries as determined by `SiliMapScanMod`. The black rectangles represent half ladders and the red circles correspond to the inner boundaries of the layers. The right hand side shows a zoomed view of L00 and the first SVX II layer. Note that the thickness of the SVX II ladders is derived from the hybrids and not the silicon sensors.

odd wedges having a different average radius then the even ones in the same layer[1]. Track fitters obviously require silicon hit information. In order to improve the timing performance of the material integration the information provided by the hits should not be ignored. Hits know about the half ladder they belong to and consequently about the layer and the ϕ-wedge. What is needed then is an unambiguous mapping of half ladders to material map layers.

The *`SiliMapScanMod::createLayerSpecs()`* method guarantees this unique mapping: using the information provided by `CdfHalfLadderSet` the minimum and maximum radius of each half ladder is computed. Next it is checked whether this pair of boundaries already occurred. Equality is checked using the finite accuracy from the symbol `SILIMAPSCAN_RANGE_EPSILON` which is defined in `SiliMapScanMod.hh`. After removing boundary overlaps in the resulting set of layers and adding a layer corresponding to the port card region the method returns. What remains to be done is filling the gaps between the layers derived from the half ladders. This is done in the *`SiliMapScanMod::prepareScan()`* method. Small gaps are filled by stretching the layer boundaries. Smallness is defined by the symbol `SILIMAPSCAN_RANGE_FILLGAP`. The remaining larger gaps are handled by introducing one map layer for each of them. This concludes the dynamic specification of layer boundaries. There are two special

[1] Whether the odd or even wedges have larger radii depends on the layer.

layers not defined by this procedure: the inner and outermost map layers, namely the beampipe and the COT inner cylinder. They are not supposed to be scanned using `G3XIntegrator` because they both can be represented by one-bin layers with material properties known from the corresponding geometry objects. They are anyway added to the layer specification list but immediately flagged as already scanned. Figure C.1 illustrates the results of the layer determination procedure.

To completely define a map layer it is necessary to specify the binning in ϕ and z. This is done in the method *`SiliMapScanMod::scanLayer()`*. The binning resolution can be configured via talk-to. A resolution of $\sim$2 mm has proven to be good choice. Given a z coverage of -100 cm to 100 cm, this leads to 999 z-bins in all layers[2]. The same resolution is desired in r-ϕ. This yields a different number of ϕ-bins in each layer. Additional constraints on the number ϕ-bins were introduced for various reasons: validation has shown that it is necessary to increase the r-ϕ resolution in the inner layers because the deviation from a perfectly cylindrical geometry has a larger impact at small radii. Thus the minimum number of ϕ-bins was chosen to be 120. On the other hand, a lower resolution is affordable in the outer region toward the COT inner cylinder. To save scanning time and memory no more than 1000 ϕ-bins are

[2] The odd number ensures that there is always a bin in the centre of the detector which was required by an early implementation of the `MaterialIntegrator` interface to `SiliMap`. This requirement was meanwhile dropped, allowing future maps to have a different binning.

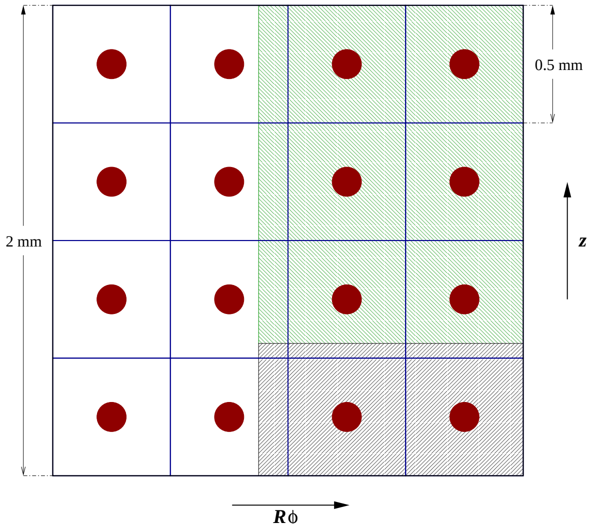


Figure C.2: A map bin of size 2mm is divided in 16 sub-bins by the scanning procedure. The red blobs mark the positions where the GEANT scans are performed. As indicated by the dashed areas on the right, there might be large differences between the material properties in the sub-bins.

allowed. Finally, the number of ϕ-bins is chosen to be a multiple of four to account for the detector's symmetry. This implicitly saves memory, as explained in section A.2.

After a map layer is fully specified, the method *SiliMapScanMod::scanBin()* is invoked for each bin in the layer. This method determines the average material properties in the bin using the G3XIntegrator. In general the scanning resolution is higher than the binning resolution. By default it is 0.5 mm for a binning resolution of 2 mm. This results in 16 GEANT scans for each bin which have to be averaged properly. This is illustrated in figure C.2.

For each sub-bin a GEANT scan is performed by *G3XIntegrator::integrate()* using a high energetic μ^-. This method returns a list of materials, each providing information about the constants $C_{dE/dx}$, I_0 and X_0 and the path length l of the trajectory in the material. Thus an averaging procedure is necessary also for the determination of the average sub-bin properties. Both averaging procedures are not trivial. One has to keep in mind that we are finally interested in the average integrated radiation length and energy loss. Consequently we have to integrate these quantities and not the material properties themselves. An arithmetic or weighted mean is not the correct choice for all material properties. The correct formulae for the mean values in a sub-bins are given below. They are derived from the path length-weighted mean of a particle's energy loss and integrated radiation length.

$$< C_{dE/dx} > \; = \; \frac{1}{\sum_{i=1}^{n} l_i} \cdot \sum_{i=1}^{n} l_i \cdot C_{dE/dx,i} \tag{1}$$

$$< I_0 > \; = \; \exp\left(\frac{1}{\sum_{i=1}^{n} l_i \cdot < C_{dE/dx} >} \cdot \sum_{i=1}^{n} l_i \cdot C_{dE/dx,i} \cdot \ln\left(I_{0,i}\right) \right) \tag{2}$$

$$< X_0 > \; = \; \frac{\sum_{i=1}^{n} l_i}{\sum_{i=1}^{n} l_i / X_{0,i}} \tag{3}$$

The equations for the averaging of the sub-bin results are easily obtained from (1),(2) and (3) for $l_i = l_j \; \forall \; i, j$; i.e. by going from a path length-weighted mean to a simple arithmetic mean:

$$< C_{dE/dx} >_{tot} \; = \; \frac{1}{n} \cdot \sum_{i=1}^{n} C_{dE/dx,i} \tag{4}$$

$$< I_0 >_{tot} \; = \; \exp\left(\frac{1}{n \cdot < C_{dE/dx} >} \cdot \sum_{i=1}^{n} C_{dE/dx,i} \cdot \ln\left(I_{0,i}\right) \right) \tag{5}$$

$$< X_0 >_{tot} \; = \; \frac{n}{\sum_{i=1}^{n} l_i / X_{0,i}} \tag{6}$$

After all bins in one layer are scanned, the corresponding instance of SiliMapLayer becomes valid. It is either added to the SiliMap or stored persistently in case of parallel scan jobs.

C.2 Setting up a Scan Job

To create a new SiliMap one needs a work release with at least the `TrackingUserMods` package. If changes to the geometry description of the base release are required one also needs to add the corresponding `SiliconGeometry` package. It is then recommended to set the environment variable `SILIMAPSCAN=yes` to avoid building unneeded object files and binaries. The scanner executable is then built by invoking '`gmake nobin TrackingUserMods.tbin`'.

The next step is to create the scan job tar ball and some helper scripts. This is easily done by launching the `siliMapScan` executable with the tcl file provided by `TrackingUserMods` package:

```
./bin/$SRT_SUBDIR/siliMapScan \
./TrackingUserMods/test/silimap_caf_job.tcl
```

This creates a tar ball for submission to CAF, a directory where the map will finally be assembled and a script that will submit the scan jobs to CAF. Although the environment driven defaults should be alright in most cases, one might want to review the `caf_submit_scan` script before executing it.

In case the user has access to computing resources without a CAF-like interface she can also create example job submission scripts suitable to run the scans on a cluster with the PBS batch system. Four tcl-switches need to be set properly to that end:

- `CreatePBSJobFiles set t`
- `PBSQueue`: a queue with sufficient CPU resources (∼4h for 8 jobs)
- `PBSInputDir`: has to be seen by nodes, job tar ball has to go here
- `PBSOutputDir`: has to be seen by nodes, results go here

The PBS job scripts are called `pbs_submit_scan` and `silimap_scan_pbs`. After copying the job tar ball to the directory specified with the `PBSInputDir` switch, one simply launches the `pbs_submit_scan` script to trigger the scan. It is impossible to support all batch systems on the market in this way. But it should not be too hard to write job scripts for other systems based on the PBS examples.

After the scan jobs finished successfully all that remains to be done is to copy the output tar balls to the `silimap_construction_site` directory, change to that directory and finally execute the `assemble_map` script. This will concatenate the layer scans to form a valid SiliMap in ASCII format and convert it to the other two supported formats, namely `StorableObject` and plain Root tree format. Note that the `assemble_map` scripts needs the scan executable for conversion and expects it at `../bin/$SRT_SUBDIR/siliMapScan`.

C.3 SiliMap Distribution

The `SiliMap` is distributed via the `silimap` UPS product that gets frozen with a given cdfsoft2 release. This ensures the correspondence of the detector description in the `SiliconGeometry` package and the `SiliMap` content. However, the `SiliMap` is technically independent of `SiliconGeometry` after the scan was performed. This makes it possible to use a map derived from a more recent, improved detector description while sticking to the same release concerning the reconstruction code. This is as simple as doing 'setup silimap <new version>' before launching an executable. Besides a `README` file, a valid silimap product contains the following directories:

- `ascii/` – containing the map in compresses plain text format
- `storable/` – containing the map in `StorableObject` format
- `tcl/` – containing the tcl files use to produce the map
- `tree/` – containing the map in Root tree format
- `ups/` – containing the table file

The directories again reside below `silimap/<version>/NULL`, denoting the product name, the version and the flavour of the product, respectively. The `<version>` should reflect the cdfsoft2 release that was used to create the silimap. E.g. when release 5.3.1 was used the version would be `v5_3_1`.

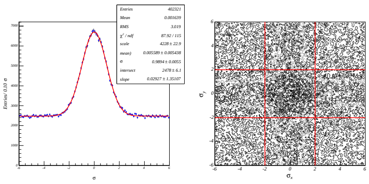
The first release the shipped with silimap included was integration release cdfsoft2 5.3.1int1. In case you want to use SiliMap with older releases, it is strongly recommended you ask the local CDF software administrator to install the silimap UPD product[3]. The SiliMap UPD product is the *only* recommended way of distributing SiliMap.

Once the silimap product is installed SiliMap is made available for offline analysis by simply doing 'setup silimap <new version>'. For releases that ship with the silimap dependency this is already triggered by the corresponding 'setup cdfsoft2 <version>'.

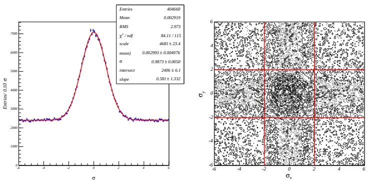
[3] If you do not know who your local CDF software administrator is, it is probably you.

Appendix D

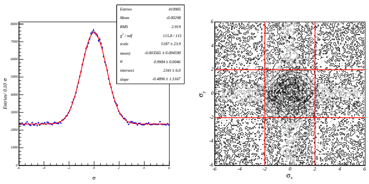
Toy Monte Carlo Results



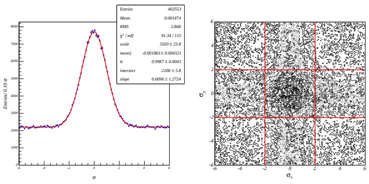
$sb = 3/4$, $rc = 0.2$,
evidence = 9.4 σ



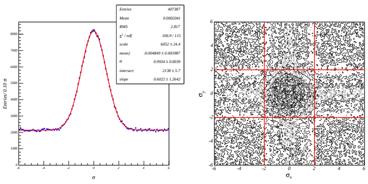
$sb = 3/4$, $rc = 0.6$,
evidence = 27.7 σ



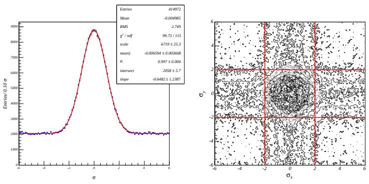
$sb = 3/4$, $rc = 1.0$,
evidence = 46.2 σ



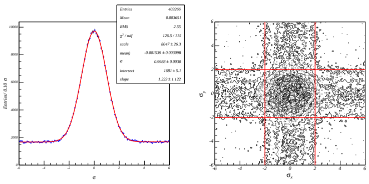
$sb = 2/3$, $rc = 0.2$,
evidence = 7.8 σ



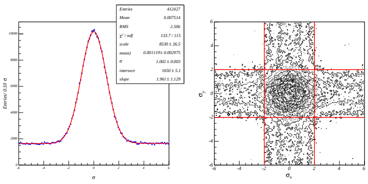
$sb = 2/3$, $rc = 0.6$,
evidence = 28.2 σ



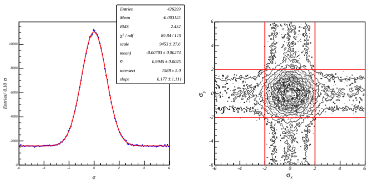
$sb = 2/3$, $rc = 1.0$,
evidence = 50.0 σ



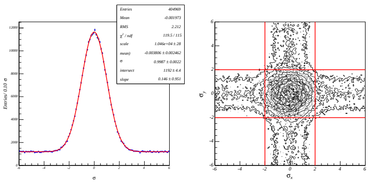
$sb = 1/2$, $rc = 0.2$,
evidence = 1.2 σ



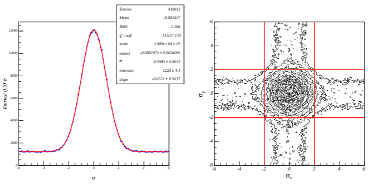
$sb = 1/2$, $rc = 0.6$,
evidence = 22.8 σ



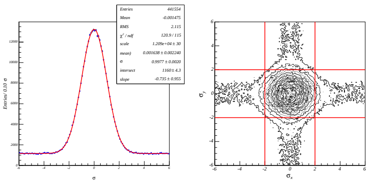
$sb = 1/2$, $rc = 1.0$,
evidence = 46.9 σ



$sb = 1/3$, $rc = 0.2$,
evidence = -6.4 σ



$sb = 1/3$, $rc = 0.6$,
evidence = 12.8 σ



$sb = 1/3$, $rc = 1.0$,
evidence = 45.1 σ

Bibliography

[1] S. Eidelman, K.G. Hayes, K.A. Olive, M. Aguilar-Benitez, C. Amsler, D. Asner, K.S. Babu, R.M. Barnett, J. Beringer, P.R. Burchat, C.D. Carone, C. Caso, G. Conforto, O. Dahl, G. D'Ambrosio, M. Doser, J.L. Feng, T. Gherghetta, L. Gibbons, M. Goodman, C. Grab, D.E. Groom, A. Gurtu, K. Hagiwara, J.J. Hernández-Rey, K. Hikasa, K. Honscheid, H. Jawahery, C. Kolda, Kwon Y., M.L. Mangano, A.V. Manohar, J. March-Russell, A. Masoni, R. Miquel, K. Mönig, H. Murayama, K. Nakamura, S. Navas, L. Pape, C. Patrignani, A. Piepke, G. Raffelt, M. Roos, M. Tanabashi, J. Terning, N.A. Törnqvist, T.G. Trippe, P. Vogel, C.G. Wohl, R.L. Workman, W.-M. Yao, P.A. Zyla, B. Armstrong, P.S. Gee, G. Harper, K.S. Lugovsky, S.B. Lugovsky, V.S. Lugovsky, A. Rom, M. Artuso, E. Barberio, M. Battaglia, H. Bichsel, O. Biebel, P. Bloch, R.N. Cahn, D. Casper, A. Cattai, R.S. Chivukula, G. Cowan, T. Damour, K. Desler, M.A. Dobbs, M. Drees, A. Edwards, D.A. Edwards, V.D. Elvira, J. Erler, V.V. Ezhela, W. Fetscher, B.D. Fields, B. Foster, D. Froidevaux, M. Fukugita, T.K. Gaisser, L. Garren, H.-J. Gerber, G. Gerbier, F.J. Gilman, H.E. Haber, C. Hagmann, J. Hewett, I. Hinchliffe, C.J. Hogan, G. Höhler, P. Igo-Kemenes, J.D. Jackson, K.F. Johnson, D. Karlen, B. Kayser, D. Kirkby, S.R. Klein, K. Kleinknecht, I.G. Knowles, P. Kreitz, Yu.V. Kuyanov, O. Lahav, P. Langacker, A. Liddle, L. Littenberg, D.M. Manley, A.D. Martin, M. Narain, P. Nason, Y. Nir, J.A. Peacock, H.R. Quinn, S. Raby, B.N. Ratcliff, E.A. Razuvaev, B. Renk, G. Rolandi, M.T. Ronan, L.J. Rosenberg, C.T. Sachrajda, Y. Sakai, A.I. Sanda, S. Sarkar, M. Schmitt, O. Schneider, D. Scott, W.G. Seligman, M.H. Shaevitz, T. Sjöstrand, G.F. Smoot, S. Spanier, H. Spieler, N.J.C. Spooner, M. Srednicki, A. Stahl, T. Stanev, M. Suzuki, N.P. Tkachenko, G.H. Trilling, G. Valencia, K. van Bibber, M.G. Vincter, D. Ward, B.R. Webber, M. Whalley, L. Wolfenstein, J. Womersley, C.L. Woody, O.V. Zenin, and R.-Y. Zhu. Review of Particle Physics. Physics Letters B, 592:1+, 2004.

[2] Bo Andersson, G. Gustafson, G. Ingelman, and T. Sjostrand. *PARTON FRAGMENTATION AND STRING DYNAMICS*. Phys. Rept., 97:31, 1983.

[3] Paolo Nason. *Heavy Quark Production*. Advanced Series on Directions in High

Energy Physics, 10, Heavy Flavours:588 pp., 1992. and references therein.

[4] E. Norrbin and T. Sjostrand. *Production and Hadronization of Heav Quarks*, 2000. hep-ph/0005110.

[5] S. K. Choi et al. *Observation of a New Narrow Charmonium State in Exclusive* $B^{\pm} \to K^{\pm}\pi^{+}\pi^{-} J/\psi$ *decays.* Phys. Rev. Lett., 91:262001, 2003.

[6] D. Acosta et al. *Observation of the Narrow State* $X(3872) \to J/\psi\,\pi^{+}\pi^{-}$ *in* $\bar{p}p$ *Collisions at* $\sqrt{s} = 1.96$ *TeV.* Phys. Rev. Lett., 93:072001, 2004.

[7] M. Feindt J. Heuser U. Kerzel. *Analysis of the quantum numbers of the X(3872)*, 2004. CDF/DOC/BOTTOM/CDFR/7311.

[8] F. Abe et al. (CDF Collaboration). *Evidence for Top Quark Production in* $\bar{p}p$ *Collisons at* $\sqrt{s}$*=1.8 TeV*, 1994. Phys. Rev. Lett. 73,225-231.

[9] T. Affolder et al. (CDF Collaboration). *Measurement of the Top Quark Mass with the Collider Detector at Fermilab*, 2001. Phys. Rev. D 63:032003.

[10] The CDF II Collaboration. *The CDF II Detector Technical Design Report*, 1996. FERMILAB-Pub-96/390-E.

[11] The CDF II Collaboration. *Proposal for the Enhancement of the CDF II Detector: An Inner Silicon Layer and A Time of Flight Detector*, 1998. Fermilab-Proposal 909.

[12] C. Ciobanu et al. *Online Track Processor for the CDF Upgrade*, 1999. IEEE Trans. Nucl. Sci., vol. 46, pp. 933-939.

[13] A. Bardi et al. *The CDF-II Online Silicon Vertex Tracker*, 2001. eConf C011127:THBT003, hep-ph/0112141.

[14] D. Acosta et al. (CDF 2 Collaboration). *Measurement of the Mass Difference m(D_s^+) - m(D^+) at CDF II*, 2003. Phys. Rev. D68, 072004.

[15] Bjarne Stroustrup. *The C++ Programming Language, 3rd Ed.*, 2003. Addison-Wesley ISBN 0201889544.

[16] C. Hays P. Tamburello A. Kotwal P. Wittich and R. Snider. *The COT Pattern Recognition Algorithm and Offline Code*, 2004. CDF/DOC/TRACKING/CDFR/6992.

[17] P. Azzi G. Busetto P. Gatti and A. Ribon. *Histogram Tracking in the COT*, 2001ls. CDF/DOC/TRACKING/CDFR/5562.

[18] K. Bloom and W.-M. Yao. *Outside-In Silicon Tracking at CDF*, 2002. CDF/DOC/TRACKING/CDFR/5991.

[19] S. Menzemer M. Feindt and K. Rinnert. *TrackingKal - A Tracking and Alignment Software Package for the CDFII Silicon Detector*, 2003. CDF/THESIS/TRACKING/PUBLIC/5968.

[20] S. Menzemer M. Feindt, C. Lecci and K. Rinnert. *Pre-Tracking PV z Finder*, 2002. CDF/DOC/TRACKING/PUBLIC/5988.

[21] Christopher Hays Yimei Huanga Ashutosh V. Kotwal Heather K. Gerberich Stephanie Menzemer Kurt Rinnert Claudia Lecci Matthew Herndon and Frederick D. Snider. *Inside-out Tracking at CDF*, 2004. NIMA 42855, Vol 538/1-3 pp 249-254.

[22] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. Transactions of the ASME–Journal of Basic Engineering, 82(Series D):35–45, 1960.

[23] The `TrackingKal` Package in CDF Run2 Software http://cdfkits.fnal.gov/CdfCode/source/TrackingKal/ .

[24] J. Boudreau C. Green P. Murat M. Paterno L. Sexton-Kennedy. *The CDF Geometry System for Run II Offline Software. A Guide for Subdetector Package Writers*, 2000. CDF/DOC/COMP_UPG/PUBLIC/5368.

[25] R. Brun, R. Hagelberg, M. Hansroul, and J. C. Lassalle. Geant: Simulation program for particle physics experiments. user guide and reference manual. CERN-DD-78-2-REV.

[26] The `SiliconGeometry` Package in CDF Run2 Software http://cdfkits.fnal.gov/CdfCode/source/SiliconGeometry/ .

[27] David Stuart. *Effects of Material in the CDF-II Silicon Detectors*, 2000. CDF/DOC/TRACKING/PUBLIC/5268.

[28] Paul Avery. *Applied Fitting Theory V. Track Fitting using the Kalman Filter*, 1992. CBX 92-39.

[29] Ken Bloom. *Getting Started with CDF Run 2 Offline*, 2004. CDF/DOC/COMP_UPG/CDFR/5294.

[30] Elizabeth Sexton-Kennedy and Pasha Murat. Commissioning the cdf offline software. ECONF, C0303241:THJT001, 2003.

[31] The CDF Run2 Software http://cdfcodebrowser.fnal.gov/CdfCode/ .

[32] Rene Brun and Fons Rademakers. *ROOT - An Object Oriented Data Analysis Framework*, 1997. Proceedings AIHENP'96 Workshop, Lausanne, Sep. 1996, Nucl. Inst. & Meth. in Phys. Res. A 389 (1997) 81-86. See also http://root.cern.ch/.

[33] Elena Gerchtein Mario Campanelli. *Calibration of the momentum scale for Kalman refitter using J/ψ events*, 2004. CDF/PHYS/BOTTOM/6905.

[34] Robyn Madrak. *Correcting the COT Covariance Matrix*, 2003. Talk at the COT workshop, Feb. 7 2003.

[35] Ben Brau David Stuart Matt Herndon. *A Study of L00 performace in 5.1.0*, 2003. CDF/DOC/TRACKING/PUBLIC/6720.

[36] The CDF Run2 tracking homepage http://www-cdf.fnal.gov/upgrades/computing/projects/reconstruction/tracking/tracking.html .

[37] The `L00AddAndRefit` interface in release 5.3.4 of CDF Run2 Software http://cdfkits.fnal.gov/CdfCode/source/TrackingUserHL/TrackingUserHL/TrackFitting/L00AddAndRefit.hh?v=5.3.4 .

[38] Ch.Paus C.Blocker, J.Boudreau. *Common Tools for B Physics Analyses in Run II at CDF*, 2002. CDF/DOC/BOTTOM/CDFR/5735.

[39] J. MARRINER. *SECONDARY VERTEX FIT WITH MASS AND POINTING CONSTRAINTS (CTVMFT)*, 1993. CDF/DOC/SEC_VTX/PUBLIC/1996.

[40] The `TrackingUserMods` Package in CDF Run2 Software http://cdfkits.fnal.gov/CdfCode/source/TrackingUserMods/ .

[41] The `TrackingSI` Package in CDF Run2 Software http://cdfkits.fnal.gov/CdfCode/source/TrackingSI/ .

[42] K. Anikeev G. Bauer I. Furic S. Gromoll A. Korn I. Kravchenko Ch. Paus A. Rakitin J. Tseng. *Calibration of Energy Loss and Magnetic Field using J/ψ Events in Run II*, 2002. CDF/DOC/BOTTOM/CDFR/5958.

[43] C. Paus G. Bauer, A. Korn. *Update on Calibration of Energy Loss and Magnetic Field using J/ψ Events in Run II*, 2003. CDF/DOC/BOTTOM/CDFR/6355.

Acknowledgements

First of all I would like to thank my supervisor Michael Feindt for the opportunity to work on this interesting project and to learn from his deep understanding of physics and statistical methods.

This work would not have been possible without the funding from the “Land Baden Württemberg” and the “Graduiertenkolleg für Hochenergiephysik und Teilchenastrophysik”.

Many members of the CDF 2 collaboration have helped making this project a success and I would like to thank them all.

I also like to thank my colleagues from the Karlsruhe CDF group. With their numerous questions they frequently put my CDF 2 software expertise to the test, often rendering me an expert in the first place.

To Patrick Schemitz I owe many insights into the principles of software development and the C++ programming language. He also started the `TrackingKal` project, thereby marking the crucial point during our software development: the time when we had to start from scratch.

Special thanks go to Stephanie Menzemer. Without her drive and ambition it would have been impossible to make the CDF 2 silicon tracking software a success. It was always a pleasure to work with her and I very much enjoyed our many fruitful discussions.