

**Projektarbeit im Rahmen des Projekts
MOBOT-III**

**Merkmalextraktion aus Sensordaten mit Hilfe der
Krümmungsfunktion**

Jörg Roth
Juni 1990

Betreuer: Thomas Knieriemen
AG v. Puttkamer

Die vorliegende Projektarbeit entstand im Rahmen des Mobot-III Projektes der AG von Puttkamer des Fachbereiches Informatik an der Universität Kaiserslautern.

Diese Arbeitsgruppe verfolgt das Ziel, einen *autonomen mobilen Roboter* zu bauen. Dazu sind zum einen Probleme der Konstruktion und des Aufbaus zu lösen, und zum anderen Programme zu erstellen, die der Sensordatenerfassung, Weiterverarbeitung und Steuerung dienen.

Zur Erstellung der Software wurde eine geeignete Simulationsumgebung entworfen, die es erlaubt, die Implementierung ohne die eigentliche Zielhardware durchzuführen. Es ist möglich, die Umwelt zu modellieren und Testdaten zu erzeugen, wie sie vom Roboter ermittelt sein könnten. Zum anderen können Fahrbefehle an die Simulationsumgebung gegeben werden, deren Ausführung dann simuliert wird.

Die vorliegende Arbeit wurde größtenteils mit Hilfe von Daten der Simulationsumgebung entwickelt und überprüft. Jedoch wurden auch, im Hinblick auf eine Tauglichkeit im realen Einsatz, Daten direkt vom Roboter gelesen und verarbeitet.

Inhalt

1. Einleitung und Motivation	4
1.1. Einleitung	4
1.2. Motivation	5
2. Spezifikaton	7
2.1. Spezifikation der Eingabe	7
2.2. Aufbau des Programms	8
2.3. Spezifikation der Ausgabe	10
3. Das Verfahren	12
3.1. Grundlagen	12
3.2. Die Krümmungsfunktion	13
3.3. Auswertung der Krümmungsfunktion	20
3.3.1. Extraktion von Sprungstellen	21
3.3.2. Extraktion von Offsets	22
3.3.3. Extraktion von Peaks	23
3.4. Generierung von Linien	26
3.5. Die Feinanalyse	28
4. Vergleich der verschiedenen Verfahren	32
4.1. Beschreibung des Testverfahrens	32
4.2. Testergebnisse	35
4.3. Auswertung der Ergebnisse	38

5. Benutzerschnittstelle	41
5.1. Menüs	41
5.2. Parameter	43
6. Implementierung	46
Anhang	48

1. Einleitung und Motivation

1.1. Einleitung

Die Aufgabe eines autonomen mobilen Roboters ist es, sich selbständig in einer zuerst unbekanntem Umgebung zu bewegen. Dabei soll er schrittweise seine Umgebung erkunden, um darin zu navigieren und gegebenenfalls einfache Aufgaben zu erfüllen.

Der in Kaiserslautern entwickelte Roboter Mobot-III soll genau diesen Bedingungen entsprechen, um eventuell Transport- oder Reinigungsaufgaben zu übernehmen. Um seine Umwelt wahrzunehmen, ist er mit vier Entfernungsmessern ausgerüstet. Diese drehen sich einmal in jeder Sekunde und vermessen den Raum. Geliefert wird eine Menge von Entfernungswerten, die aneinandergereiht einen Rundum-Blick der Umgebung bieten.

Die Auswertung der so gewonnenen Rohdaten ist streng hierarchisch geordnet. So gewinnt man mit jeder Komponente einen höhere, abstraktere Sicht auf die Umwelt. Die vorliegende Arbeit befaßt sich mit der niedrigsten Stufe dieser Hierarchie. Als Eingabe liegen die Entfernungsdaten der Sensoren vor, als Ausgabe liefert die Komponente eine erste vereinfachte Sicht, nämlich Linieninformationen.

Die Komponente, die diese Aufgabe bewerkstelligt wird hier *PFE* (primary feature extraction) genannt. Nachfolgende Einheiten können nun die Umwelt in einer einfacheren (zum Beispiel von Meßfehlern bereinigten) Sicht wahrnehmen. Sie kann nun kartographiert werden, um beispielsweise in ihr zu navigieren.

Die vorliegende Arbeit befasst sich mit einer neuen Strategie der Merkmalextraktion nämlich der *Extraktion mit Hilfe der Krümmungsfunktion*. Sie baut auf eine Diplomarbeit auf, die schon mehrere Verfahren beinhaltete. Auch das Rahmenprogramm stammt aus dieser Arbeit, so daß hier nur auf die neue Strategie sowie auf die wesentlichen Änderungen eingegangen wird. Nähere Informationen und das Listing des Rahmenprogramms kann man der Arbeit

entnehmen.

1.2. Motivation

In der angesprochenen Diplomarbeit wurden insgesamt fünf verschiedene Algorithmen zur Merkmalsextraktion vorgestellt und implementiert. Die Hauptmotivation, ein neues Verfahren einzuführen, kann man in drei Punkten angeben:

- Der hier neu vorgestellte Algorithmus ist sehr unkompliziert, das heißt er braucht weder viel Speicherplatz noch viel Rechenzeit. Hinzu kommt, daß in der bisherigen Konzeption der PFE-Komponente ein Teil der Berechnungen sowieso schon angefertigt werden muß, da er für andere Ausgaben benutzt wird. Das Verfahren ist ein *sukzessives Verfahren*, das heißt es benötigt nicht eine komplette Umdrehung der Sensoren, um mit der Berechnung zu beginnen. Im Gegensatz dazu stehen die totalen Verfahren, die erst eine vollständige Aufnahme einlesen, um diese dann zu analysieren. Der Algorithmus kann später auf der realen Maschine so implementiert werden, daß nach einer gewissen Drehung der Sensoren schon Linien an die nachfolgenden Komponenten übergeben werden. Dies begünstigt eine Verarbeitung im Echtzeitbetrieb.
- Der Algorithmus liefert zusätzliche Informationen ohne Mehraufwand. Damit kann die auszugebende Karte wesentlich aussagekräftiger gestaltet werden. So fällt bei der Berechnung zweier Linien, die in einer Ecke zusammenstoßen der Eckenwinkel ab, ohne daß hier aufwendige Berechnungen über trigonometrische Funktionen ausgeführt werden müssen.
- Überdies hinaus kann der Algorithmus völlig neue Merkmale erkennen. Während sich die bisherigen Verfahren auf das Erkennen von Strecken beschränkten, können hier gekrümmte Linien erkannt werden. Bisher wurden gekrümmte Linien durch eine Menge kleiner

Strecken approximiert. Das hatte jedoch zur Folge, daß die Datenmenge stark anstieg. Zusätzlich können Komponenten zur Objekterkennung gekrümmte Linien viel leichter bestimmten Gegenständen zuordnen. So kann man beispielsweise manche Objekte allein schon daran erkennen, daß sie im wesentlichen aus einer gekrümmten Oberfläche bestehen (z.B. Papierkorb oder Säule).

2. Spezifikation

2.1. Spezifikation der Eingabe

Die Eingabe der PFE-Komponente ist eine Menge von Punktinformationen, der sogenannte *Laserscan*. Für jeden Punkt werden folgende Werte übermittelt:

- Entfernung:
wie weit ist der gemessene Punkt vom Zentrum des Entfernungsmessers entfernt
- Drehwinkel:
welchen Winkel hatte der Sensor gemessen zur Fahrtrichtung des Roboters
- Neigungswinkel:
wie war der Sensor zur Waagerechten geneigt.

Zusätzlich werden noch Konfigurationsdaten des Lasers geliefert. Diese sind:

- Systemzeit und Positionszähler
- Mobotposition (Z-Koordinate entspricht der Höhe des Lasers)
- Orientierung des Mobots
- Anzahl der Sensoren
- Anzahl der Punkte pro Aufnahme

Als Eingabe kann die Komponente Laserscans aus der Simulationsumgebung einlesen, sowie reale Scans, die vom Mobot-III berechnet wurden.

2.2. Aufbau des Programms

PFE besteht aus folgenden Komponenten:

- Preprocessing:

hier werden die Daten einem Filter unterworfen, der extreme Werte eliminiert und Punkte, die stark von den Nachbarn abweichen einordnet. Dadurch wird die Qualität der Aufnahme erheblich verbessert und Meßfehler ausgeglichen. Zusätzlich werden in dieser Komponente die kartesischen Koordinaten der auf den Boden projizierten Punkte berechnet. Die so erzeugte Datenstruktur wird *Radarshot* genannt. Man gewinnt hier eine 2-dimensionale Sicht auf die Daten. Zur Liniengenerierung reicht das völlig aus, man wird aber später wieder die Höheninformation hinzufügen. Auf die genaue Darstellung wird jedoch verzichtet, sondern man ordnet die Höhen in entsprechende Höhenintervalle ein. Das erleichtert den nachgeschalteten Programmen den Vergleich generierter Linien.

- Histogrammgenerierung:

hier werden Punktpaare betrachtet, die einen bestimmten Winkel zum Entfernungsmesser bilden. Man berechnet deren Winkel zueinander im lokalen Koordinatensystem. Häufig vorkommende Winkel lassen auf Raumbegrenzungen schließen. So gewinnt man eine Vorzugsrichtung. Ermittelt wird sie, indem man Peaks aus der Funktion auswählt, die jedem Winkel die Häufigkeit zuordnet, mit dem er auftritt. Zusätzlich wird hier die sogenannte *Krümmungsfunktion* berechnet, ein integraler Bestandteil des Verfahrens.

- Line/Clustergenerierung:

jetzt werden die Punkte betrachtet und versucht, mit Hilfe der Krümmungsfunktion eine Annäherung durch Linien, Cluster (kleine Punktgruppen) und Kreissegmente durchzuführen. Zusätzlich werden die so gewonnenen Objekte mit Eigenschaften versehen. So wird beispielsweise abgelegt, ob eine Linie eine andere verdeckt oder ob sie verdeckt wird. Die generierte Datenstruktur wird *Line/Clustermap* genannt.

Histogramme und die generierten Linien und Cluster bilden die Ausgabe der PFE-Komponente, die sogenannte *Radarmap*.

In der ersten PFE-Version waren die Histogramm- und die Line/Clu-

stergenerierung unabhängig, das heißt sie hätten parallel laufen können. Deshalb waren auch für die Zielhardware hierfür zwei verschiedene Prozessoren vorgesehen. Für das neue Verfahren gilt das nicht. Hier wurde die Histogrammgenerierung leicht modifiziert, so daß ein Teil der Berechnung, nämlich die der Krümmungsfunktion, schon hier erfolgen kann, ohne daß dies in die Rechenzeit merklich einfließt.

Der folgende Plan zeigt die Komponenten sowie deren Datenfluß. Dabei stehen die runden Symbole für die Daten, und die Rechtecke für die verarbeitenden Komponenten.

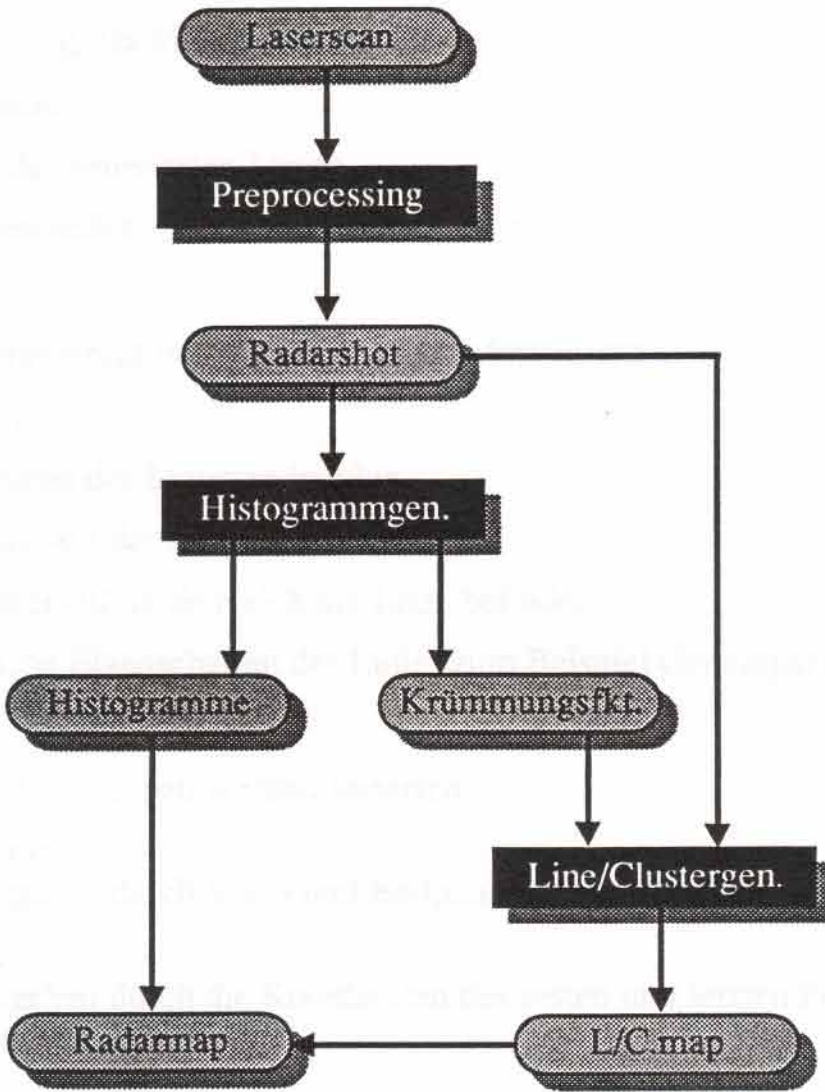


Bild 2.2.a: Anordnung der Komponenten in PFE

2.2. Spezifikation der Ausgabe

Wie schon erwähnt, besteht die Ausgabe von PFE im wesentlichen aus einer Folge von Linien. Da zu den bestehenden Verfahren neue Informationen hinzugekommen sind, wurde die Radarmap um einige Einträge erweitert. Dort findet man folgende Informationen:

- Systemzeit und Positionszähler
- Position des Mobots
- Orientierung des Mobots
- Histogramme
- Anzahl der generierten Linien
- die Linien selbst

Die Linieneinträge umfassen folgende Informationen:

- Linientyp
- Koordinaten der Linienendpunkte
- Eigenschaften der Linienendpunkte
- Höhenintervall, in dem sich die Linie befindet
- Zusätzliche Eigenschaften der Linie (zum Beispiel Genauigkeit)

Folgende Linientypen werden generiert:

- Objektlinie:
gegeben durch Start- und Endpunkt der Linie
- Cluster:
gegeben durch die Koordinaten des ersten und letzten Punktes der Punktgruppe
- Kreissegment:
gegeben durch Start- und Endpunkt des Segmentes sowie durch Radius, Öffnungswinkel und Koordinaten des Zentrums

Folgende Eigenschaften sind für Linienendpunkte vorgesehen:

- verdeckt
- verdeckend
- konvex
- konkav
- Bodenpunkt
- Übergang in ein anderes Höhenintervall

Ist die Eigenschaft konvex oder konkav, dann wird außerdem der Winkel abgelegt, der zur nächsten Linie gebildet wird. In bestimmten Fällen kann dieser Winkel jedoch nicht berechnet werden. Dann steht dort 0° , da dieser Wert sonst nicht vorkommen kann.

Spezifikation

3. Das Verfahren

3.1. Grundlagen

Vom Wesen her ist die Merkmalsextraktion mit Hilfe der Krümmungsfunktion ein statistisches Verfahren. Dies bedeutet, daß bestimmte Heuristiken angewendet werden, um Aussagen über Merkmale zu treffen. Der Vorteil dieser Vorgehensweise ist, daß der Aufwand wesentlich geringer ist, als wenn man eine mathematisch exakte Lösung anstrebt. Dies bringt jedoch eine gelegentliche Abweichung vom optimalen Ergebnis mit sich.

Wie wir später noch sehen werden, reicht deswegen in der Regel die Auswertung der Krümmungsfunktion nicht aus, um zuverlässige Ergebnisse zu erzielen. Da aber die extrahierten Merkmale im Grunde schon sehr nah an der Realität liegen, liegt es nahe, mit einem Filteralgorithmus die Merkmale zu überprüfen, und in den seltenen Fällen, in denen das Verfahren fehlschlägt eine andere Strategie anzuwenden.

Dadurch, daß die Auswertung der Krümmungsfunktion im wesentlichen eine korrekte Einteilung vornimmt, kann der Filteralgorithmus sich darauf beschränken das generierte Merkmal grob zu überprüfen. Deswegen ist dieser Algorithmus auch relativ einfach gehalten und benötigt keine nennenswerte Rechenzeit. Die Robustheit des Verfahrens steigt aber beträchtlich.

Begünstigend für das zeitliche Verhalten ist, daß das Verfahren ein sukzessives Verfahren ist. Insbesondere im Hinblick auf Echtzeitanwendungen sind nur solche Verfahren interessant, da diese sich auch für parallele Rechnerarchitekturen eignen. Das Verfahren kann derart aufgeteilt werden, daß man für verschiedene Verarbeitungsschritte separate Prozessoren vorsieht. Jeder Prozessor gibt dann mit einem Zeitversatz Daten an den nächsten weiter, so daß nach einem gewissen Zeitpunkt alle Prozessoren eine Pipeline-Verarbeitung vornehmen. Da der Entfernungsmesser permanent Daten liefert, bleibt dieser Zustand erhalten, womit ein günstiges Zeitverhalten hervorgerufen wird.

3.2. Die Krümmungsfunktion

Die Krümmungsfunktion gibt den Verlauf der Winkeländerung in jedem Punkt wieder. In Bild 3.2.a ist dargestellt, wie sie gewonnen wird:

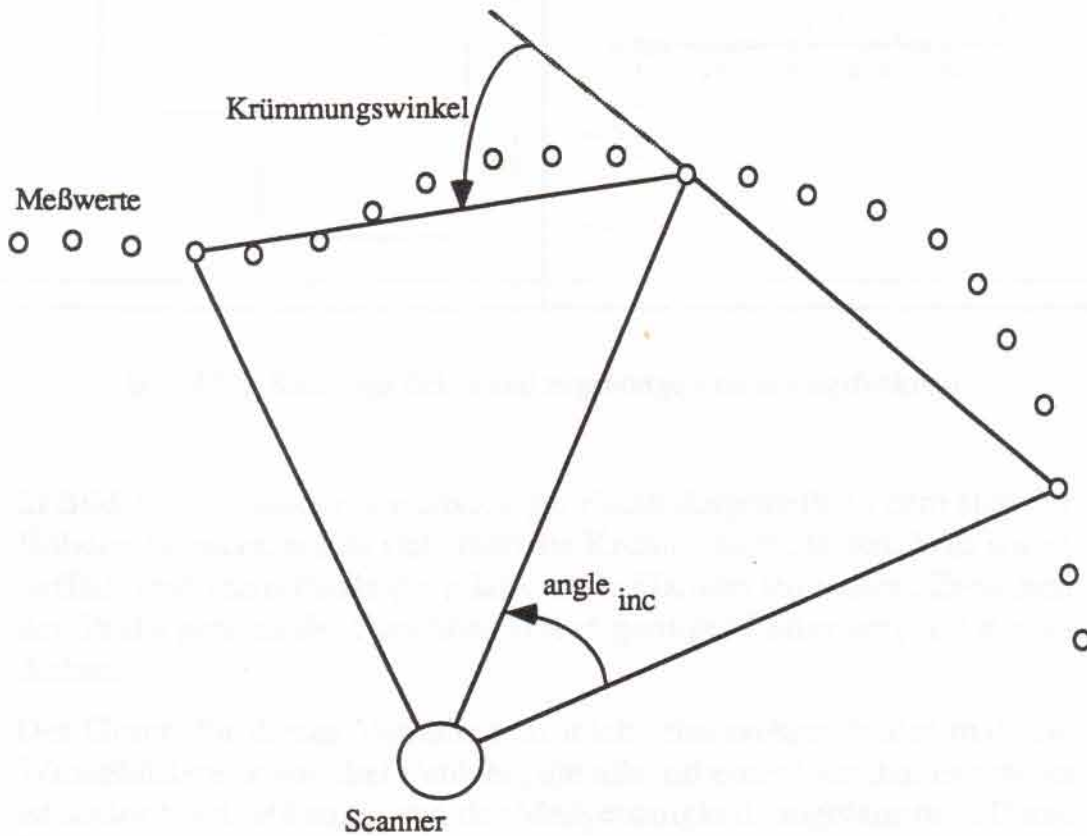


Bild 3.2.a: Gewinnung der Krümmungsfunktion

Um Meßtoleranzen auszugleichen, werden Tripel von Meßpunkten genommen, die einen gewissen Winkelversatz (angle_{inc}) haben. Der Winkel, der durch die beiden gedachten Linien eingeschlossen wird (Krümmungswinkel), ist der Funktionswert für den mittleren Punkt. Wendet man dieses Verfahren auf alle Punkte an, erhält man die komplette Krümmungsfunktion.

Der Verlauf dieser Funktion gibt jetzt Auskunft über Art und Anordnung der Merkmale im Entfernungsbild. Dies soll anhand einiger Beispiele erläutert werden.

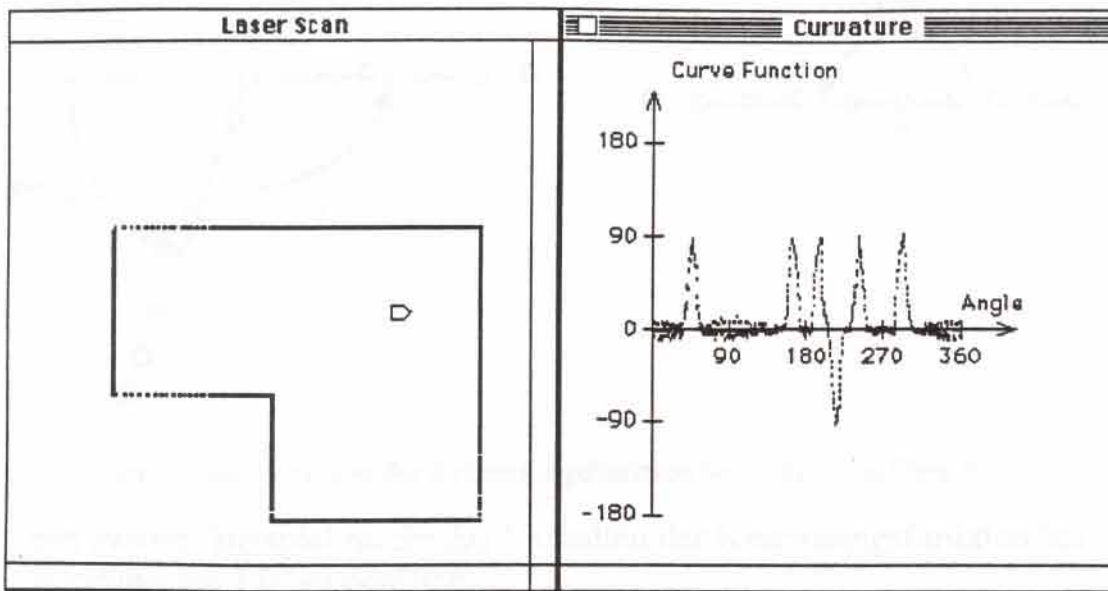


Bild 3.2.b: Raum mit Ecken und zugehörige Krümmungsfunktion

In Bild 3.2.b ist links ein sechseckiger Raum dargestellt, in dem sich der Roboter befindet, rechts sieht man die Krümmungsfunktion. Was sofort auffällt sind sechs *Peaks* die relativ steile Flanken aufweisen. Zwischen den Peaks pendelt der Funktionswert in geringer Entfernung um die x-Achse.

Der Grund für dieses Verhalten ist leicht einzusehen: bildet man die Winkeldifferenz von drei Punkten, die alle auf einer Geraden liegen, so ist dieser Wert, abhängig von der Meßgenauigkeit, ungefähr null. Diese Bereiche werden im folgenden *Nullbereiche* genannt.

Befindet sich jedoch der mittlere Punkt genau auf einer Ecke, so verlaufen die beiden Schenkel entlang der aneinanderstoßenden Wände. Damit entspricht der Wert der Krümmungsfunktion in diesem Punkt genau 180° minus dem Eckenwinkel. Um den Eckpunkt herum werden die Funktionswerte vom Betrag her kleiner, wodurch man die charakteristischen Peaks erhält.

In Bild 3.2.c ist das Verhalten der Krümmungsfunktion bei Ecken und Geraden nocheinmal zu sehen.

3.2. Die Krümmungsfunktion

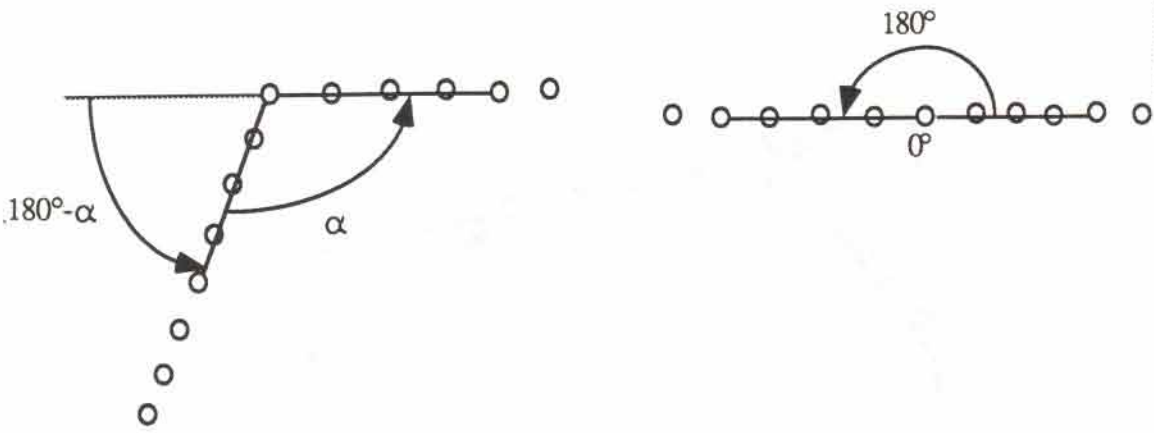


Bild 3.2.c: Verhalten der Krümmungsfunktion bei Ecken und Geraden

Ein zweites Beispiel macht das Verhalten der Krümmungsfunktion bei gekrümmten Linien deutlich.

Das Verfahren

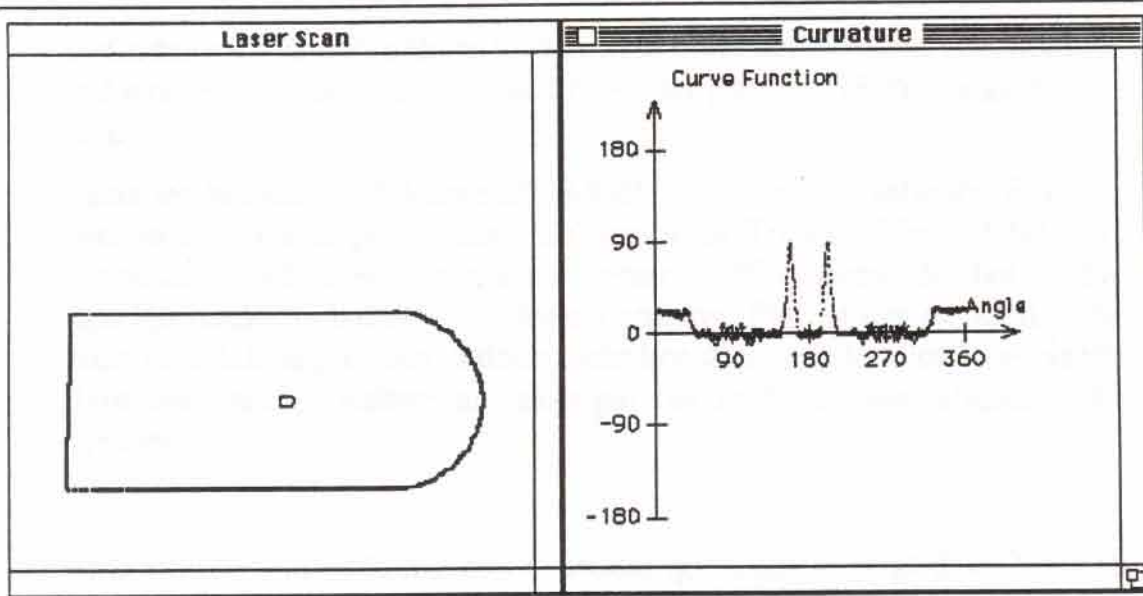


Bild 3.2.d: Raum mit gekrümmter Wand sowie zugehörige Krümmungsfunktion

Gut zu sehen sind hier wieder die beiden 90° -Peaks, die die beiden Ecken links im Raum wiedergeben. Zusätzlich befindet sich aber um die 0° -Marke ein Bereich, in dem der Wert der Krümmungsfunktion einen konstanten Wert annimmt. Dieser Bereich repräsentiert die gekrümmte Linie an der rechten Zimmerseite.

Wie leicht einzusehen ist, bildet sich dieser konstante Bereich, der im folgenden *Offset* genannt wird, immer aus, wenn eine gekrümmte

3.2. Die Krümmungsfunktion

Linie vorliegt, wie in Bild 3.2.e dargestellt wird.

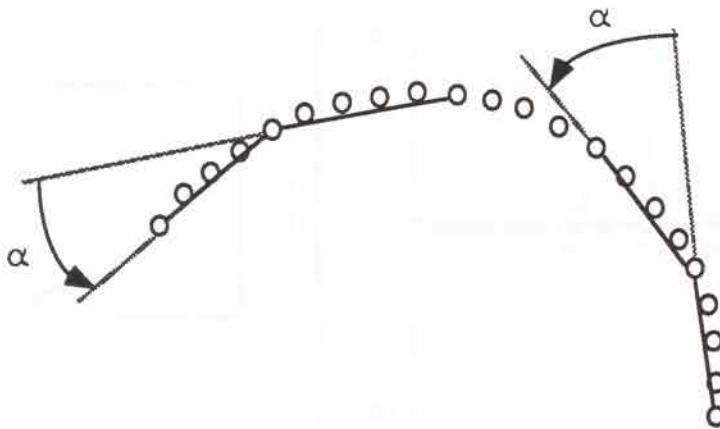


Bild 3.2.e: Verhalten der Krümmungsfunktion bei gekrümmten Linien

Befinden sich drei Punkte in immer dem gleichen Abstand voneinander auf einem Kreisumfang, so wird immer der gleiche Winkel eingeschlossen.

Genauer betrachtet, bilden sich jedoch nur in einem seltenen Fall die gleichen Abstände unter den Punkten eines Tripels. Dieser liegt vor, wenn sich der Entfernungsmesser genau im Mittelpunkt des betreffenden Kreissektors befindet. In jedem anderen Fall ist der Bereich zwar auch deutlich angehoben, jedoch nicht konstant, sondern leicht verzerrt. Trotzdem ist eine Krümmung aber gut aus der Krümmungsfunktion zu ermitteln.

Eine weitere Ausprägung der Krümmungsfunktion zeigt Bild 3.2.f, in dem ein Raum mit einer geöffneten Tür dargestellt ist.

Hier sind wieder gut die vier Peaks für die Raumecken sowie ein kleiner Peak für die Ecke zwischen Tür und der unteren Wand zu erkennen.

Der Bereich um circa 250° zeigt jedoch ein neues Verhalten. Dort springt die Krümmungsfunktion drei mal. Dieser Bereich kann man der Sprungkante zwischen geöffneter Tür und Wand zuordnen. Bei der Berechnung der Krümmungsfunktion wandert das Punktetripel auch über die Sprungstelle. Immer wenn ein neuer Punkt des Tripels die Stelle hinter sich gelassen hat, ändert sich sprunghaft auch der Wert der Krümmungsfunktion. Die drei Sprünge haben deshalb genau den

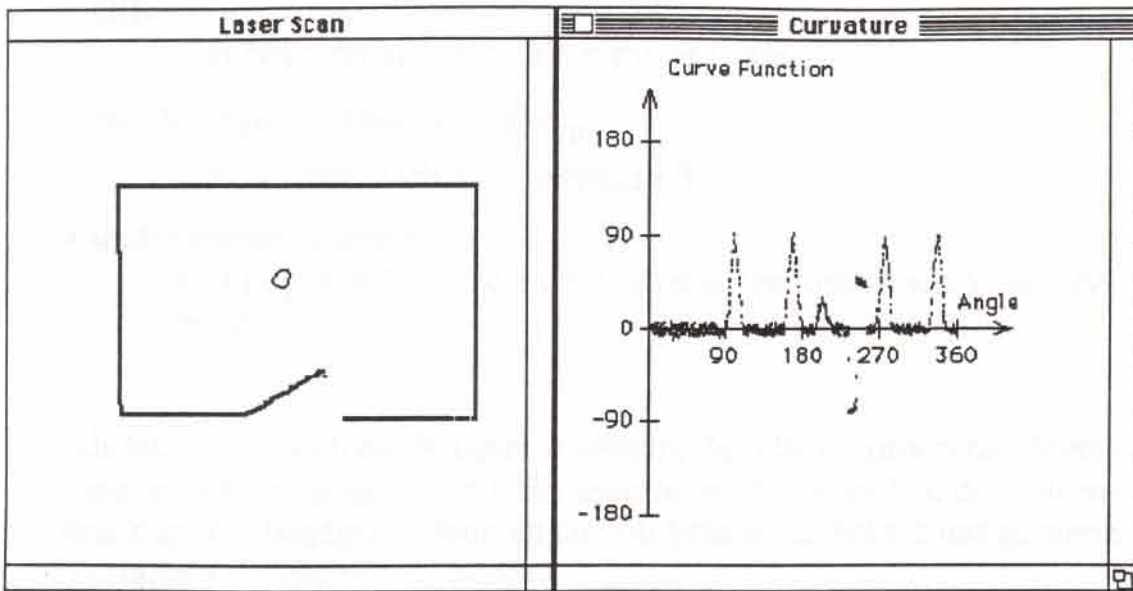


Bild 3.2.f: Raum mit Sprungstelle sowie zugehörige Krümmungsfunktion

Abstand angle_{inc} . Es ist leicht einzusehen, daß dieser Bereich kaum noch auswertbar ist. Aus diesem Grund wird er auch für alle weiteren Betrachtungen ausgeschlossen (was, wie wir später noch sehen werden, einige Nachteile mit sich bringt).

Das letzte vorkommende Merkmal in der Krümmungsfunktion tritt dann auf, wenn für bestimmte Winkel keine Entfernungswerte existieren, was beispielsweise dann der Fall ist, wenn die maximale Reichweite des Entfernungsmessers überschritten wurde. Da zur Berechnung der Krümmungsfunktion aber Werte im Bereich $\pm \text{angle}_{inc}$ nötig sind, kann die Krümmungsfunktion an den Rändern nicht berechnet werden. Dort erhält die Krümmungsfunktion den Wert *undefiniert*.

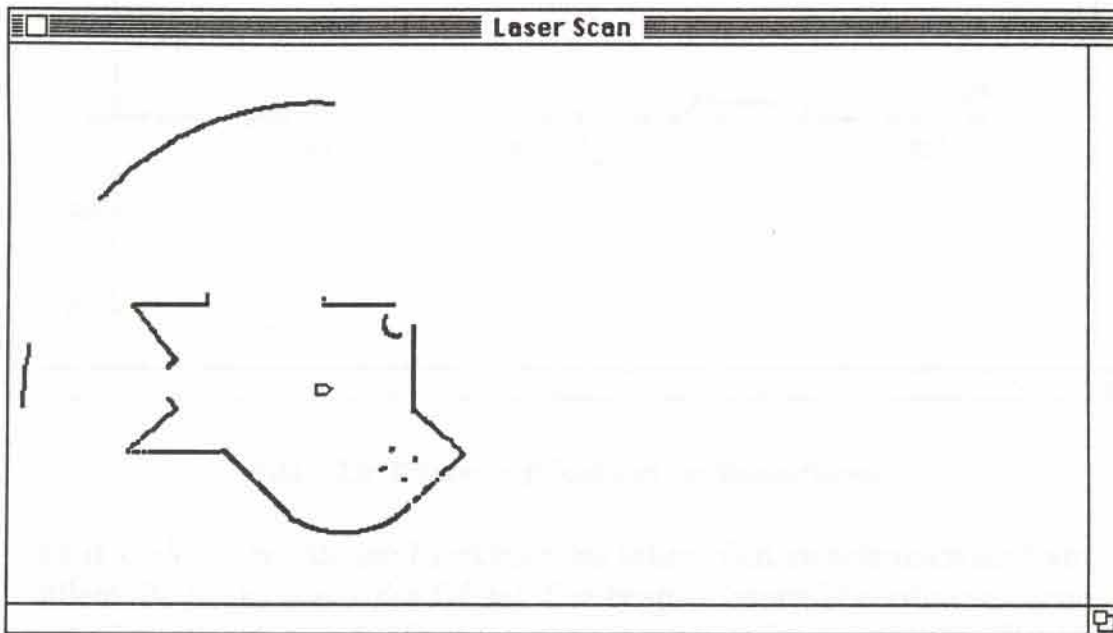
Zusammengefasst kann man also die Krümmungsfunktion auf folgende Merkmale reduzieren:

- Nullbereich:
hier befindet sich eine gerade Linie
- Peak:
hier befindet sich eine Ecke; die Höhe des Peaks gibt den Winkel der Ecke wieder

3.2. Die Krümmungsfunktion

- Offset:
hier befindet sich eine gekrümmte Linie
- drei Spünge im Abstand angle_{inc} :
hier befindet sich eine Sprungstelle
- undefinierter Bereich:
hier liegt ein Rand zu einem nicht aufgenommenen Winkelbereich

Als letztes sei noch ein Beispiel angeführt, das alle besprochenen Merkmale in sich vereinigt. Diese Beispielszene wird uns noch in den folgenden Kapiteln begegnen, denn an ihr soll Schritt für Schritt das gesamte Verfahren durchgespielt werden.

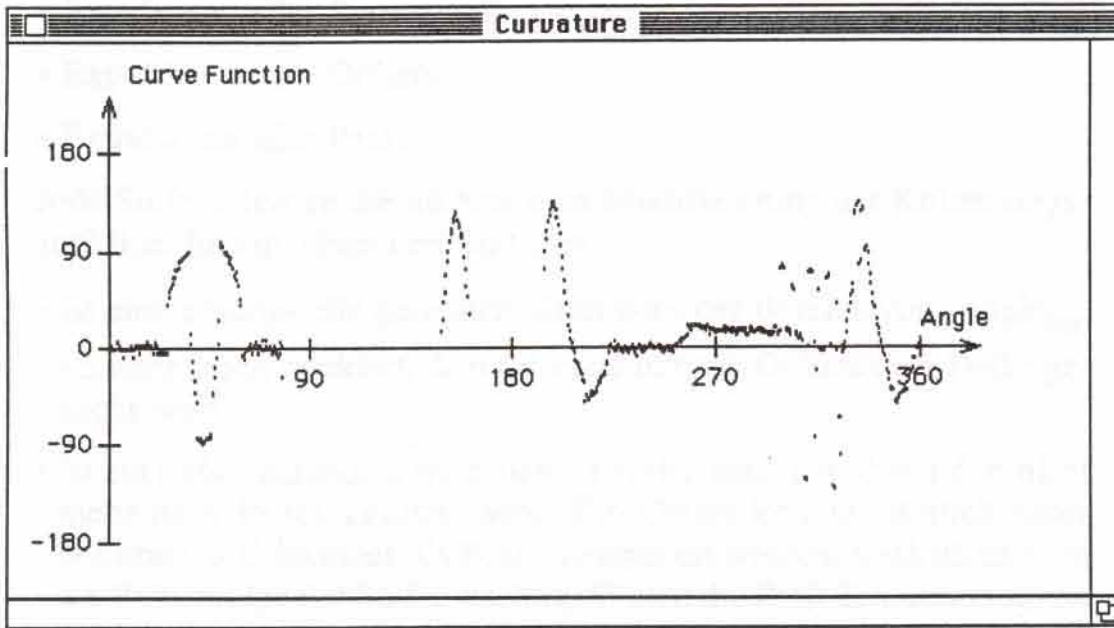


Das Verfahren

Bild 3.2.g: Entfernungsbild einer Beispielszene

Das Entfernungsbild wurde in einem Raum aufgenommen, der durch zwei Flügeltüren erreichbar ist. Rechts unten sieht man vier Punktgruppen. Diese könnten durch die Beine eines Stuhls entstanden sein. Oben rechts kann man einen runden Gegenstand sehen, beispielsweise ein Papierkorb. Bei der Aufnahme war der Laserstrahl des Entfernungsmessers leicht nach unten geneigt. Da der Raum an zwei Stellen geöffnet ist, wurde so der Boden erfasst, was zu den Kreisabschnitten oben und links führte.

Die zugehörige Krümmungsfunktion ist im folgenden Bild dargestellt.



Das Verfahren

Bild 3.2.h: Krümmungsfunktion der Beispielszene

Dort sind alle erwähnten Merkmale zu sehen. Gut zu erkennen sind vor allem die Peaks sowie der Offset. Die besprochenen Merkmale müssen jetzt "nur" noch aus der Funktion extrahiert werden, womit die Einteilung in Linien und Kurven im Prinzip schon vorgenommen ist.

3.3. Auswertung der Krümmungsfunktion

Obwohl das geschulte Auge die Merkmale aus der Krümmungsfunktion ohne Schwierigkeiten ermitteln kann, gestaltet sich dies auf algorithmischem Weg schwieriger als zuerst angenommen.

Der hier entwickelte Algorithmus fertigt die Analyse in drei Stufen an:

- Extrahieren aller Sprungstellen
- Extrahieren aller Offsets
- Extrahieren aller Peaks

Jede Stufe liefert an die nächste eine Modifizierung der Krümmungsfunktion. Im einzelnen bedeutet dies:

- Ist eine Sprungstelle gefunden, dann wird der Bereich von $\pm \text{angle}_{inc}$ als undefiniert markiert, damit dort nicht nach Offsets oder Peaks gesucht wird.
- Ist ein Offset gefunden, wird dieser Bereich markiert, damit dort nicht mehr nach Peaks gesucht wird. Ein Offset könnte nämlich unter bestimmten Umständen als Peak interpretiert werden. Deshalb ist auch die Reihenfolge der Stufen wichtig. Würde die Peak-Extraktion zuerst erfolgen, würden viele Offsets fälschlicherweise als Peak deklariert werden, da diese Stufe wesentlich gröber arbeitet.
- Ist ein Peak gefunden, so wird auch diese Stelle markiert. Automatisch wird der nun noch unmarkierte Bereich als Nullbereich angenommen.

Wird nun jede Stufe auf die Krümmungsfunktion angewendet, besteht sie nachher nur noch aus den gesuchten Merkmalen.

Im folgenden wird jede Stufe beschrieben. Insbesondere wird Wert auf die Algorithmen der Peak- und Offset-Extraktion gelegt, da sich hier unerwartet Probleme ergaben. Die Schwierigkeit liegt darin begründet, daß Peaks und Offsets sich nicht wesentlich unterscheiden.

3.3.1. Extraktion von Sprungstellen

Wie oben beschrieben, befindet sich genau dort in der Entfernungskarte eine Sprungstelle (verursacht beispielsweise durch eine geöffnete Tür), wo sich in der Krümmungsfunktion drei Sprünge im Abstand von genau $\text{angle}_{\text{inc}}$ befinden. Es wäre also denkbar, daß man die Krümmungsfunktion nach genau solchen Konstellationen durchsucht.

Dies erweist sich jedoch als nicht sehr vorteilhaft, da sich zum Beispiel zwei Sprungtripel überlappen können, oder was noch problematischer ist, sie können genau in einem Abstand von $\text{angle}_{\text{inc}}$ vorliegen, was eine umständliche Analyse nötig machen würde.

Aus diesem Grund wurde hier eine Analyse der Entfernungsdaten direkt vorgenommen. Die Entfernungsdaten müssen vorliegen um die Krümmungsfunktion zu berechnen, also kann man sie auch hinzuziehen, um Sprünge zu entdecken.

Das Verfahren

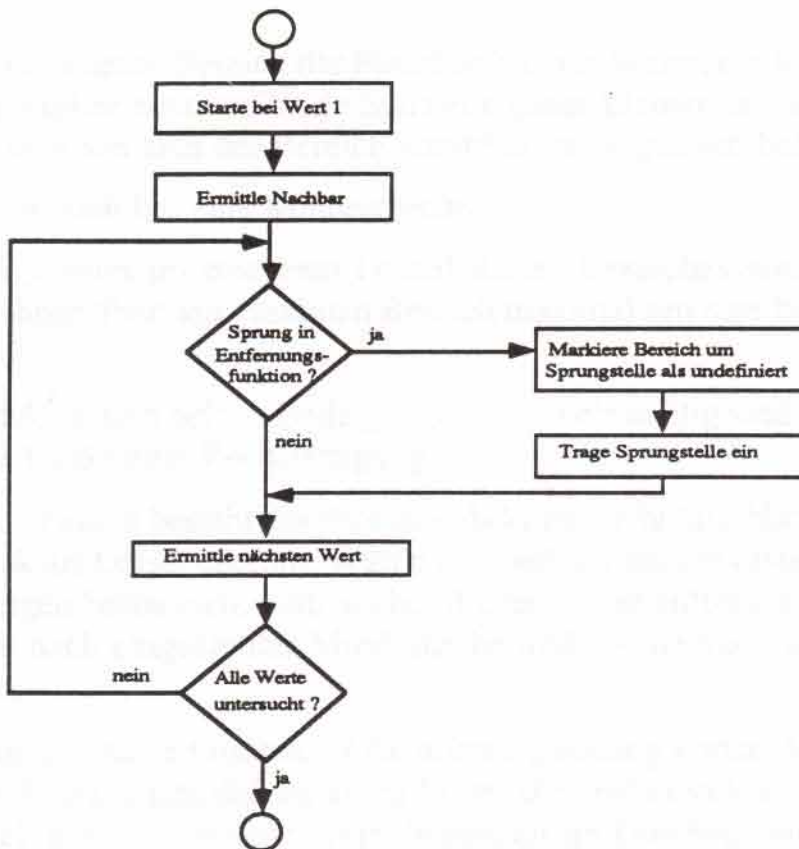


Bild 3.3.1.a: Flußdiagramm zur Sprungstellenextraktion

Die Analyse gestaltet sich als relativ einfach. Man vergleicht alle Entfernungswerte mit dem des jeweiligen Nachbarn. Überschreitet die Differenz eine voreinstellbare Schwelle, so wird ein Sprung angenommen. Die Schwelle richtet sich nach der Genauigkeit der vorliegenden Daten. Sie sollte deutlich höher liegen, als die zu erwartende Meßungenauigkeit des Entfernungsmessers. Damit sieht der Algorithmus wie in Bild 3.3.1.a aus.

3.3.2. Extraktion von Offsets

Der Algorithmus zu Erkennung von Offsets wurde besonders sensibel gestaltet, da sich beim Start dieser Stufe noch Peaks in der Funktion befinden. Es gilt diese von Offsets zu unterscheiden. Die dritte Stufe kann dann entsprechend einfacher gehalten werden.

Das Verfahren

Ein Offset wird als solcher erkannt, wenn folgende Bedingungen vorliegen:

- Ein geschlossener Bereich der Funktion hat nur Werte, die höher sind als eine vorher einzustellende Schwelle (oder kleiner der negativen Schwelle, wenn sich der Bereich komplett im Negativen befindet).
- Dieser Bereich hat eine Mindestbreite.
- Das Maximum im mittleren Drittel dieses Bereiches weicht vom Durchschnitt über den gesamten Bereich maximal um eine bestimmte Größe ab.

Während die ersten beiden Bedingungen sofort einsichtig sind, verdient der dritte Punkt eine Rechtfertigung.

Wie oben erwähnt besteht das Problem, daß man nicht fälschlicherweise einen Peak als Offset erkennt. Würde man jedoch nur die ersten beiden Bedingungen betrachten, könnte aber dieser Fehler auftreten, da diese Punkte je nach eingestellter Mindesthöhe und -breite auch auf Peaks zutreffen.

Genau um dies zu verhindern, ist die dritte Bedingung vorhanden. Peaks haben die Eigenschaft, daß sie in der Mitte (also insbesondere im mittleren Drittel) sehr viel größere Werte haben, als im Durchschnitt. Offsets dagegen haben bestenfalls an den Rändern stark überhöhte Werte. Dieser Fall tritt bei konvexen Krümmungen auf. Damit erhält man einen Algorithmus wie in Bild 3.3.2.a dargestellt.

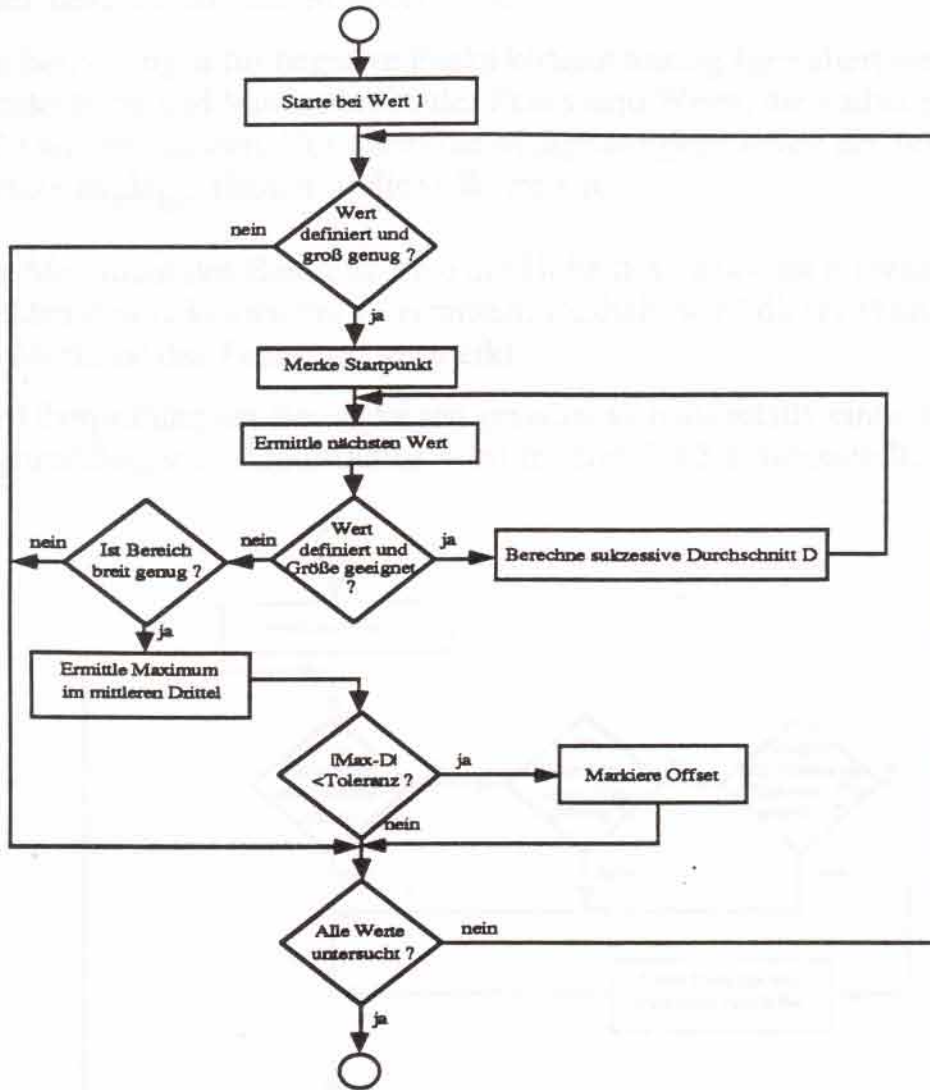


Bild 3.3.2.a: Flußdiagramm zur Offsetextraktion

3.3.3. Extraktion von Peaks

Ein Peak ist ein Bereich mit lokalem Maximum. Er besitzt folgenden Eigenschaften:

- Der Bereich liegt vollständig im Positiven.
- Das Maximum liegt genau in der Mitte des Bereichs.
- Das Maximum hat einen Mindestwert.

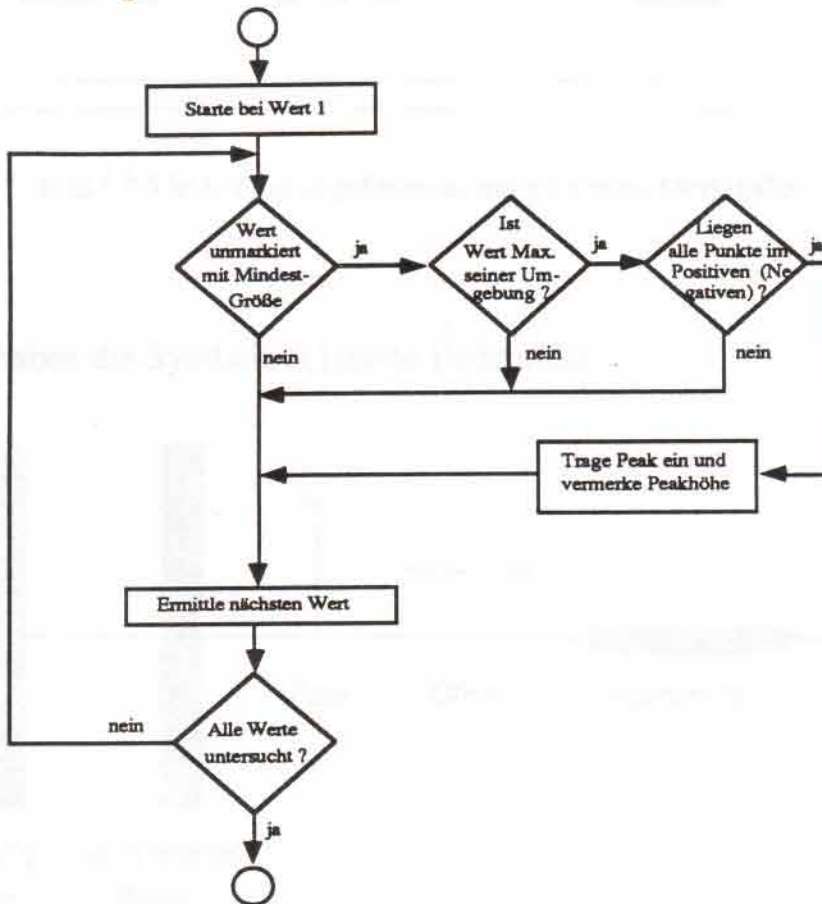
3.3.3. Extraktion von Peaks

- Der Bereich hat eine Mindestbreite.

Die Bedingungen für negative Peaks können analog formuliert werden. Mindesthöhe und Mindestbreite des Peaks sind Werte, die vorher eingestellt werden müssen. Vor allem die Meßgenauigkeit sowie der Winkelversatz angle_{inc} fließen in diese Werte ein.

Das Maximum des Bereichs, also die Höhe des Peaks, ist relevant, um nachher den Eckenwinkel zu ermitteln. Deshalb wird dieser Wert auch als Merkmal des Peaks mit vermerkt.

Die Überprüfung der Bedingungen gestaltet sich als relativ einfach. Der zugrundeliegende Algorithmus wird in Bild 3.3.3.a dargestellt.



Das Verfahren

Bild 3.3.3.a: Flußdiagramm zur Peakextraktion

Damit können nun alle relevanten Merkmale aus der Krümmungsfunktion ermittelt werden, und wir können zu unserer Beispielszene zurückkehren. Werden die drei Algorithmen in der beschriebenen Weise auf die Krümmungsfunktion angewendet, erhält man folgendes Bild:

3.3.3. Extraktion von Peaks

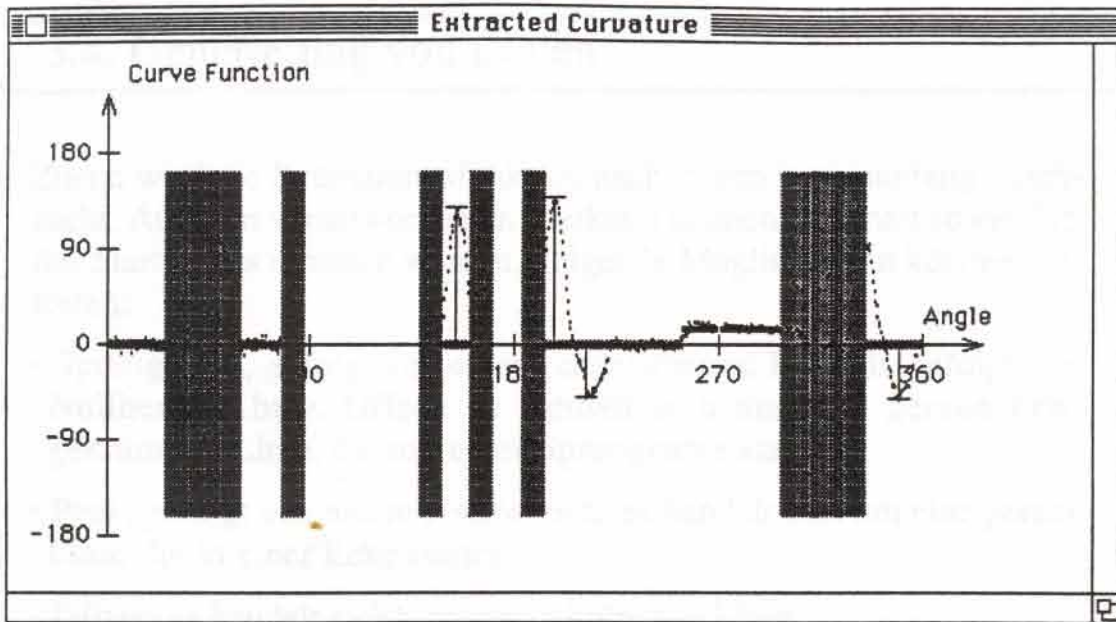


Bild 3.3.3.b: Krümmungsfunktion mit erkannten Merkmalen

Das Verfahren

Dabei haben die Symbole folgende Bedeutung:

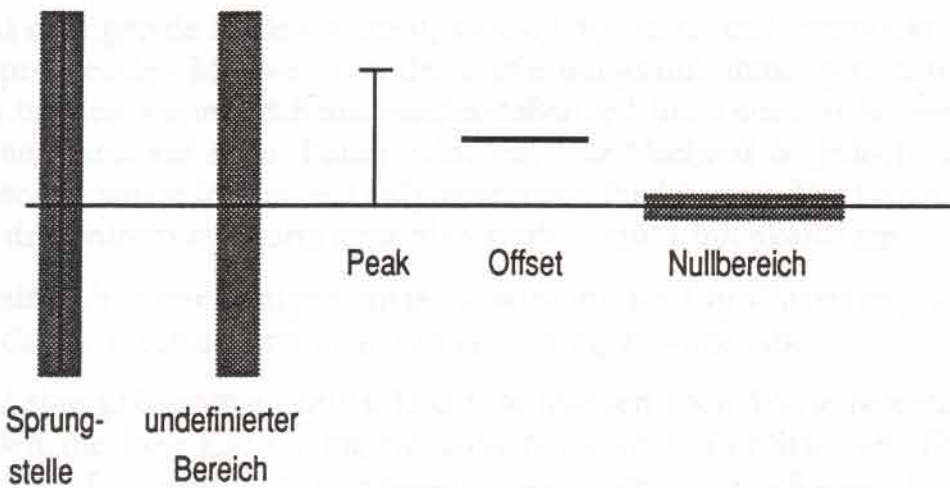


Bild 3.3.3.c: Legende zur Krümmungsfunktion

Damit ist die Einteilung vorgenommen. Das nächste Ziel ist nun, ausgehend von dieser Segmentierung Linien zu generieren.

3.3.3. Extraktion von Peaks

3.4. Generierung von Linien

Zuerst wird die Krümmungsfunktion nach einem Linienanfang durchsucht. Aus dem verantwortlichen Merkmal können Linienart sowie Typ des Startpunkts ermittelt werden. Folgende Möglichkeiten können auftreten:

- Sprungstelle, gefolgt von einem undefinierten Bereich, gefolgt von Nullbereich bzw. Offset: es handelt sich um eine gerade bzw. gekrümmte Linie, die mit einer Sprungkante startet.
- Peak, gefolgt von einem Nullbereich: es handelt sich um eine gerade Linie die in einer Ecke startet.
- Offset: es handelt sich um eine gekrümmte Linie.
- undefinierter Bereich, gefolgt von Nullbereich bzw. Offset: es handelt sich um eine gerade bzw. gekrümmte Linie, die an einen nicht aufgenommenen Bereich grenzt.

Entsprechend wird das Linienende ermittelt, und der Typ des Linienendpunktes gesetzt. So wird mit allen Linien fortgefahren, bis alle Meßwerte erfaßt sind.

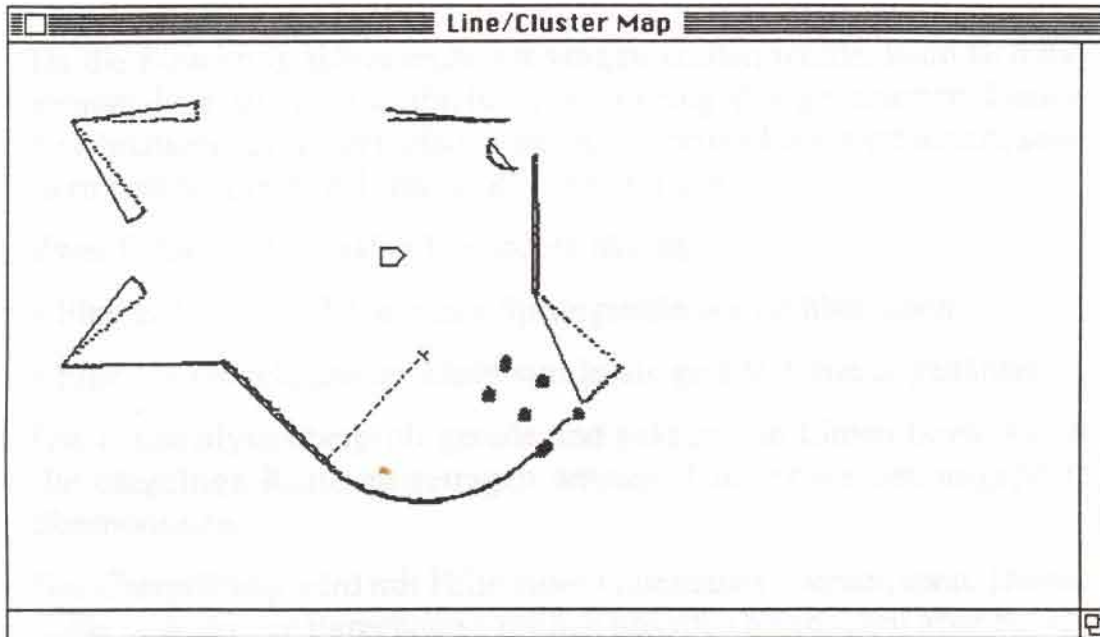
Wird eine gerade Linie ermittelt, so wird als Start- und Endpunkt die entsprechenden Meßwerte in der Entfernungsaufnahme genommen. Dies hat den Vorteil, daß aneinanderstoßende Linien auch in der generierten Karte auf einen Punkt zulaufen. Der Nachteil ist jedoch, daß Linienendpunkte immer real aufgenommene Punkte sind. Die Genauigkeit der Entfernungskarte geht also stark in die Linienkarte ein.

Hat eine Linie nur wenige Punkte, so wird ihr der Typ Cluster zugeordnet, da die Informationen zur Linienbildung zu vage sind.

Wird eine gekrümmte Linie erkannt, so müssen noch Werte berechnet werden, die diese Krümmung eindeutig bestimmen. Der Start- und Endpunkt der Linie sind schon bekannt; nimmt man jetzt noch einen Punkt der genau in der Mitte liegt hinzu, so kann man daraus Radius und Zentrum der Krümmung berechnen.

Der Öffnungswinkel ist zwar nicht essentiell, man kann ihn aber leicht aus der Krümmungsfunktion ermitteln. Dieser Wert erhebt dann aber nicht mehr den Anspruch auf mathematische Exaktheit, sondern ist ein statistischer Wert ähnlich dem Eckenwinkel bei aufeinanderstoßenden Linien.

Wendet man nun das Verfahren auf unseren Beispielszene an, so erhält man eine Karte, wie in Bild 3.4.a dargestellt.



Das Verfahren

Bild 3.4.a: Ermittelte Linien aus der Beispielszene

Die Linienkarte wurde hier über die Entfernungskarte gelegt. Graue Punkte deuten Cluster an. Die gekrümmte Linie wurde mit einem Radius versehen um sie von geraden Linien zu unterscheiden.

Positiv ist zu vermerken, daß das Verfahren die Krümmung sowie die wesentlichen Raumecken erkannt hat. Ein Nachteil fällt jedoch sofort ins Auge: Ecken unmittelbar vor einer Sprungkante wurden nicht entdeckt. Gut zu sehen ist dies besonders bei der rechten Zimmerecke, die völlig abgeschnitten wurde. Außerdem wurde der Papierkorb nur durch eine gerade Strecke dargestellt.

Der Grund für dieses Verhalten ist der folgende: bei einer Sprungkante ist ein kompletter Bereich nicht mehr verwertbar. Befinden sich dort Ecken oder runde Gegenstände, so werden diese nicht erkannt.

Damit das Verfahren jedoch einsetzbar ist, muß dieser Nachteil durch eine nachgeschaltete Analyse behoben werden.

3.5. Die Feinanalyse

Da die Einteilung schon recht gut vorgenommen wurde, kann sich die Feinanalyse auf eine einfache Überprüfung der generierten Linien beschränken. Es werden also keine völlig neuen Linien generiert, sondern bestenfalls eine Linie in kleinere zerteilt.

Zwei Fehler treten bisher besonders häufig auf:

- Eine Ecke in der Nähe einer Sprungstelle wurde übersehen.
- Eine kleine gekrümmte Linie wurde als gerade Linie angenähert.

Die Feinanalyse überprüft gerade und gekrümmte Linien bevor sie in die entgültige Karte eingetragen werden. Cluster werden ungeprüft übernommen.

Die Überprüfung wird mit Hilfe eines Gütemaßes übernommen. Dieses sollte sich bei der Berechnung zeitlich unkritisch verhalten aber zuverlässig sein. Hier wurde folgendes Maß verwendet:

An einigen Punkten, die äquidistant auf der Linie liegen, wird der Abstand zur berechneten Linie ermittelt. Von allen Abständen wird das Maximum gebildet.

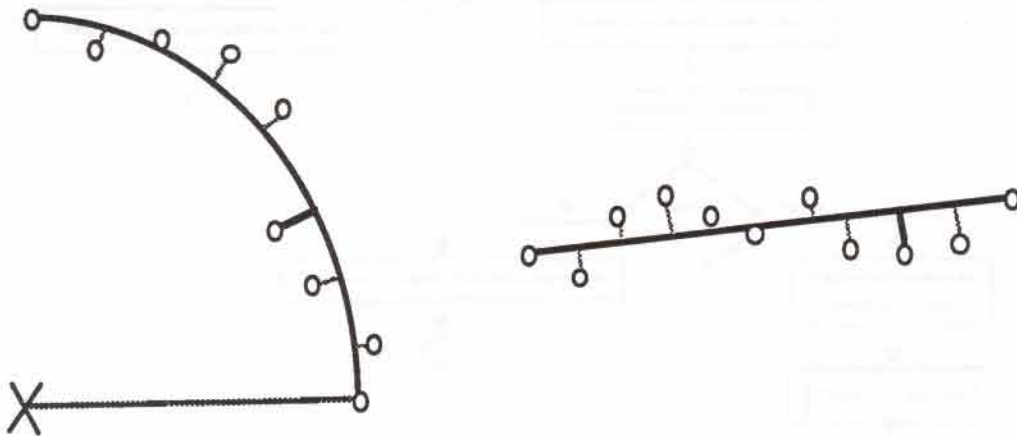


Bild 3.5.a: Gütemaß der generierten Linien

Dieses Maß ist für gekrümmte und gerade Linien gleichermaßen einfach zu berechnen. Die Anzahl der Testpunkte ist so zu wählen, daß mit hoher Wahrscheinlichkeit eventuell vorkommende Abweichungen erkannt werden.

Wird eine Abweichung erkannt, so wird in Abhängigkeit vom vorliegenden Linientyp entweder versucht, die Linie nocheinmal zu unterteilen, oder sie durch einen anderen Typ darzustellen. Konkret heißt dies:

- War der Typ eine gerade Linie, so wird zuerst versucht eine gekrümmte Linie zu nehmen. Schlägt dies auch fehl, so wird der Punkt mit der größten Entfernung zur Strecke ermittelt, und an dieser Stelle die Linie geteilt.
- War der Typ eine gekrümmte Linie, so wird versucht, daraus zwei gerade Linien zu bilden.

Da es sich bei falsch erkannten Linien immer um relativ kleine Stücke handelt, ist der Aufwand gering. Man erzielt jedoch erstaunliche Erfolge. Die Algorithmen, um gerade und gekrümmte Linien in die entgeltige Karte einzutragen, sind auf den folgenden Bildern dargestellt.

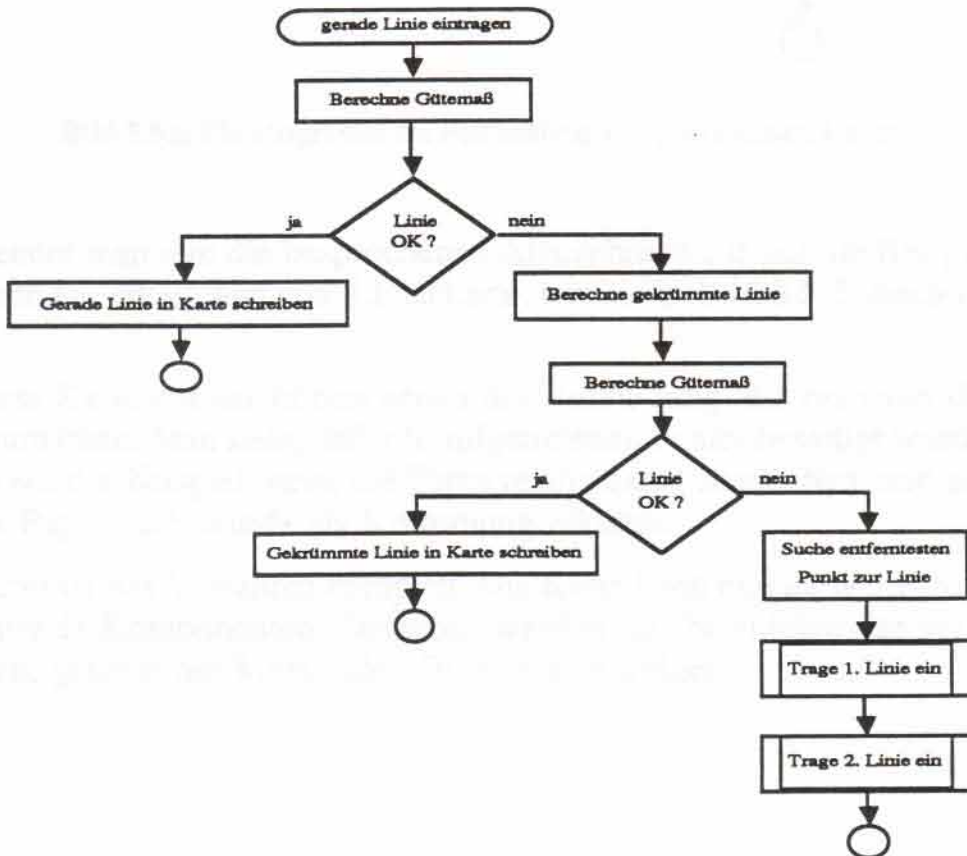


Bild 3.5.b: Flußdiagramm zur Feinanalyse von geraden Linien

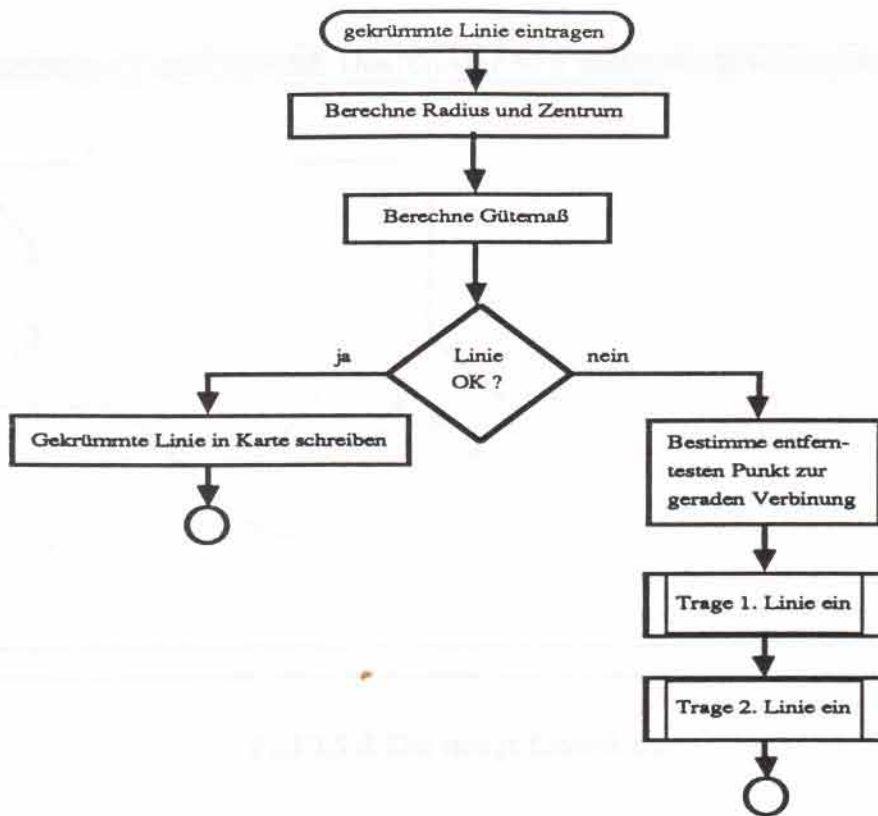
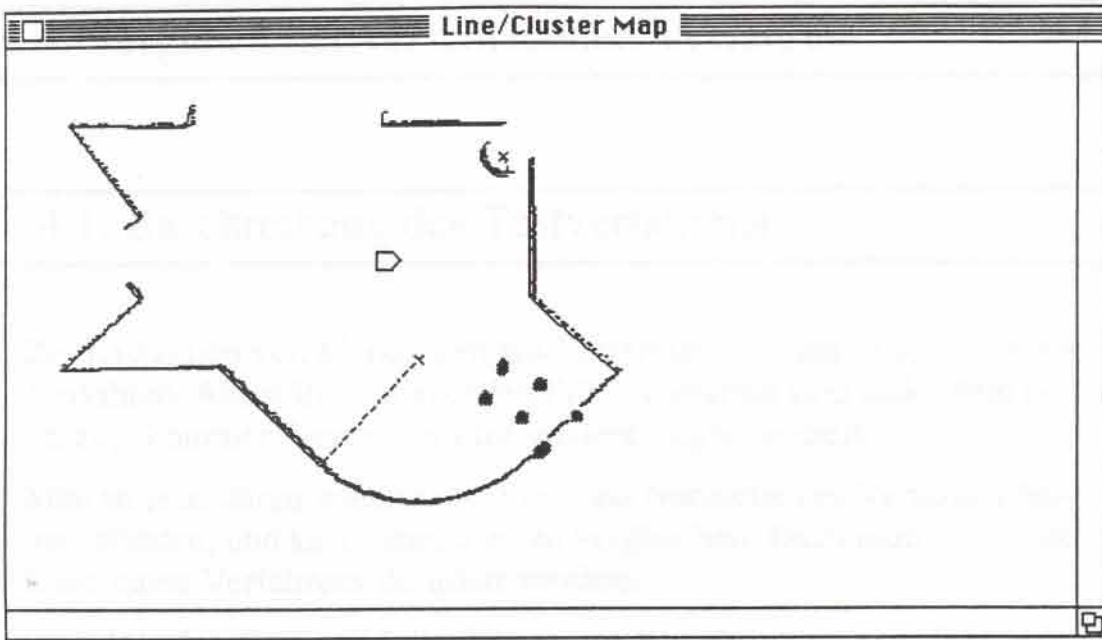


Bild 3.5.c: Flußdiagramm zur Feinanalyse von gekrümmten Linien

Wendet man nun die besprochenen Algorithmen auf auf die Beispielszene an, erhält man eine Linienkarte, wie sie in Bild 3.5.d abgebildet ist.

Diese Karte entspricht nun genau der Vorstellung, die man von dem Raum hatte. Man sieht, daß alle aufgetretenen Fehler beseitigt wurden. So wurden beispielsweise die Türen relativ exakt angenähert, und auch der Papierkorb wurde als Krümmung erkannt.

Damit ist das Verfahren komplett. Die Karte kann nun an weiterverarbeitende Komponenten übergeben werden, die beispielsweise versuchen, generierten Merkmalen Objekte zuzuordnen.



Das Verfahren

Bild 3.5.d: Die fertige Linienkarte

4. Vergleich der verschiedenen Verfahren

4.1. Beschreibung des Testverfahrens

Zur Extraktion von Merkmalen aus Entfernungswerten existieren viele Verfahren. Allein im vorliegenden PFE-Programm sind außer dem neu hinzugekommen Verfahren fünf weitere implementiert.

Man ist jetzt daran interessiert, Vor- und Nachteile der Verfahren herauszufinden, und sie miteinander zu vergleichen. Dazu muß zuerst die Güte eines Verfahrens definiert werden.

Im folgenden Test, wird die Güte eines Verfahrens anhand folgender Kriterien bestimmt:

- die Zeit, die ein Verfahren für die Generierung benötigt:
da die Verfahren im Echtzeitbetrieb eingesetzt werden, ist dieser Punkt interessant.
- die Anzahl der generierten Merkmale:
je weniger Merkmale benötigt werden, um eine Entfernungskarte zu repräsentieren, umso einfacher ist die Verarbeitung für nachgeschaltete Programme.
- die Genauigkeit der Linienkarte:
je besser die generierten Merkmale die Entfernungswerte annähern, umso verlässlicher sind die Aussagen, die man treffen kann.

Es ist klar, daß kein Verfahren alle Verfahren in sich vereinigen kann, da sich die Bedingungen hierfür widersprechen. Man ist aber an Aussagen interessiert, die die Verfahren untereinander in Beziehung setzen.

Der Test besteht darin, daß verschiedene Beispielszenen mit allen Verfahren ausgewertet werden, und dabei charakteristische Größen erhoben werden.

Bei Vergleichstests dieser Art existiert jedoch das Problem, daß eventuell durch die Wahl der Meßgrößen und der Testszenen eine Bevorzugung eines bestimmten Verfahrens erfolgt (bewußt oder unbewußt). Dem entgegenzuwirken wurden acht Szenen, die das gesamte Spektrum

der Komplexität von Szenen überstreichen, aus der Menge der Testszenen ausgewählt, drei davon sind Szenen direkt vom realen Entfernungsmesser. Folgende Szenen wurden ausgewertet:

1. Sehr einfacher Raum mit offener Tür
2. Sehr einfacher Raum mit Tür zu einem Flur
3. Fünfeckiger Raum mit Gegenständen
4. Raum mit Türen und runden Wänden
5. Komplexer Raum mit Gegenständen und runden Wänden
6. - 8. Reale Scans des selben Raums aus drei Positionen

Die Entfernungskarten der Räume sind in den Bildern 4.1.a bis 4.1.h dargestellt.

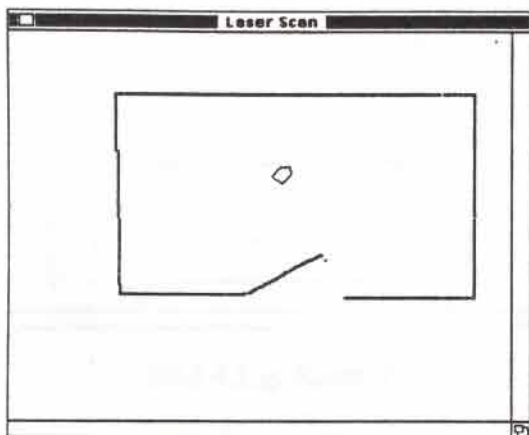


Bild 4.1.a: Raum 1

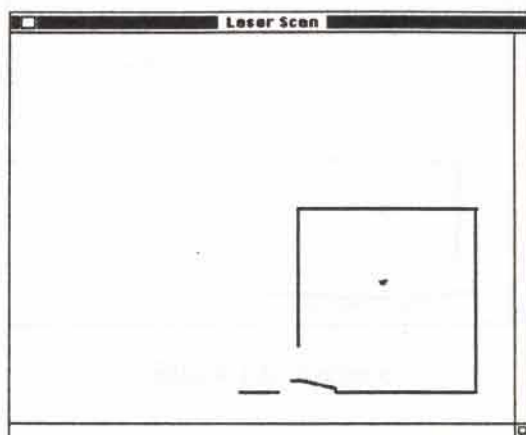


Bild 4.1.b: Raum 2

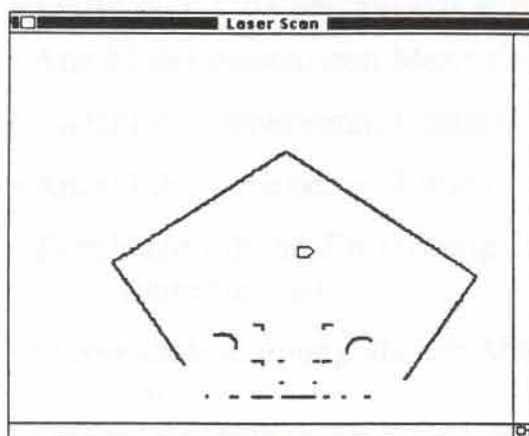


Bild 4.1.c: Raum 3

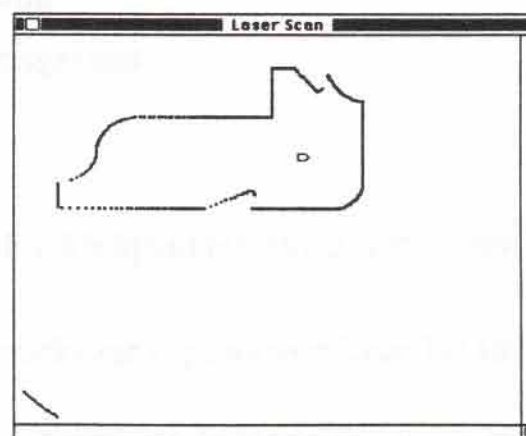


Bild 4.1.d: Raum 4

Vergl. der
Verfahren

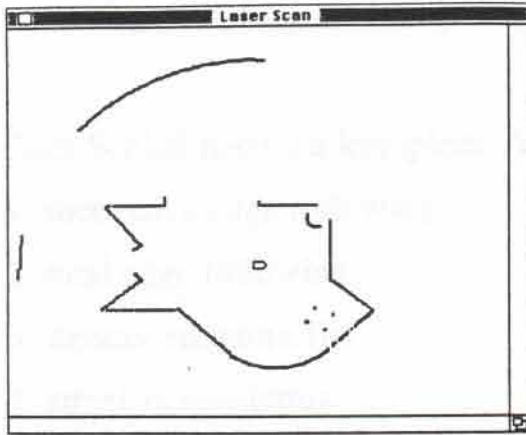


Bild 4.1.e: Raum 5

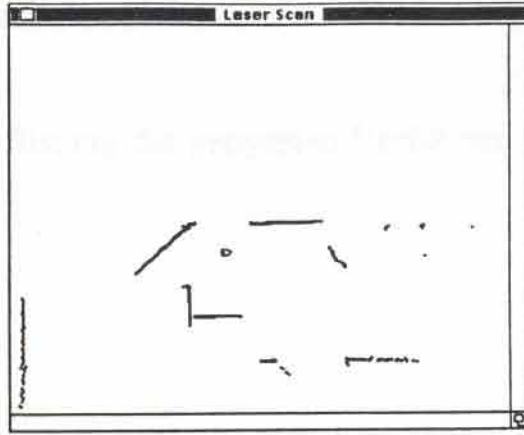


Bild 4.1.f: Raum 6

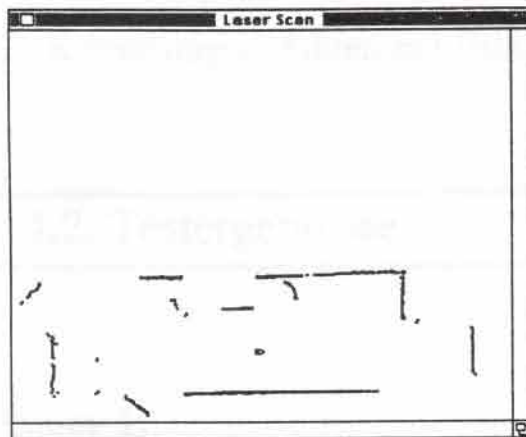


Bild 4.1.g: Raum 7

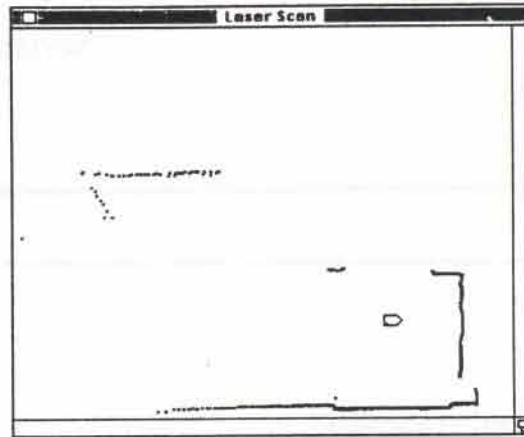


Bild 4.1.h: Raum 8

Vergl. der
Verfahren

Bei den Testgrößen wurde versucht, alle Werte die aussagekräftig sind, zu ermitteln. Folgende wurden gewählt:

- Anzahl der generierten Merkmale insgesamt
- Anzahl der generierten Cluster
- Anzahl der generierten Linien
- Durchschnittliche Entfernung jedes Meßpunktes zur zugehörigen Linie (in cm)
- Maximale Entfernung, die ein Meßpunkt zur zugehörigen Linie hat (in cm)
- Zeit die die Line/Clustergenerierung benötigt (gemessen auf einem Macintosh II in Sekunden)
- Zeit die ein kompletter Durchlauf benötigt (Preprocessing, Histog-

ramme, Line/Clustergenerierung in Sekunden)

Zum Schluß noch die komplette Auflistung der getesteten Verfahren:

1. successive edge following
2. total edge following
3. iterativ endpoint fits
4. street Algorithmus
5. line tracking
6. Krümmungsverfahren ohne Feinanalyse
7. Krümmungsverfahren mit Feinanalyse

4.2. Testergebnisse

Raum 1:

	succ edge	total egde	endpointfits	street	line tracking	Krümmungsverfahren	
						ohne Fein.	mit Fein.
Features	8	8	6	6	6	6	7
Linien	8	8	6	6	6	6	7
Cluster	0	0	0	0	0	0	0
Durch. Entf.	0.000	0.000	0.333	2.833	5.333	0.500	0.143
Max. Entf.	1	1	5	10	17	5	2
Zeit L/C	0.867	0.817	1.167	0.817	1.017	0.600	0.600
Zeit total	2.417	2.367	2.717	2.367	2.567	2.150	2.150

Vergl. der
Verfahren

Raum 2:

Krümmungsverfahren

	succ edge	total egde	endpointfits	street	line tracking	ohne Fein.	mit Fein.
Features	7	7	7	10	8	6	7
Linien	6	6	7	10	8	6	7
Cluster	1	1	0	0	0	0	0
Durch. Entf.	0.000	0.000	0.000	0.600	3.375	0.500	0.000
Max. Entf.	3	3	3	9	21	7	3
Zeit L/C	0.783	0.750	1.100	0.833	0.983	0.550	0.550
Zeit total	2.317	2.283	2.633	2.367	2.517	2.083	2.083

Raum 3:

Krümmungsverfahren

	succ edge	total egde	endpointfits	street	line tracking	ohne Fein.	mit Fein.
Features	27	29	23	23	21	21	21
Linien	18	20	14	14	12	12	12
Cluster	9	9	9	9	9	9	9
Durch. Entf.	0.222	0.250	0.429	0.929	2.917	1.250	0.250
Max. Entf.	5	5	4	6	15	9	3
Zeit L/C	0.783	0.783	1.100	0.833	0.983	0.600	0.600
Zeit total	2.383	2.383	2.700	2.433	2.583	2.217	2.200

Vergl. der
Verfahren**Raum 4:**

Krümmungsverfahren

	succ edge	total egde	endpointfits	street	line tracking	ohne Fein.	mit Fein.
Features	13	13	19	19	18	14	24
Linien	12	12	17	17	16	12	22
Cluster	1	1	2	2	2	2	2
Durch. Entf.	6.083	6.083	0.588	1.588	2.312	2.083	0.045
Max. Entf.	93	93	5	11	13	17	2
Zeit L/C	0.750	0.733	1.133	0.817	0.983	0.550	0.683
Zeit total	2.133	2.177	2.517	2.200	2.367	1.933	2.067

4.2. Testergebnisse

Raum 5:

	Krümmungsverfahren						
	succ edge	total egde	endpointfits	street	line tracking	ohne Fein.	mit Fein.
Features	23	22	25	24	21	17	22
Linien	18	16	19	18	15	11	16
Cluster	5	6	6	6	6	6	6
Durch. Entf.	1.722	1.938	0.263	1.111	2.867	3.182	0.062
Max. Entf.	49	49	3	7	10	28	2
Zeit L/C	0.617	0.617	0.950	0.700	0.800	0.483	0.550
Zeit total	1.833	1.833	2.167	1.917	2.017	1.700	1.767

Raum 6:

	Krümmungsverfahren						
	succ edge	total egde	endpointfits	street	line tracking	ohne Fein.	mit Fein.
Features	39	37	20	24	19	14	24
Linien	34	30	13	17	12	10	20
Cluster	5	7	7	7	7	4	4
Durch. Entf.	0.265	0.300	0.538	0.647	0.917	1.500	0.300
Max. Entf.	6	6	6	8	8	12	4
Zeit L/C	0.500	0.483	0.617	0.517	0.567	0.317	0.467
Zeit total	1.450	1.433	1.567	1.467	1.517	1.267	1.417

Vergl. der
Verfahren

Raum 7:

	Krümmungsverfahren						
	succ edge	total egde	endpointfits	street	line tracking	ohne Fein.	mit Fein.
Features	46	46	28	32	28	32	39
Linien	30	29	14	18	14	15	22
Cluster	16	17	14	14	14	17	17
Durch. Entf.	0.400	0.414	0.857	0.778	0.929	0.867	0.364
Max. Entf.	6	6	6	8	7	11	5
Zeit L/C	0.733	0.667	0.833	0.717	0.833	0.500	0.567
Zeit total	2.000	1.933	2.100	1.983	2.100	1.767	1.833

4.2. Testergebnisse

Raum 8:

Krümmungsverfahren

	succ edge	total egde	endpointfits	street	line tracking	ohne Fein.	mit Fein.
Features	28	28	12	13	11	17	21
Linien	25	25	8	9	7	15	19
Cluster	3	3	4	4	4	2	2
Durch. Entf.	0.240	0.240	0.875	0.333	1.286	0.600	0.211
Max. Entf.	5	5	6	6	11	13	6
Zeit L/C	0.533	0.500	0.683	0.550	0.650	0.400	0.433
Zeit total	1.567	1.533	1.717	1.583	1.683	1.433	1.467

4.3. Auswertung der Ergebnisse

An dieser Stelle soll versucht werden, die gewonnenen Ergebnisse zu interpretieren. Um einen besseren Eindruck zu gewinnen ist der Durchschnitt über alle Testszenen angegeben. Dagegen spricht, daß man damit alle Beispiele gleich bewertet und daß die Anzahl der Szenen relativ gering ist. Doch obwohl damit die Aussagekraft zumindest mit Vorsicht zu genießen ist, kann man so einen Überblick gewinnen.

Vergl. der
Verfahren**Durchschnitt:**

Krümmungsverfahren

	succ edge	total egde	endpointfits	street	line tracking	ohne Fein.	mit Fein.
Features	23.88	23.75	17.50	18.88	16.50	15.88	20.62
Linien	18.88	18.25	12.25	13.63	11.25	10.88	15.62
Cluster	5.00	5.50	5.25	5.25	5.25	5.00	5.00
Durch. Entf.	1.117	1.153	0.485	1.102	2.492	1.310	0.172
Max. Entf.	21.00	21.00	4.75	8.13	16.00	12.75	3.375
Zeit L/C	0.696	0.669	0.948	0.723	0.852	0.500	0.556
Zeit total	2.013	1.985	2.265	2.040	2.196	1.819	1.873

Zuerst fällt auf, daß die ersten beiden Verfahren (successive und total edge following) fast dieselben Eigenschaften besitzen. So erzeugen sie in nahezu derselben Zeit praktisch dieselbe Anzahl Linien und Cluster. Auch die Fehler sind bei beiden Verfahren ähnlich. Im Vergleich zu den anderen Algorithmen besitzen sie beide die größte Fehlerate. So sind die Fehler in Raum 4 (maximal 93 cm) und Raum 5 (maximal 49 cm) nicht mehr zu tolerieren. Von der Komplexität sind diese Verfahren jedoch relativ günstig.

Auch das Verfahren line tracking verursacht grobe Fehler. Diese sind zwar nicht so groß wie in den ersten beiden Algorithmen, aber teilweise sind die maximalen Fehler größer als jene vom Krümmungsverfahren ohne Feinanalyse.

Vergeicht man alle Berechnungszeiten miteinander, so schneidet das Krümmungsverfahren am besten ab. Dabei fällt die Zeit für die Feinanalyse verglichen mit der Gesamtzeit kaum ins Gewicht.

Die Feinanalyse selbst bewirkt eine enorme Qualitätssteigerung. Während ohne Feinanalyse noch grobe Fehler auftraten (z.B. in Raum 5), ist das Verfahren mit Feinanalyse sehr präzise. So treten hier im Vergleich mit allen anderen Verfahren die geringsten durchschnittlichen Fehler auf. Auch die maximalen Fehler sind sehr gering. Jedoch wurden deswegen mehr Linien erzeugt. Beispielsweise generiert das Verfahren in Raum 4 gegenüber den anderen Verfahren die meisten Linien. Diese Anzahl ist aber anscheinend notwendig, da alle anderen Verfahren hier relativ große Fehler machen. Insgesamt liegt die Anzahl der generierten Merkmale im Vergleich zu anderen Verfahren ungefähr in der Mitte.

Vergl. der
Verfahren

Zusammenfassend kann man über die Verfahren folgende Aussagen treffen:

- successive, total edge following:
günstiges Zeitverhalten, Fehlerrate trotz großer Linienzahl sehr hoch
- iterativ endpointfits:
ungünstiges Zeitverhalten, relativ präzise
- street algorithm:
zeitliches Verhalten sowie Präzision durchschnittlich
- line tracking:
weniger günstiges Zeitverhalten, ungünstige Fehlerrate

- Krümmungsverfahren ohne Feinanalyse:
sehr günstiges Zeitverhalten, grobe Fehler
- Krümmungsverfahren mit Feinanalyse:
günstiges Zeitverhalten, sehr präzise

Diese Zusammenhänge sind nocheinmal in der folgenden Übersicht angegeben. Dabei bedeuten die Symbole:

- ++:sehr günstig
- + :günstig
- 0 :durchschnittlich
- :ungünstig
- :sehr ungünstig

	Zeit	Präzision	Datenmenge
succ. edge following	+	--	--
total edge following	+	--	--
iterative endpointfits	--	+	0
street algorithm	0	0	0
line tracking	-	-	+
Krümmungsv. ohne	++	-	++
Krümmungsv. mit Fein.	++	++	-

Vergl. der Verfahren

Bild 4.3.a: Bewertung der Verfahren

5. Benutzerschnittstelle

5.1. Menus

In der neuen PFE Version sind einige neuen Menüpunkte implementiert worden. Diese sollen im folgenden erläutert werden.

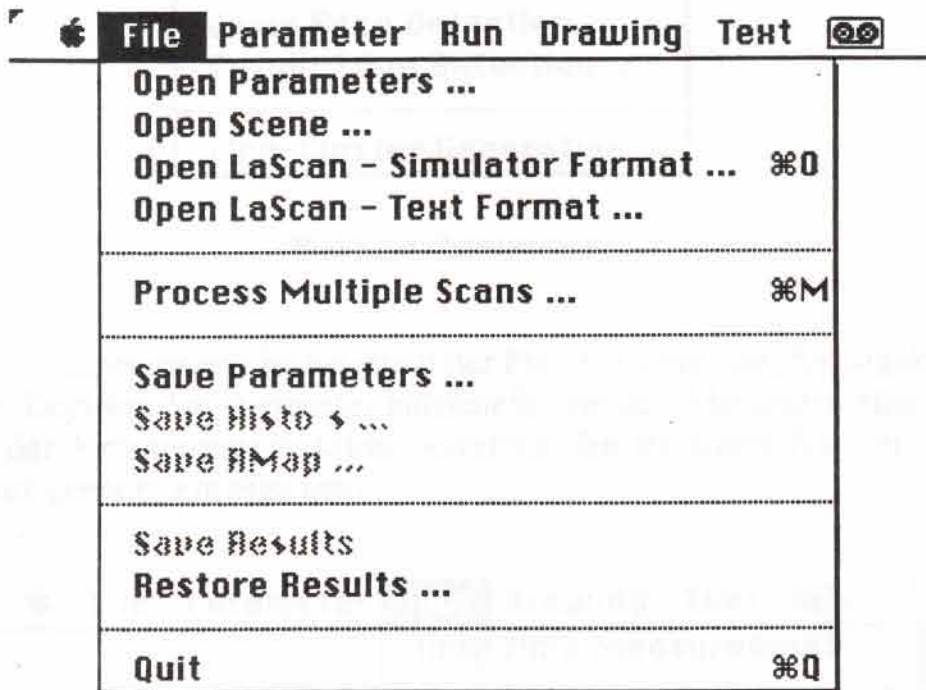


Bild 5.1.a: Filemenü

Benutzer-
schnittstelle

Gegenüber der alten Version sind zum Filemenü zwei Menüpunkte hinzugekommen. Zum ersten ist es möglich, eine Szene zu laden ("Open Scene"), um diese mit der Entfernungskarte zu vergleichen. Diese dient jedoch nur der optischen Untermalung, sie fließt in keine Berechnung ein.

Zum anderen ist es möglich Berechnungen für eine Serie von Laserscans automatisch durchzuführen ("Process Multiple Scans"). Eine Serie ist dadurch gekennzeichnet, daß alle Dateien denselben Namen besitzen, sowie die Extension LSCN <n>. Die Nummer ist hierbei eine laufende Nummer, die von 1 startet. Die berechneten Radarmaps werden in den-

selben Ordner wie die Scans geschrieben. Außerdem wird eine Protokolldatei angefertigt, in der eventuell auftretende Fehler vermerkt werden.

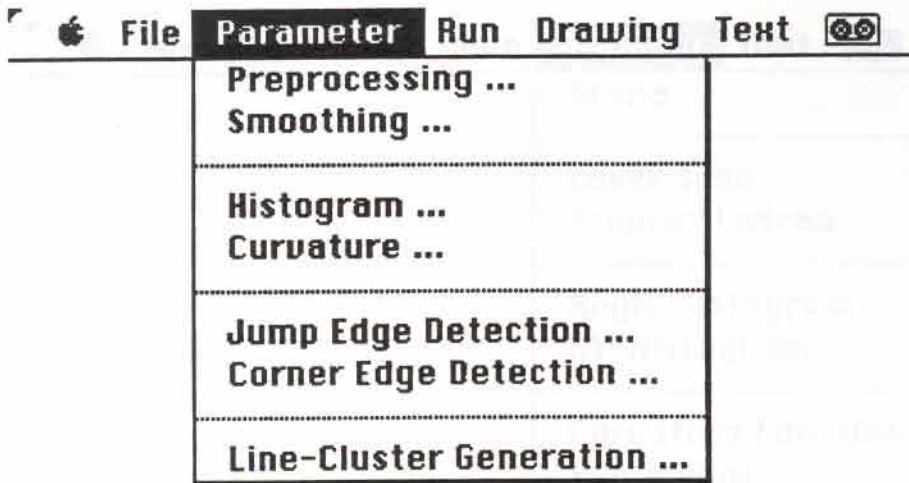


Bild 5.1.b: Parametermenü

Zum Parametermenü ist lediglich der Punkt "Curvature" hinzugekommen. Dort werden Parameter eingestellt, die den Merkmalsextraktion mit der Krümmungsfunktion betreffen. Im nächsten Kapitel wird darauf genauer eingegangen.

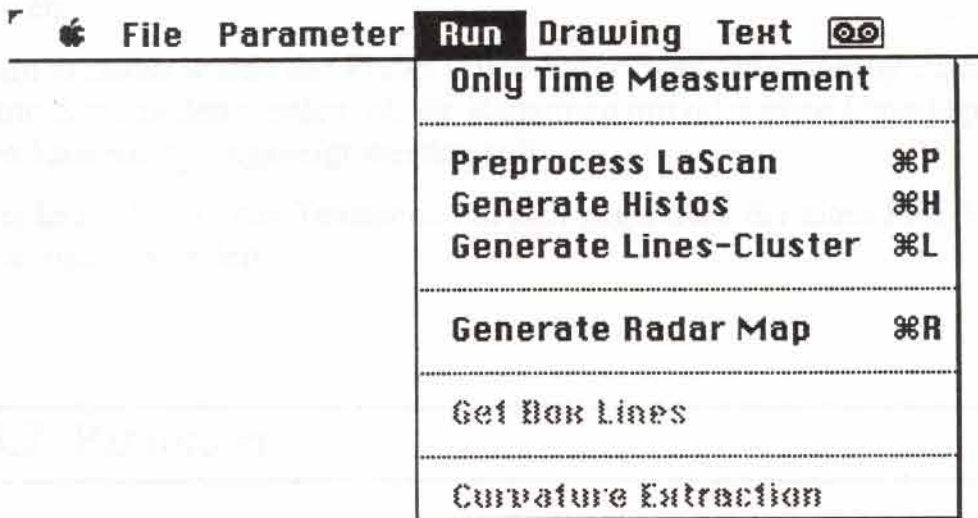


Bild 5.1.c: Runmenü

Benutzer-
schnittstelle

Das Runmenü wurde um den Punkt "Curvature Extraction" erweitert. Hier kann die Auswertung der Krümmungsfunktion explizit gestartet werden. Normalerweise wird sie ohne Kenntnis des Benutzers ausgewertet und nicht angezeigt.

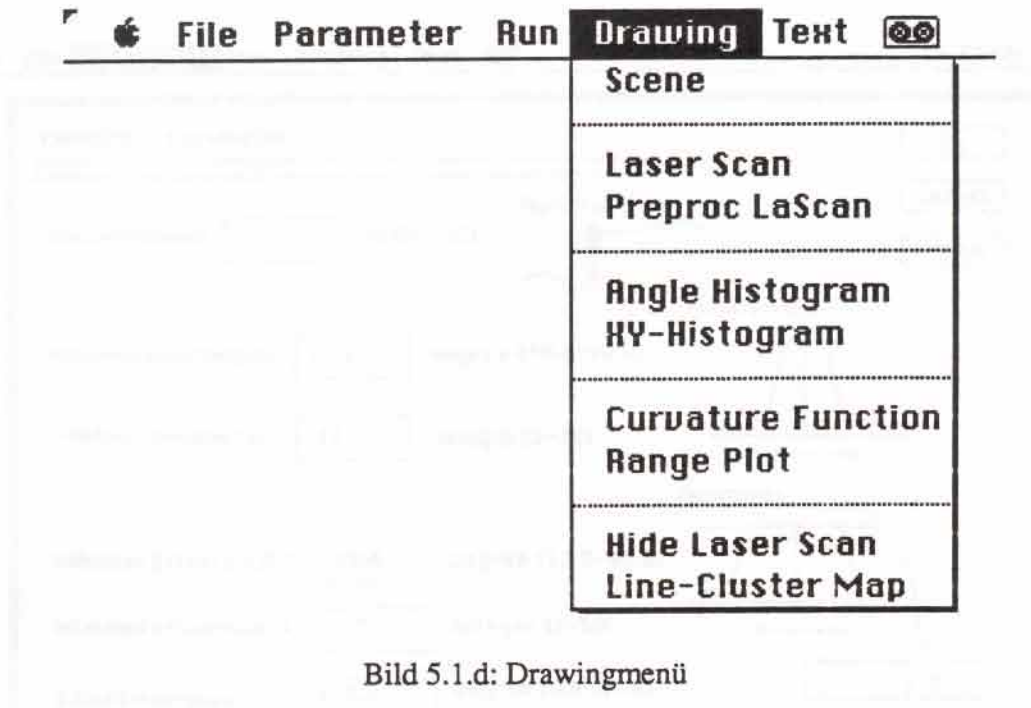


Bild 5.1.d: Drawingmenü

Das Drawingmenü wurde um zwei Punkte erweitert. Zum einen kann man sich mit "Curvature Function" die Krümmungsfunktion anzeigen lassen.

Zum anderen wurde der Punkt "Hide Laser Scan" hinzugefügt. Damit kann entschieden werden, ob die Radarmap mit oder ohne Unterlegung des Laserscans angezeigt werden soll.

Das letzte Menü, das Textmenü hat sich gegenüber der alten PFE-Version nicht geändert.

Benutzer-
schnittstelle

5.2. Parameter

Das Verfahren mit Hilfe der Krümmungsfunktion hängt von vielen Werten ab, die durch die jeweilige Konfiguration der Meßeinrichtung gegeben sind. So muß beispielsweise angegeben werden, ab welcher

Entfernungsdifferenz zweier Punkte ein Sprung angenommen wird. Die Menge aller einzustellenden Werte sind in einem Dialog zusammengefaßt. Dieser kann über das Menü Curvature (Parametermenü) angesprochen werden. Der Dialog hat folgendes Aussehen:

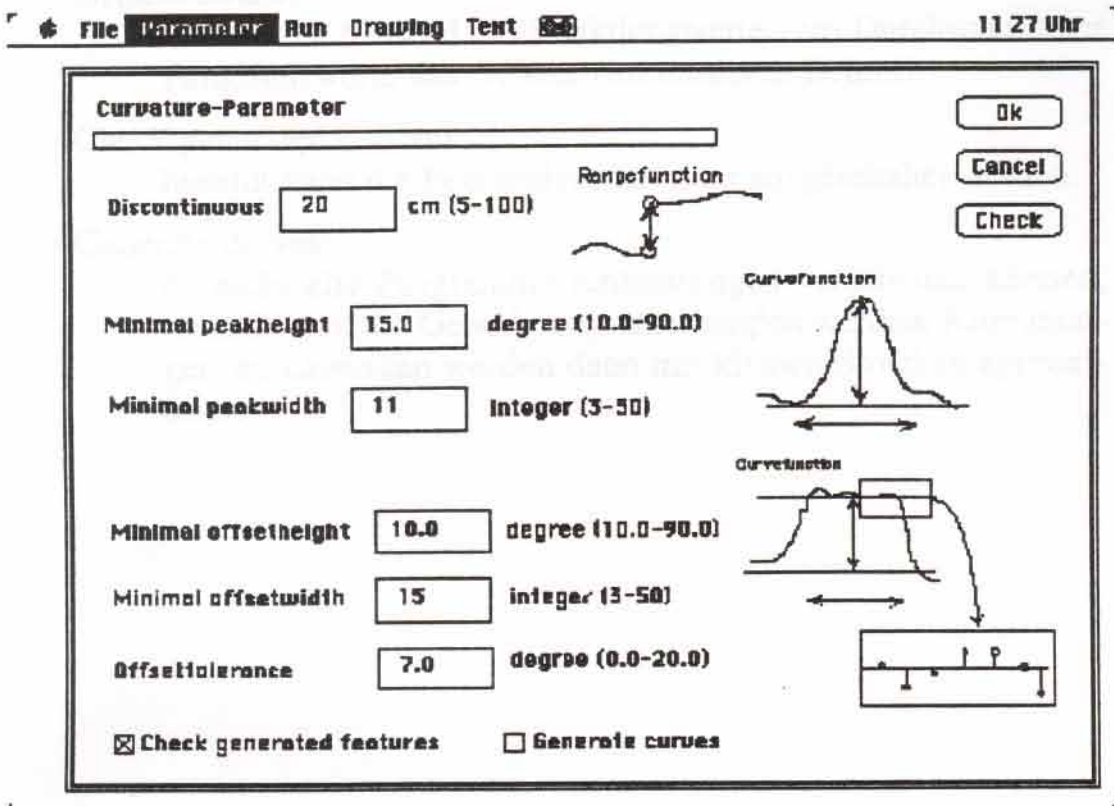


Bild 5.2.a: Dialog zum Verfahren

Dabei haben die Einträge folgende Bedeutung:

- Discontinuous:
Differenz der Entfernungswerte zweier benachbarter Punkte, ab der ein Sprung angenommen wird.
- Minimal peakheight:
minimaler Wert der Krümmungsfunktion, ab der ein Peak erkannt wird.
- Minimal peakwidth:
minimale Anzahl von Meßwerten, die ein Peak besitzen muß.
- Minimal offsetheight:
minimaler Wert der Krümmungsfunktion, ab der ein Offset

erkannt wird.

- **Minimal offsetwidth:**
minimale Anzahl von Meßwerten, die ein Offset besitzen muß.
- **Offsettolerance:**
maximaler Abstand der Funktionswerte vom Durchschnitt der Funktionswerte des Offsets (im mittleren Drittel).
- **Check generated features:**
hiermit kann die Feinanalyse an- oder ausgeschaltet werden.
- **Generate curves:**
da nicht alle Programme Krümmungen verarbeiten können, kann hiermit die Generierung unterbunden werden. Krümmungen im Laserscan werden dann mit kleinen Strecken approximiert.

6. Implementierung

Das PFE-Programm besteht in der vorliegenden Fassung aus 40 Modulen. Gegenüber der alten Version sind vier Module hinzugekommen:

- **CurvatureAlgorithms:**
hier befinden sich alle Unterprogramme, die zur Durchführung des Verfahrens dienen. Die Krümmungsfunktion selbst wird im Modul HistogrammGeneration berechnet. In CurvatureAlgorithms werden ihre Merkmale bestimmt, und daraus die Radar-Map erzeugt.
- **DrawCurvature:**
in diesem Modul befinden sich die Zeichenroutinen für die Krümmungsfunktion. Im wesentlichen sind dies das Zeichnen der Funktion im Urzustand, sowie nach der Extraktion.
- **PFEProcessMultipleScans:**
da es für lange Serien von Scans umständlich ist, die RadarMap einzeln zu erzeugen, werden hier Routinen bereitgestellt, die dies automatisch übernehmen.
- **PFERemoteCall:**
da das PFE-Programm nur eines in einer Reihe von Programmen ist, wird hiermit die Möglichkeit gegeben, PFE ohne Benutzerführung aufzurufen. Von einem übergeordneten Programm, dem *Supervisor* erhält dann PFE einen Auftrag, einen oder mehrere Scans zu bearbeiten. Nach Erledigung terminiert PFE, und die nächste Komponente wird gestartet. RemoteCall importiert Routinen einer Unit Remote, die für alle Programme ein Interface zum Supervisor darstellt.

Das folgende Bild stellt die Import/Export-Beziehung unter den Modulen dar. Dabei bedeutet eine gesetzte Markierung, daß das zugehörige obere Modul Routinen, Variablen, Typen oder Konstanten exportiert, während das entsprechende Modul links diese importiert.

A.1.:

Beschreibung der Datentypen und Routinen, der neu hinzugekommenen Module

A.2.:

Listing der neuen Module